المركز الوطني
لتطوير المناهج
National Center
for Curriculum
Development

المهارات الرقمية

الصف الثاني عشر - كتاب الطالب

الفصل الدراسي الأول

12

لجنة الإشراف على التأليف

أ.د. وليد خالد سلامة

أ.د.باسل علي محافظة

أ.د. خالد إبراهيم العجلوني

ليلى محمد العطوي

هذا الكتاب جزء من مشروع الشباب والتكنولوجيا والوظائف
لدى وزارة الإقتصاد الرقمي والريادة.

الناشر: المركز الوطني لتطوير المناهج

يسر المركز الوطني لتطوير المناهج استقبال آرائكم وملحوظاتكم على هذا الكتاب عن طريق العناوين الآتية:

06-5376262 /237 | 06-5376266 | P.O.Box: 2088 Amman 11941

@nccdjor | feedback@nccd.gov.jo | www.nccd.gov.jo

قرَّرت وزارة التربية والتعليم تدريس هذا الكتاب في مدارس المملكة الأردنية الهاشمية جميعها، بناءً على قرار المجلس الأعلى للمركز الوطني لتطوير المناهج في جلسته رقم (3/ 2025)، تاريخ (27/ 3/ 2025)، وقرار مجلس التربية والتعليم رقم (65/ 2025)، بدءًا من العام الدراسي (2025/ 2026)

ISBN 978-9923-41-886-4

المملكة الأردنية الهاشمية

رقم الإيداع لدى دائرة المكتبة الوطنية:

(2025/2/845)

الأردن، المركز الوطني لتطوير المناهج

المهارات الرقمية، كتاب الطالب: الصف الثاني عشر، الفصل الدراسي الأول

عمّان، المركز الوطني لتطوير المناهج، 2024

373.19

علم الحاسوب/ / أساليب التدريس/ / المناهج/ / التعليم الثانوي/

يتحمَّل المُؤلِّف كامل المسـؤولية القانونية عن محتوى مُصنَّفه، ولا يُعبِّر هذا المُصنَّف عن رأي دائرة المكتبة الوطنية.

فريق التأليف المكلف من شركة عالم الاستثمار للتنمية والتكنولوجيا

د. أسماء حسن حمدان

أ. د. محمد يونس العزة

د. محمد رجب عبدالمجيد

حنان حسني أبوراشد

منهاجي

متعة التعليم الهادف

الطبعة الأولى (التجريبية)

1446هـ/ 2025م

المقدمة

انطلاقًا من إيمان المملكة الأردنية الهاشـــمية بأهمية تنمية قدرات الإنسان الأردني، وتسليحه بالعلم والمعرفة؛ ســـعى المركز الوطني لتطوير المناهج، بالتعاون مـــع وزارة التربية والتعليم، إلى تحديث المناهج الدراســـية وتطويرها؛ لتكون مُعينًا للطلبة على الارتقاء بمستواهم المعرفي والمهاري، ومجاراة أقرانهم في الدول المُتقدِّمة. ونظرًا إلى أهمية مبحث المهارات الرقميـــة ودوره في تنمية مهارات التفكير لدى الطلبة، وفتح آفاق جديدة لهم تُواكِب مُتطلَّبات ســـوق العمل؛ فقد أَوْلى المركز مناهجه عناية فائقة، وأَعدَّها وَفق أفضل الأســـاليب والطرائق المُتَّبَعة عالميًّا بإشراف خبراء أردنيين؛ لضمان توافقها مع القِيَم الوطنية الأصيلة، ووفائها بحاجات الطلبة.

يُعَدُّ مبحث المهارات الرقمية واحدًا من أهمِّ المباحث الدراسية؛ إذ يُمثِّل الخطوة الأولى لتعريف الطلبة بمناحي التكنولوجيا والتطوُّر الرقمي الحديث بصورة موثوقة وآمنة. وقد اشتمل كتاب المهارات الرقمية على موضوعات تراعي التدرُّج في تقديم المعلومة، وعرضها بأســـلوب مُنظَّم وجاذب، وتعزيزها بالصور والأشـــكال؛ ما يُثْري المعرفة لدى الطلبة، ويُعزِّز رغبتهم في التعلُّم، ويُحفِّزهم إلى أداء أنشـــطة الكتاب المُتنوِّعة بيسر وسهولة، فضلًا عن تذكيرهم بالخبرات والمعارف التعليمية التي اكتسبوها سابقًا.
روعي في إعداد الكتاب الربط بين الموضوعات الجديدة على نحوٍ شامل ومُتكامِل، وتقديم موضوعاته بصورة شائقة تُعْنى بالســـياقات الحياتية التي تَهُمُّ الطلبة، وتزيد من رغبتهم في تعلُّم المهارات الرقمية. وقد أُلحِق بكل وحدة مقاطع تعليمية مُصوَّرة، تساعد الطلبة على الفهم العميق للموضوع، وتُرسِّخ لديهم ما تضمَّنه من معلومات وأفكار.

ونظرًا إلى ما تُمثِّله الأنشـــطة من أهمية كبيرة في فهم الموضوعات وتعزيـــز الطلاقة الإجرائية لدى الطلبة؛ فقد اشتمل الكتاب على أنشطة مُتنوِّعة تحاكي واقع الطلبة وما يحيط بهم، وتدعم تعلُّمهم، وتُثْري خبراتهم، فضلًا عن اشتماله على روابط إلكترونية يُمكِن للطلبة الاستعانة بها عند البحث في الأوعية المعرفية. ومن ثَمَّ، فإنَّ المهارات الرقمية والتقنية ترتبط ارتباطًا وثيقًا بمسيرة الطلبة التعليمية والمهنية.

ونحن إذ نُقدِّم هذا الكتاب، فإنَّنا نأمل أنْ يُســـهِم في بناء جيل واعٍ ومُبتكِر وقـــادر على التعامل مع التكنولوجيا بمسؤولية وإبداع، وأنْ يكون لَبِنة أساسية في تقدُّم المملكة الأردنية الهاشمية وازدهارها.

المركز الوطني لتطوير المناهج

الفهرس

الوحدة الأولى: الخوارزميات والبرمجة (Algorithms and Programing) 8

الدرس الأوَّل: دورة حياة تطوير النظام (System Development Life Cycle: SDLC). 10

مفهوم دورة حياة تطوير النظام (SDLC) 11

مراحل دورة حياة تطوير النظام 11

النماذج الخاصة بدورة حياة تطوير النظام (SDLC Models) 16

الدرس الثاني: الخوارزميات (Algorithms) 26

حَلُّ المسألة (Problem Solving) 27

الجمل الشرطية 32

الخوارزمية لجمل التكرار 41

الدرس الثالث: هياكل تراكيب البيانات (Data Structures) 47

البيانات المُركَّبة في لغة البرمجة بايثون 48

1– القوائم (Lists) 48

2– الصفوف (Tuples) 50

3– القواميس (Dictionaries) 53

4– المجموعات (Sets) 55

5– نوع البيانات المُجرَّدة (Abstract Data Type: ADT) 56

الدرس الرابع: الدالَّة الراجعة (Recursion Function) 60

الدالَّة الراجعة (Recursion) 61

الدرس الخامس: الحُزَم (Packages) 71

الوحدة (Module) والحُزَم (Packages) 72

أسئلة الوحدة 82

تقويم ذاتي (Self-Checklist) 84

الدرس الأوَّل: مُقدِّمة في تحليل البيانات (Introduction to Data Analysis) 90
البيانات: مفهومها، وأنواعها (Data : Concept and Types) 91
أنواع البيانات (Data Types) 92
تحليل البيانات (Data Analysis) 94
مراحل عملية تحليل البيانات 95
أهمية تحليل البيانات 97
أنواع تحليل البيانات (Types of Data Analysis) 102

الدرس الثاني: تقنيات تحليل البيانات (Data Analysis Techniques) 108
أدوات تحليل البيانات (Data Analysis Tools) 116
المفاضلة بين أدوات تحليل البيانات 120

الدرس الثالث: جمع البيانات وتحليلها باستخدام برمجية بايثون
(Python Data Collection & Analyses Methods) 124
تثبيت المكتبات الخارجية في برمجية بايثون 127
استدعاء ملف بيانات في برمجية بايثون 128
عرض ملف البيانات كاملًا 131
تحليل الإحصاءات الوصفية (Descriptive Statistics) في برمجية بايثون 132
عرض البيانات باستخدام برمجية بايثون 135
عرض البيانات برسم بياني خَطّي (Line Chart) 136
عرض البيانات على هيئة رسم بياني بالأعمدة (Bar Chart) 139
عرض البيانات على شكل قطاع دائري (Pie Chart) 141
تحديد الأنماط في الرسوم البيانية 144

الدرس الرابع: علاقة السبب والنتيجة في البيانات
(Data Cause-and-Effect Relation) 150
علاقة السبب والنتيجة (Cause and Effect) 151

الدرس الخامس: الفرضيات في تحليل البيانات (Hypotheses in Data Analysis) ... 161
الفرضية (Hypothesis) 162

أسئلة الوحدة 170

تقويم ذاتي (Self-Checklist) 174

دلالات أيقونات الكتاب

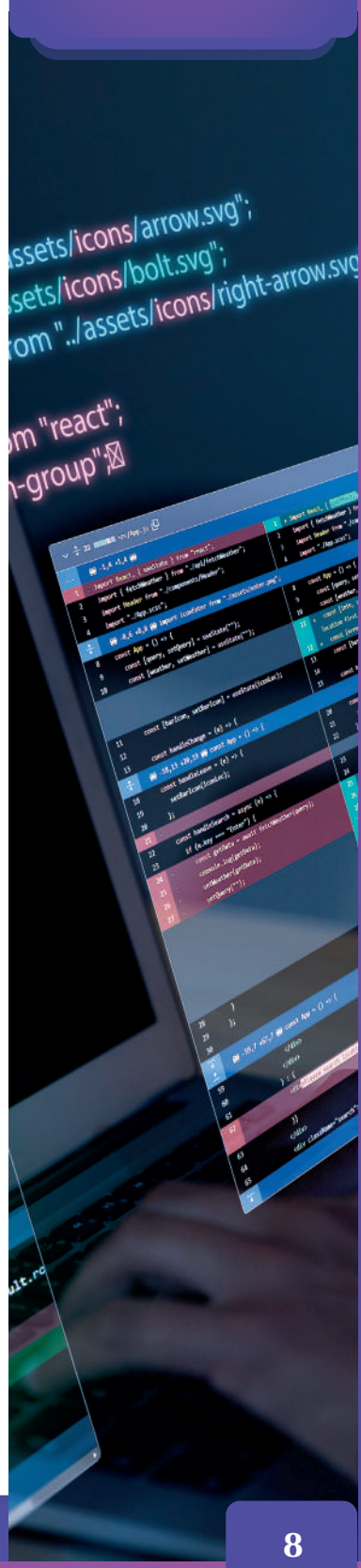
نشاط تمهيدي	نشاط استهلالي يربط التعلُّم السابق بالتعلُّم الحالي.
نشاط عملي	نشاط تطبيقي مُرتبِط بمهارات الدرس.
نشاط	نشاط مُرتبِط بالمحتوى المعرفي أو المحتوى المهاري. للدرس.
نشاط فردي	نشاط يُطبَّق بشكل فردي.
نشاط جماعي	نشاط يُطبَّق في مجموعات.
أبحث	استخدام شبكة الإنترنت في البحث عن المعلومات.

إثراء	توسُّع في المعلومات مُرتبِط بمحتوى الدرس.
أُناقِش	عرض الأفكار وتبادلها مع الزملاء والمُعلِّم/ المُعلِّمة.
إضاءة	معلومة إضافية.
أُشاهِد	عرض محتوى فيديو مُرتبِط بالمحتوى.
مشروع	نشاط تكاملي تُوظَّف فيه معارف الوحدة ومهاراتها.
المواطنة الرقمية	الإجراءات الواجب اتِّباعها لتحقيق مبادئ المواطنة الرقمية.
المهارات الرقمية	المهارات التكنولوجية التي سأُطبِّقها في الوحدة.

1.2TB
1.2TB

الوحدة 1

الخوارزميات والبرمجة (Algorithms and Programing)



نظرة عامة على الوحدة:

سأتعرَّف في هذه الوحدة دورة حياة تطوير النظام، وكيف تُستخدَم المُخطَّطات والخوارزميات شبه الرمزية وهياكل البيانات، ثمَّ أستكشف كيفية عمل الدوال الراجعة والحُزَم البرمجية، وأُنشِئ حُزَمًا برمجيةً باستخدام لغة البرمجة بايثون (Python).

يُتوقَّع منّي في نهاية الوحدة أنْ أكون قادرًا على:

- تعريف دورة حياة تطوير النظام (System Development Life Cycle: SDLC).
- وصف دورة حياة تطوير النظام باستخدام أحد النماذج الآتية: (Spiral, Agile, or Waterfall).
- استخدام المُخطَّطات (Flowcharts) والخوارزمية شبه الرمزية (Pseudocode) في تمثيل الخوارزميات المُركَّبة.
- استخدام أدوات تمثيل المُخطَّطات استخدامًا صحيحًا.
- المقارنة بين هياكل تراكيب البيانات المناسبة أثناء حَلِّ مشكلة مُحدَّدة.
- توضيح مبدأ عمل الدوال الراجعة.
- إيجاد حلول برمجية باستخدام جمل برمجية مُعَدَّة مُسبَقًا (الروتين الفرعي: Modules).
- استخدام لغة البرمجة بايثون في إنشاء مكتبة برمجية (Library).
- تعريف الأنماط البرمجية.
- تبسيط المشكلة وتجزئتها إلى مشكلات صغيرة يُمكِن حَلُّها باستخدام برمجيات أو أجزاء من برامج مُعَدَّة مُسبَقًا.

المهارات الرقمية: التفكير الحاسوبي، حَلُّ المشكلات، الإبداع والابتكار، التواصل الرقمي، المسؤولية الرقمية.

فهرس الوحدة:

- **الدرس الأوَّل:** دورة حياة تطوير النظام (System Development Life Cycle: SDLC).
- **الدرس الثاني:** الخوارزميات (Algorithms).
- **الدرس الثالث:** هياكل تراكيب البيانات (Data Structures).
- **الدرس الرابع:** الدالَّة الراجعة (Recursion Function).
- **الدرس الخامس:** الحُزَم (Packages).

الدرس الأوَّل

دورة حياة تطوير النظام
(System Development Life Cycle: SDLC)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس دورة حياة تطوير النظام ومراحلها، وأستكشف بعض النماذج الشهيرة الخاصة بتطوير الأنظمة وتطبيقها في تصميم البرامج.

المفاهيم والمصطلحات:

دورة حياة تطوير النظام (System Development Life Cycle: SDLC)، التخطيط وتحليل المُتطلَّبات (Planning and Requirement Analysis)، تحديد المُتطلَّبات (Defining Requirements)، التصميم (Design)، التطوير (Development)، الاختبار (Testing)، النشر والصيانة (Development and Maintenance)، مُتطلَّبات البرنامج (Software Requirement Specification: SRS)، التكرار في النموذج الرشيق (Agile Iteration).

نتاجات التعلُّم (Learning Outcomes):

- أُعرِّف دورة حياة تطوير النظام (System Development Life Cycle).
- أَصِف مراحل دورة حياة تطوير النظام.
- أَصِف دورة حياة تطوير النظام باستخدام أحد النماذج الآتية: (Spiral, Agile, or Waterfall).
- أُوضِّح خصائص كل نموذج من نماذج تطوير الأنظمة.
- أُطبِّق طرائق دورة حياة تطوير النظام عند تصميم البرامج.

تعرَّفْتُ سابقًا دورة حياة تطوير البرمجيات، التي تتضمَّن خطوات تطوير المشروع، وتساعد على تنظيم عملية تطوير البرمجيات وإدارتها بصورة فاعلة، فهل توجد نماذج يُعتمَد عليها في تطوير البرمجيات؟

نشاط تمهيدي

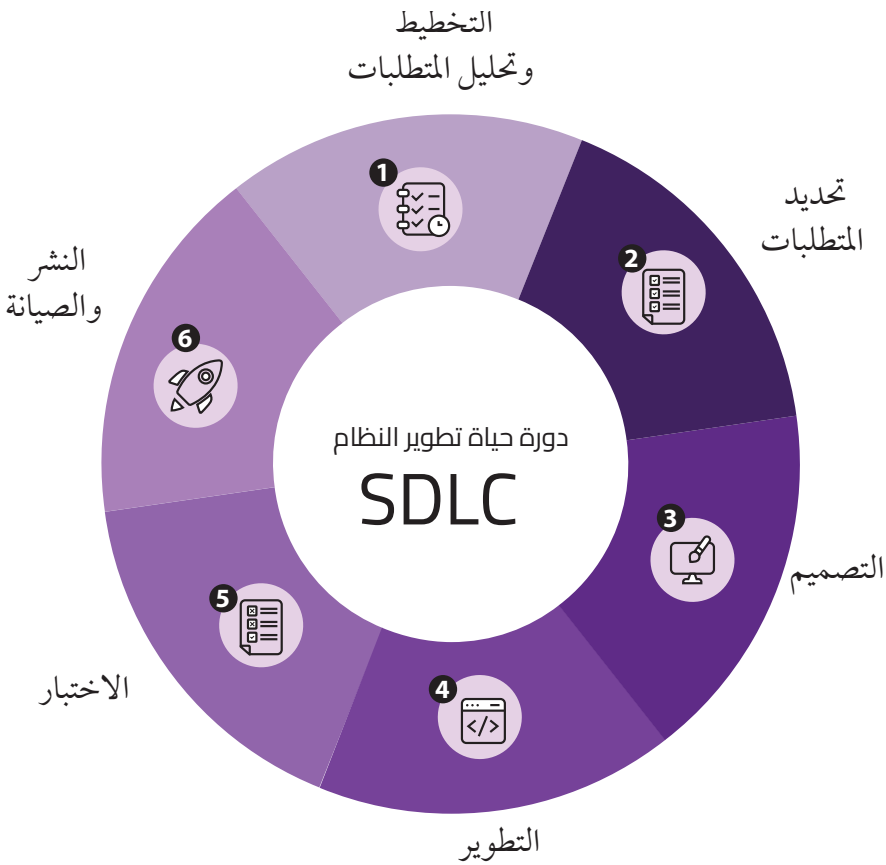
أُفكِّر في المراحل الرئيسة لدورة حياة تطوير النظام (SDLC)، ثمَّ أختار إحدى هذه المراحل للحديث عنها أمام زملائي/ زميلاتي في الصف، وأذكر أمثلة عملية عليها من واقع الحياة.

مفهوم دورة حياة تطوير النظام (SDLC):

تُعرَّف دورة حياة تطوير النظام بأنَّها عملية منهجية مُنظَّمة تَصِف كيف يُمكِن تطوير برامج مُعيَّنة وصيانتها وتحسينها خطوة خطوة بهدف إنتاج برامج فائقة الجودة، وقادرة على الوفاء باحتياجات المُستخدِمين ومُتطلَّباتهم.

مراحل دورة حياة تطوير النظام:

تَمُرُّ عملية تطوير النظام بمراحل مُنظَّمة ومُرتَّبة يتعيَّن على مهندس البرمجيات أو المُطوِّر للأنظمة الحاسوبية أنْ يتَّبعها أثناء هذه العملية لضمان الحصول على مُنتَج مُتميِّز يَفي باحتياجات المُستخدِمين ومُتطلَّباتهم. أنظر الشكل (1-1) الذي يُبيِّن مراحل دورة حياة تطوير النظام.



الشكل (1-1): دورة حياة تطوير النظام.

المرحلة الأولى: التخطيط وتحليل المُتطلَّبات
(Planning and Requirement Analysis)

تتمثَّل هذه المرحلة في تحليل المشروع والتخطيط له عن طريق دراسة الجوانب الآتية: نطاق المشروع، والمشكلة التي يُراد حَلُّها، والعناصر الواجب توافرها من موارد بشرية وأدوات مُتعدِّدة، وجدوى المشروع الاقتصادية، والمخاطر التي قد تعترض عملية تنفيذ المشروع وسُبُل التعامل معها. كذلك يجب الالتقاء بالعملاء لجمع البيانات الخاصة بالمشروع وتعرُّف مواصفاته، ثمَّ إعداد خارطة طريق للتطوير اعتمادًا على ما جُمِع من بيانات، وتضمين خارطة الطريق الجدول الزمني المُحدَّد لتنفيذ المشروع. يلي ذلك تحليل المُتطلَّبات، والإحاطة بما ينبغي للنظام أنْ يُحقِّقه عن طريق التفاعل مع الجهات المَعْنِيَّة من عملاء ومُستخدِمين؛ إمّا بإجراء مقابلات، وإمّا باستخدام استبانات، وإمّا بعمل دراسات وبحوث للوقوف على احتياجات السوق. أنظر الشكل (1-2) الذي يُبيِّن خطوات هذه المرحلة.

الشكل (1-2): خطوات مرحلة التخطيط وتحليل المُتطلَّبات في دورة حياة تطوير النظام.

مثال:

أُطوِّر مشروعًا خاصًّا بإدارة المهام اليومية، وأعمل لذلك على جمع المُتطلَّبات الآتية وتحليلها:

1. تعيين الفئة المُستهدَفة، وتحديد الميزانية المُتوقَّعة والأدوات التي يُراد استخدامها، ووضع جدول زمني مُحدَّد لتنفيذ المشروع.
2. إضافة المهام المطلوبة.
3. تعديل بعض المهام، وحذف ما يَلزم منها.
4. تصنيف المهام بحسب الأولوية.
5. دراسة أنظمة شبيهة لتعرُّف مَواطِن القوَّة ومَواطِن الضعف، وكيف يُمكِن الاستفادة منها.

نشاط فردي

يبحث طبيب أسنان عن برنامج لتنظيم المواعيد الخاصة بعيادته. أُفكِّر في التخطيط لهذا البرنامج عن طريق تحديد المُتطلَّبات الآتية: الفئة المُستهدَفة، والأهداف، والموارد. بعد ذلك أستعمل ورق قلّاب (Flipchart) لرسم مُخطَّط، ثمَّ أُناقِشه مع زملائي/ زميلاتي في الصف.

هل اختلف مُخطَّطي عن مُخطَّطات الزملاء؟ أُبرِّر إجابتي.

المرحلة الثانية: تحديد المُتطلَّبات (Defining Requirements)

تُعَدُّ وثيقة المواصفات الخاصة بمُتطلَّبات البرنامج (Software Requirement Specification: SRS) مُخرَجات هذه المرحلة؛ إذ يُوثَّق فيها كل ما يُحدَّد من مُتطلَّبات النظام الحاسوبي (البرنامج) المُستهدَف بالاتِّفاق مع العملاء ومُحلِّلي السوق. كذلك تشمل هذه المرحلة تحديد مهام النظام، والمُتطلَّبات التكنولوجية الخاصة به.

مثال:

يشتمل مشروع إدارة المهام اليومية على ما يأتي:

أ. **وثيقة مُتطلَّبات البرنامج التي تُحدِّد مهام المُستخدِم**

- إضافة المهام اليومية، وتحديد مواعيد تنفيذها النهائية.
- تعديل المهام.
- حذف المهام.
- تحديد حالة المهمة (مُكتمِلة/ غير مُكتمِلة).
- تصنيف المهام بحسب الأولوية.
- إرسال البرنامج إشعارات إلى المُستخدِم لتذكيره بالأولويات.

ب. **تحديد المُتطلَّبات التكنولوجية** : مثل الأجهزة والأدوات الرقمية والبرمجيات.

نشاط فردي

1- أُحدِّد مهام البرنامج الخاص بحجز المواعيد في عيادة الأسنان ومُتطلَّباته التكنولوجية.

2- ما الطرائق التي سأستخدمها في عملية جمع البيانات؟

3- مَنِ الأشخاص الذين يُمكِن الاستفادة منهم في جمع البيانات اللازمة؟

أُدوِّن النتائج التي أتوصَّل إليها، ثمَّ أُناقِشها مع مُعلِّمي/ مُعلِّمتي وزملائي/ زميلاتي.

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن تعريف لكلٍّ من مفهوم العملاء ومفهوم مُحلِّلي السوق، ثمَّ أُدوِّن ما أتوصَّل إليه في ملف خاص، ثمَّ أُشارِكه مع الزملاء/ الزميلات في الصف.

المرحلة الثالثة: التصميم (Design)

يستفاد من وثيقة المواصفات الخاصة بمُتطلَّبات البرنامج (SRS) في إعداد تصاميم للنظام، وهي تُضمَّن في وثيقة تُسمّى **مواصفات وثيقة التصميم (Design Document Specification: DDS)**. بعد ذلك يُحدَّد التصميم المناسب للنظام عمليًّا ومنطقيًّا بالاتِّفاق مع العملاء ومُحلِّلي السوق. كذلك تُحدَّد في هذه المرحلة المُدخَلات والمُخرَجات وأجزاء النظام، وتُصمَّم واجهة المُستخدِم وقواعد البيانات الخاصة بالنظام، إضافةً إلى تحديد طريقة عمل النظام، وهي المرحلة التي تسبق برمجة النظام.

مثال:

يتطلَّب مشروع إدارة المهام اليومية تصميم واجهة المُستخدِم وقاعدة البيانات على النحو الآتي:

1. **واجهة المُستخدِم:**
   - أ. إنشاء زِرٍّ يُمكِن للمُستخدِم أنْ يضغط عليه لإضافة مهمة جديدة.
   - ب. إنشاء صفحة لعرض قائمة المهام فيها.
   - ج. إمكانية التعديل على المهام أو حذفها من صفحة قائمة المهام.

2- قاعدة البيانات:

- أ. إنشاء جدول للمهام يحوي عنوان المهمة، ووصفًا لها، وحالتها، والأولوية، والموعد النهائي.
- ب. رسم المُخطَّطات اللازمة لقاعدة البيانات.

نشاط فردي

أُفكِّر في طريقة لتصميم واجهة المُستخدِم في نظام حجز المواعيد في عيادة الأسنان، وآليَّة عمل نظام حجز للمواعيد في العيادة.

أُدوِّن الأفكار التي أتوصَّل إليها، وأرسم المُخطَّطات اللازمة للنظام، ثمَّ أُناقِشها مع زملائي/ زميلاتي في الصف.

المرحلة الرابعة: التطوير (Development)

تمتاز هذه المرحلة بتحويل مُخرَجات مرحلة التصميم إلى صيغة برمجية يُمكِن استخدامها بصورة عملية؛ إذ تتضمَّن هذه المرحلة كتابة الكود (المقاطع البرمجية) الذي يختصُّ بالنظام، ويكون قابلًا للتطوير. كذلك تتضمَّن هذه المرحلة مراجعة المقاطع البرمجية، والعمل على تحسين النظام باستمرار.

نشاط عملي

أُحوِّل مُخرَجات المرحلة السابقة في نظام حجز المواعيد الخاص بعيادة الأسنان إلى برنامج باستخدام لغة البرمجة بايثون (Python)، ثمَّ أُطلِع الزملاء/ الزميلات على البرنامج، ونتبادل معًا الأفكار والمُقترَحات لتحسينه.

المرحلة الخامسة: الاختبار (Testing)

يتمُّ في هذه المرحلة اختبار النظام يدويًّا وآليًّا من قِبَل مُطوِّر النظام؛ للتأكُّد أنَّه يعمل بصورة صحيحة، وأنَّه يُحقِّق الهدف الذي أُنشِئ من أجله.

تهدف هذه المرحلة إلى الحصول على التغذية الراجعة من قبل مُطوِّر النظام والمُستخدِمين ومالك النظام؛ بُغْيَةَ تصحيح الأخطاء (إنْ وُجِدت) والتطوير والتحسين.

نشاط عملي

أختبر البرنامج الخاص بحجز المواعيد في عيادة الأسنان، وذلك بتنفيذه، والتحقُّق من مُخرَجاته. بعد ذلك أُدوِّن ملاحظاتي على البرنامج، ثمَّ أُناقِشها مع الزملاء/ الزميلات بهدف تحسين النظام.

نشاط جماعي

أُنظِّم جلسة نقاش مع زملائي/ زميلاتي في المجموعة للإجابة عن الأسئلة الآتية:

- ما الإجراءات الواجب اتِّخاذها إذا أخفق النظام في أداء المهام المنوطة به بصورة صحيحة؟
- فيمَ يستفاد من الوثائق التي جُمِّعت في المراحل السابقة؟
- كيف يُمكِن الاستفادة من جميع العمليات السابقة في تحسين النظام؟

أُناقِش إجابات الأسئلة مع زملائي/ زميلاتي في المجموعة، ثمَّ أعرضها أمام أفراد المجموعات الأُخرى بهدف التوصُّل إلى استنتاجات مُشترَكة.

المرحلة السادسة: النشر والصيانة (Publishing and Maintenance)

تتضمَّن هذه المرحلة نشر النظام؛ أيْ جعله متاحًا للاستخدام في بيئة حقيقية، ويكون ذلك ضمن عدد من المراحل؛ للتأكُّد أنَّ النظام يعمل بسلاسة وسهولة ويُسْر على النحو المُخطَّط له. كذلك تتضمَّن هذه المرحلة إخضاع النظام للصيانة الدورية؛ لضمان تنفيذه جميع المهام المنوطة به بصورة صحيحة.

إضاءة

توجد ثلاث طرائق لاستخدام النظام، هي:

1. **الاستخدام المباشر**: تمتاز هذه الطريقة بالتحوُّل إلى النظام الجديد مباشرة، وإلغاء النظام القديم.
2. **الاستخدام المُتزامِن**: تمتاز هذه الطريقة باستخدام النظام الجديد، جنبًا إلى جنب مع النظام القديم؛ للتأكُّد أنَّه يُنفِّذ جميع المهام المنوطة به على النحو الصحيح.
3. **الاستخدام المرحلي المُتدرِّج**: تُستخدَم هذه الطريقة إذا كان النظام ضخمًا وكبيرًا، وذلك باستعمال نظام فرعي جديد فقط، والإبقاء على بقيَّة الأنظمة الفرعية مع النظام القديم لحين التحقُّق من صلاحيته، ثمَّ يتمُّ الانتقال إلى نظام فرعي آخر، وهكذا.

نشاط جماعي

يُعَدُّ التوثيق والتدريب والدعم من أهمِّ العمليات التي تضمن استمرارية عمل النظام.

أُناقِش - بالتعاون مع أفراد مجموعتي - المراحل التي يُمكِن أنْ تحدث فيها عمليات التوثيق والتدريب والدعم، ثمَّ أُشارِك أفراد المجموعات الأُخرى في ما توصَّلنا إليه من نتائج.

نشاط فردي

أُطبِّق مراحل تطوير النظام على نظام بسيط لتسجيل الطلبة في إحدى المدارس، وأستخدم برمجية سكراتش (Scratch) أو برمجية بايثون (Python) في تطوير البرنامج الخاص بالنظام. بعد ذلك أُدوِّن الخطوات التي استخدمتها في ذلك، ثمَّ أُشارِك أفراد المجموعات الأُخرى في ما توصَّلْتُ إليه من خلال مجموعتي.

النماذج الخاصة بدورة حياة تطوير النظام (SDLC Models):

يوجد لدورة حياة تطوير النظام كثير من النماذج التي تختلف في ما بينها من حيث آلية التطبيق، والإيجابيات والسلبيات. وفي ما يأتي أهمُّ هذه النماذج:

1- نموذج الشلّال (Waterfall):

يُعَدُّ هذا النموذج الأساس لبقيَّة النماذج؛ ذلك أنَّه يتكوَّن من مراحل بسيطة أساسية، وهو يُناسِب المشروعات التي تكون فيها المُتطلَّبات واضحة ومُحدَّدة، لكنَّه لم يعد مُستخدَمًا اليوم بسبب عدم مرونته.

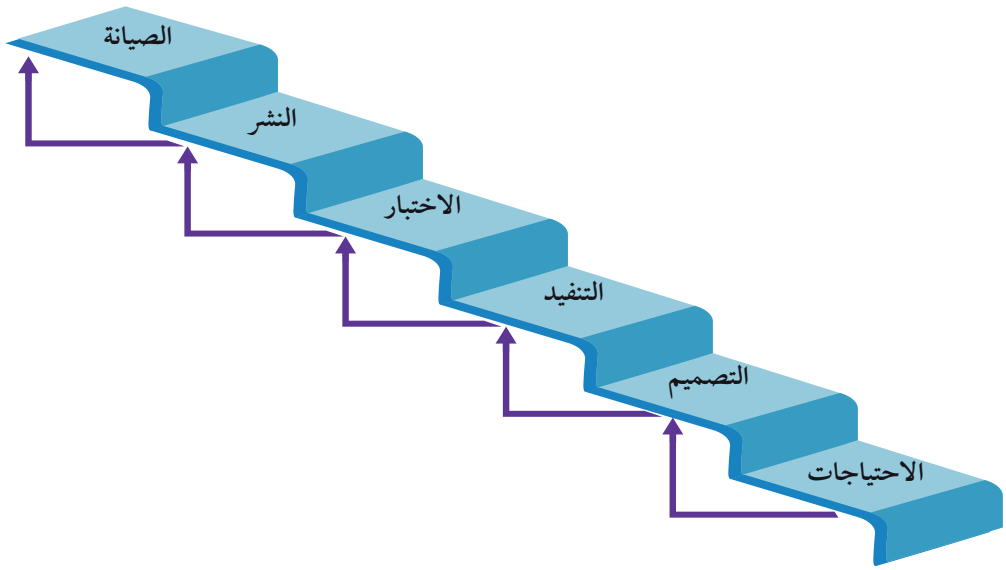
مزايا نموذج الشلّال (Waterfall):

يُمكِن إجمال مزايا نموذج الشلّال في ما يأتي:

1. **التسلسل الخَطِّي**: يمتاز نموذج الشلّال بأنَّه خَطِّي ومُتسلسِل؛ إذ تنتهي كل خطوة بصورة كاملة قبل الانتقال إلى الخطوة التي تليها.
2. **التوثيق الجيِّد**: يراعى في نموذج الشلّال التوثيق الجيِّد لجميع مراحل المشروع؛ لضمان سَيْر العمل بصورة صحيحة، وتمكين الفريق من تحقيق أهداف المشروع على نحوٍ واضح ودقيق.
3. **الجودة العالية**: تخضع جميع مراحل المشروع في نموذج الشلّال لاختبارات خاصة، فضلًا عن ملاحظة جودة كل مرحلة ومتابعتها؛ للتأكُّد أنَّها تعمل وَفق ما هو مُخطَّط له.
4. **التخطيط الدقيق**: يرتكز نموذج الشلّال في عمله على التخطيط الدقيق للمشروع، ويشمل ذلك تحديد نطاق المشروع، والجداول الزمنية، والمُنتَجات النهائية التي تخضع للمراقبة والمتابعة طوال دورة حياة المشروع.

مراحل نموذج الشلّال (Waterfall):

يتألَّف نموذج الشلّال من ست مراحل يُبيِّنها الشكل (1-3).



الشكل (1-3): مراحل نموذج الشلّال (Waterfall).

مُحدِّدات استخدام نموذج الشلّال (Waterfall):

1. **الجمود وعدم المرونة**: ليس من السهل في نموذج الشلّال إجراء تغييرات على المُتطلَّبات أو التصميم بعد الانتهاء من المرحلة ذات الصلة.
2. **عدم ملاءمة المشروعات الديناميكية**: لا يُناسِب نموذج الشلّال المشروعات التي قد تتغيَّر فيها المُتطلَّبات بمرور الوقت، وتُكتشَف فيها الأخطاء مُتأخِّرًا.
3. **التأخُّر في تقديم المُنتَج النهائي**: لا يُمكِن للمُستخدِمين الحصول على أيِّ نتائج إلّا بعد انتهاء جميع المراحل.
4. **صعوبة التعامل مع المشروعات الكبيرة والمشروعات المُعقَّدة**: قد يؤدّي استخدام نموذج الشلّال في المشروعات الكبيرة والمشروعات المُعقَّدة إلى ظهور مشكلات في عمليتي التتبُّع والتنفيذ إذا كانت التفاصيل كثيرة ومُتشعِّبة.

أُناقِش

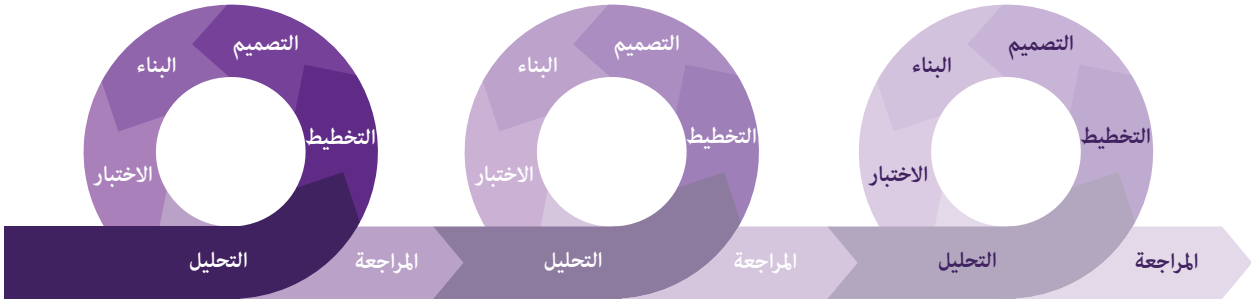
أُناقِش أفراد مجموعتي في الحالات التي يُمكِن فيها استخدام نموذج الشلّال، مُبيِّنين أسباب ذلك، ثمَّ نعرض ما نتوصَّل إليه من نتائج على أفراد المجموعات الأُخرى، ونعمل على مناقشتها معًا.

2- النموذج الرشيق (Agile):

يُقسَّم المشروع بحسب هذا النموذج إلى مجموعة من التكرارات (أو المراحل) الصغيرة التي تُسمّى (Iteration) أو (Sprint). وهذه التكرارات تتبع مسارًا خَطِّيًّا، وفيها ينتهي كل تكرار (أو مرحلة) بانتهاء المُدَّة الزمنية المُحدَّدة له، التي قد تمتدُّ من أسبوع واحد إلى أربعة أسابيع، لتبدأ بعدها مراجعة ما تحقَّق من مُتطلَّبات، وتلقّي التغذية الراجعة؛ لتعرُّف ما يجب إدخاله من تعديلات في المشروع أوَّلًا بأوَّل، وتحديد المُتطلَّبات التي يتعيَّن نقلها إلى المرحلة التالية؛ ما يُسهِم في منح النظام مرونة أكثر عند تطويره، ويجعله أكثر قدرة على التكيُّف مع المُتغيِّرات. ولهذا يُطلَق على النموذج الرشيق اسم (Change Driven)؛ أيْ مُوجَّه بالتغيير، فهو يُستخدَم في الحالات التي تتطلَّب تكيُّفًا سريعًا مع كل مُتغيِّر، علمًا بأنَّ لهذا النموذج العديد من المنهجيات التي تقوم جميعها على المبدأ نفسه بالرغم من وجود اختلافات بسيطة في ما بينها.

مراحل النموذج الرشيق (Agile):

يَمُرُّ كل تكرار (Sprint) في هذا النموذج بخمس مراحل، هي: البناء، والتصميم، والتخطيط، والتحليل، والاختبار. أنظر الشكل (1-4).



الشكل (1-4): مراحل النموذج الرشيق (Agile)، وخطوات كل مرحلة.

مبادئ النموذج الرشيق (Agile):

يعتمد هذا النموذج على مجموعة من المبادئ التي تُعزِّز التعاون والتطوير المستمر؛ لضمان تحقيق جميع مُتطلَّبات العميل واحتياجاته بفاعلية وسرعة. ومن أبرز هذه المبادئ:

- **الحفاظ على اتِّصال وثيق بالعميل**، والحرص على إشراكه في كل مرحلة من المراحل؛ لضمان فهم واضح لمُتطلَّباته واحتياجاته، والوقوف على مستوى التقدُّم في سَيْر العمل، وإعادة تقييم المُتطلَّبات والاحتياجات.
- **التوجُّه نحو نشر البرامج بصورة مُتكرِّرة بدلًا من الاعتماد على التوثيق الشامل**، وتسليم إصدارات مُتزايِدة خلال مُدَد زمنية قصيرة (بضعة أسابيع).
- **وجوب استخدام فِرَق عمل يَتَّصِف أفرادها بالتميُّز والكفاءة والقدرة على التواصل الفاعل في ما بينهم**، إضافةً إلى عقد اجتماعات دورية؛ لمناقشة مستوى التقدُّم الذي تحقَّق، وضمان التنسيق بين الأطراف جميعًا.

خصائص العمليات في النموذج الرشيق (Agile):

تمتاز العمليات في هذا النموذج بخصائص عِدَّة، أبرزها:

- **المرونة:** يُمكِن للعمليات في النموذج الرشيق أنْ تتكيَّف مع المُتغيِّرات الفنية والمُتغيِّرات البيئية للنظام.
- **التطوير التدريجي:** تخضع العمليات في النموذج الرشيق للتطوُّر والتحسين بصورة تدريجية مستمرة.
- **التفاعل مع العميل:** تتيح العمليات في النموذج الرشيق استخدام ملاحظات العميل في تعديل النظام البرمجي وَفقًا لمُتطلَّباته واحتياجاته.
- **السرعة:** تُسلَّم التعديلات الخاصة بالعمليات في وقت قصير لتحقيق قيمة مضافة بسرعة.

أُناقِش

أُناقِش زملائي/ زميلاتي في الحالات التي يُمكِن فيها استخدام النموذج الرشيق، مُبيِّنين أسباب ذلك، ثمَّ نبحث معًا عن إجابات للأسئلة الآتية:

- هل يُمكِن تطبيق النموذج الرشيق على جميع أنواع المشروعات؟
- ما التحدِّيات والمُعوِّقات التي قد تَحول دون استخدام هذا النموذج في المشروعات الكبيرة والمشروعات المُعقَّدة؟
- كيف يُمكِن ضمان الاستفادة الفاعلة من هذا النموذج في حال كان تفاعل العملاء محدودًا؟

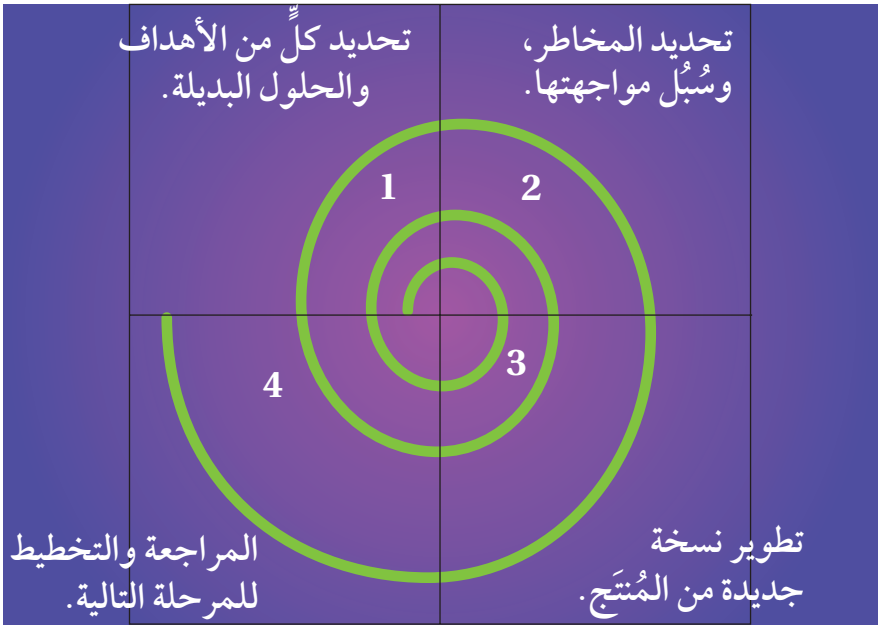
أُشارِك زملائي/ زميلاتي في آرائي المُتعلِّقة بهذا النموذج، وأستمع إلى وِجهات نظرهم المختلفة، مُقدِّمين أمثلة واقعية لدعم النقاش وإثرائه.

3- نموذج الحلزون (Spiral):

يُعَدُّ هذا النموذج واحدًا من أشهر النماذج المُستخدَمة في المشروعات، وهو يمتاز بشكله الحلزوني، ووجود حلقات مُتعدِّدة فيه، يختلف عددها من مشروع إلى آخر، وتُمثِّل كلٌّ منها مرحلة من مراحل النموذج. يتبع نموذج الحلزون نهجًا تكراريًّا كما هو الحال في النموذج الرشيق (Agile)، لكنَّه يُوفِّر - في الوقت نفسه- نهجًا مُنظَّمًا لإدارة المخاطر في المشروعات المُعقَّدة التي تتطلَّب دراسة دقيقة للمخاطر التقنية والمخاطر التشغيلية.

مراحل نموذج الحلزون (Spiral):

يُبيِّن الشكل (1-5) المراحل التي يَمُرُّ بها نموذج الحلزون.



الشكل (1-5): مراحل تطوير البرمجية باستخدام نموذج الحلزون (Spiral).

مواجهة المخاطر في نموذج الحلزون (Spiral):

تُعرَّف المخاطرة بأنَّها أيُّ عامل قد يُؤثِّر سلبًا في نجاح المشروع.

تُستخدَم في نموذج الحلزون استراتيجية فاعلة للتعامل مع المخاطر ومواجهتها؛ لضمان نجاح المشروع. وتتمثَّل أهمية هذه الاستراتيجية في ما يأتي:

1. **التحديد المستمر للمخاطر**: يُمكِن تحديد المخاطر المُحتمَلة في كل مرحلة من مراحل النموذج بصورة دورية؛ ما يساعد على اتِّخاذ الإجراءات اللازمة لمعالجة هذه المخاطر في الوقت المناسب.
2. **توفير النماذج الأوَّلية**: يتمثَّل ذلك في إنشاء نموذج أوَّلي لكل مرحلة من مراحل تطوير البرمجيات؛ ما يتيح الكشف المُبكِّر عن المشكلات والمخاطر والمُعوِّقات، وإيجاد الحلول المناسبة لها قبل الانتقال إلى المرحلة التالية.
3. **تحييد المخاطر المعروفة مُسبَقًا**: تتيح هذه الاستراتيجية التعامل الفاعل مع المخاطر التي حُدِّدت قبل البَدْء بتطوير البرمجيات، علمًا بأنَّ النماذج الأوَّلية قد لا تكون كافية لمواجهة المخاطر غير المُتوقَّعة التي قد تظهر أثناء عملية التنفيذ.

يُعَدُّ نموذج الحلزون (Spiral) مناسبًا للاستخدام في الحالات الآتية:

1. **المشروعات الضخمة**: يُستخدَم نموذج الحلزون في المشروعات الكبيرة التي تتطلَّب تخطيطًا وتنفيذًا دقيقين، واختبارًا مستمرًّا للمخاطر في مراحل مُتعدِّدة.
2. **الإصدارات المُتكرِّرة**: يُستخدَم نموذج الحلزون عند الحاجة إلى إصدار نسخ مُتكرِّرة من المُنتَج بصورة دورية.
3. **النماذج الأوَّلية**: تتطلَّب عملية تطوير البرمجيات أحيانًا إنشاء نموذج أوَّلي؛ لفهم المُتطلَّبات اللازمة، أو تقييم الحلول المُمكِنة.
4. **تقييم المخاطر**: قد يكون تحليل المخاطر وتقييمها جزءًا أساسيًّا من عملية التطوير، كما في المشروعات العسكرية، والمشروعات المالية، والمشروعات الطبية.
5. **المُتطلَّبات المُعقَّدة والغامضة**: يُفضَّل استخدام نموذج الحلزون في المشروعات ذات المُتطلَّبات غير الواضحة أو المُعقَّدة التي قد تتغيَّر مع تقدُّم سَيْر العمل في المشروع.

نشاط جماعي

في ما يأتي مجموعة من الحالات العملية لمشروعات مختلفة. أقرأ كل حالة بتدبُّر ورَوِيَّة، ثمَّ أُحلِّلها بالتعاون مع أفراد مجموعتي؛ لتحديد النموذج الأنسب لتطوير النظام مع التعليل:

1- شركة مُتخصِّصة في تطوير البرمجيات تعمل على تنفيذ مشروع يتضمَّن تصميم نظام لإدارة الامتحانات المدرسية، علمًا بأنَّ مُتطلَّبات النظام واضحة وثابتة منذ بَدْء المشروع، ولا يُتوقَّع تغييرها أثناء عملية التطوير.

2- فريق مُتخصِّص في تطوير تطبيقات الهواتف الذكية يعمل على إنشاء تطبيق للتسوُّق الإلكتروني، وهو مشروع يتطلَّب تكرارًا مستمرًّا لتجربة الوظائف الجديدة بناءً على آراء المُستخدِمين الذين يختبرون التطبيق في كل مرحلة من مراحل التطوير.

3- شركة طيران ترغب في تطوير نظام لحجز تذاكر الطيران يَتَّسِم بالتعقيد الكبير، ويشمل العديد من المخاطر المُتعلِّقة بالأمان والدِّقَّة. وهذا المشروع يتطلَّب تقييم المخاطر وتذليلها بصورة مُتكرِّرة قبل الانتقال إلى المراحل التالية.

4- شركة ناشئة تعمل على تطوير مُنتَج تجريبي جديد بناءً على أفكار غير مُكتمِلة، مع توقُّع حدوث تغييرات مُتكرِّرة بناءً على ملاحظات العملاء عند الاستخدام.

أُناقِش

أُناقش أفراد مجموعتي في النموذج المختار، وفي أسباب اختيار هذا النموذج دون غيره من النماذج، ثمَّ أعرض النتائج التي توصَّلنا إليها في المجموعة أمام مُعلِّمي/ مُعلِّمتي وزملائي/ زميلاتي لمناقشتها معًا.

إضاءة

يُطلَق على نموذج الحلزون (Spiral) اسم النموذج الفوقي (Meta Model)؛ لأنَّه يضمُّ جميع النماذج الأُخرى لدورة حياة تطوير النظام.

المواطنة الرقمية:

- **الثقافة الرقمية**: أُطبِّق ما أتعلَّمه من مهارات رقمية في حياتي اليومية؛ بأنْ أنظر إلى أيِّ عمل أقوم به على أساس أنَّه مشروع ينبغي تحليله، ثمَّ التخطيط له جيِّدًا، ثمَّ إيجاد الحلول المناسبة له.
- **الأمان الرقمي** (Digital Security): أحرص على حماية البيانات والمعلومات في مرحلتي التحليل والتصميم، وأتحقَّق من أمان الكود في مرحلة التنفيذ.
- **احترام حقوق الآخرين الرقمية**: ألتزم بقوانين حماية الملكية الفكرية عند جمع مُتطلَّبات النظام، ولا أستخدم أيَّة موارد أو أفكار من دون إذن.
- **المسؤولية الرقمية** (Digital Responsibility): أحرص على توثيق العمل، وأُؤكِّد الالتزام بالمعايير الأخلاقية.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أُوضِّح المقصود بدورة حياة تطوير النظام.

السؤال الثاني: أملأ الفراغ بالمصطلح المناسب في كلٍّ من الجمل الآتية:

1. يُطلَق على المرحلة التي تُحدَّد فيها أهداف المشروع اسم..... .
2. يُطلَق على المرحلة التي تُصمَّم فيها واجهة المُستخدِم وقواعد البيانات للنظام اسم ..... .
3. .....: مرحلة يتمُّ فيها التأكُّد من عمل النظام بصورة صحيحة، وتحقيق الهدف الذي أُنشِئ من أجله.
4. النموذج الذي يعتمد على التخطيط الجيِّد يُسمّى ..... .
5. .....: نموذج يُعْنى بالتعامل مع المخاطر.
6. يُطلَق على النموذج الذي يُعْنى بالاتِّصال الوثيق مع العميل اسم..... .

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُقارِن بين نموذج الشلّال والنموذج الرشيق ونموذج الحلزون من حيث المزايا، والعيوب، والمخاطر.

السؤال الثاني: أبحث عن أهمِّ منهجيتين من منهجيات النموذج الرشيق (Agile)، وأذكر أوجه الاختلاف بينهما.

السؤال الثالث: أُطبِّق الخطوات والمراحل الخاصة بالنموذج الرشيق (Agile) على مشروع إدارة مكتبة إلكترونية.

السؤال الرابع: لديَّ مشروع لتطوير موقع إلكتروني. أُبيِّن كيف يُمكِن تنفيذ النظام الخاص لهذا المشروع باستخدام نموذج الشلّال (Waterfall) ونموذج الحلزون (Spiral)، ثمَّ أُحدِّد أيُّ النموذجين أنسب لهذه الحالة، وأُبرِّر إجابتي.

السؤال الخامس: في ما يأتي مجموعة من الحالات العملية لمشروعات مختلفة. أقرأ كل حالة بتدبُّر ورَوِيَّة، ثمَّ أُحدِّد النموذج الأنسب من النماذج الثلاثة (الشلّال، الرشيق، الحلزون) لتطوير كل مشروع، وأُبرِّر إجابتي:

1. مشروع تصميم نظام لإدارة الموارد البشرية في شركة صغيرة، علمًا بأنَّ مُتطلَّبات النظام واضحة وثابتة منذ بَدْء المشروع، ولا يُتوقَّع تغييرها أثناء عملية التطوير.
2. مشروع تطوير لعبة إلكترونية مُبتكَرة، وهو مشروع يتطلَّب إضافات مُتكرِّرة بناءً على تجربة المُستخدِمين في كل مرحلة من مراحل التطوير.
3. مشروع تصميم نظام لإدارة العمليات المصرفية، علمًا بأنَّ النظام مُعقَّد جدًّا، وفيه مخاطر تتعلَّق بالأمان والدِّقَّة، ومن ثَمَّ يتطلَّب المشروع تقييمًا مستمرًّا للمخاطر قبل الانتقال إلى المراحل التالية.
4. مشروع تطوير تطبيق للهواتف الذكية خاص بتتبُّع اللياقة البدنية. وهذا التطبيق يتطلَّب إدخال تحسينات مستمرة بناءً على آراء المُستخدِمين الذين يختبرون كل إصدار منه.
5. مشروع إنشاء بوّابة إلكترونية حكومية، وهو مشروع يتطلَّب تخطيطًا وتنفيذًا مُتسلسِلًا، إضافةً إلى تحديد المُتطلَّبات منذ بَدْء تنفيذه، وهي مُتطلَّبات لا يُتوقَّع تغييرها أثناء عملية التطوير.
6. مشروع إطلاق مِنصَّة تعليم إلكتروني تفاعلية، وهو مشروع يتطلَّب إجراء تجارب عديدة وإصدار نسخ مُتعدِّدة منه بصورة تدريجية؛ للوفاء باحتياجات المُستخدِمين، إضافةً إلى إمكانية تعديل التصميم والوظائف والمهام بناءً على ملاحظات المُستخدِمين.

الدرس الثاني

الخوارزميات (Algorithms)

الفكرة الرئيسة:

سأتعلَّم في هذا الدرس كيف أكتب الخوارزمية شبه الرمزية، ورسم مُخطَّطات سَيْر العمليات للتعبير عن مسألة ما، وتتبُّع نواتج خوارزمية مكتوبة مُسبَقًا.

المفاهيم والمصطلحات:

الخوارزمية شبه الرمزية (Pseudocode)، مُخطَّط سَيْر العمليات (Flowchart).

نتاجات التعلُّم (Learning Outcomes):

- أُمثِّل الخوارزميات المُركَّبة باستخدام مُخطَّطات سَيْر العمليات (Flow Charts).
- أُمثِّل الخوارزميات المُركَّبة باستخدام الخوارزمية شبه الرمزية (Pseudocode).
- أُطبِّق الخوارزميات المُقترَحة على مجموعة مُتنوِّعة من المُدخَلات؛ لاختبار أدائها، وتحليل نتائجها.
- أُعدِّل الخوارزميات؛ لتحسين أدائها، وضمان تحقيقها للأهداف المنشودة بناءً على نتائج اختبارها وتحليلها.
- أُقيِّم الخوارزميات بناءً على معايير مُحدَّدة بدِقَّة.
- أختار أنسب خوارزمية لحَلِّ مشكلة مُحدَّدة بناءً على وضوحها ودِقَّتها وفاعليتها في التوصُّل إلى الحَلِّ المطلوب.

تعرَّفْتُ سابقًا أنَّ برنامج الحاسوب هو مجموعة من الأوامر التي تُكتَب بإحدى لغات البرمجة؛ بُغْيَةَ حَلِّ مشكلة ما، أو أداء مهمة مُحدَّدة باستخدام جهاز الحاسوب. تعرَّفْتُ أيضًا أنَّ مجموعة الخطوات المُتسلسِلة والمنطقية التي توصِل إلى حَلِّ المسألة تُسمّى الخوارزمية، وأنَّه يُمكِن وصفها عن طريق الخوارزمية شبه الرمزية (Pseudocode)، أو مُخطَّط سَيْر العمليات (Flowchart).

نشاط تمهيدي

توجد أشكال قياسية لمُخطَّطات سَيْر العمليات، ويدلُّ كل رمز فيها على عملية مُعيَّنة. في ما يأتي مجموعة من العمليات، أرسم الرمز المناسب لكلٍّ منها باستخدام برنامج حاسوبي:

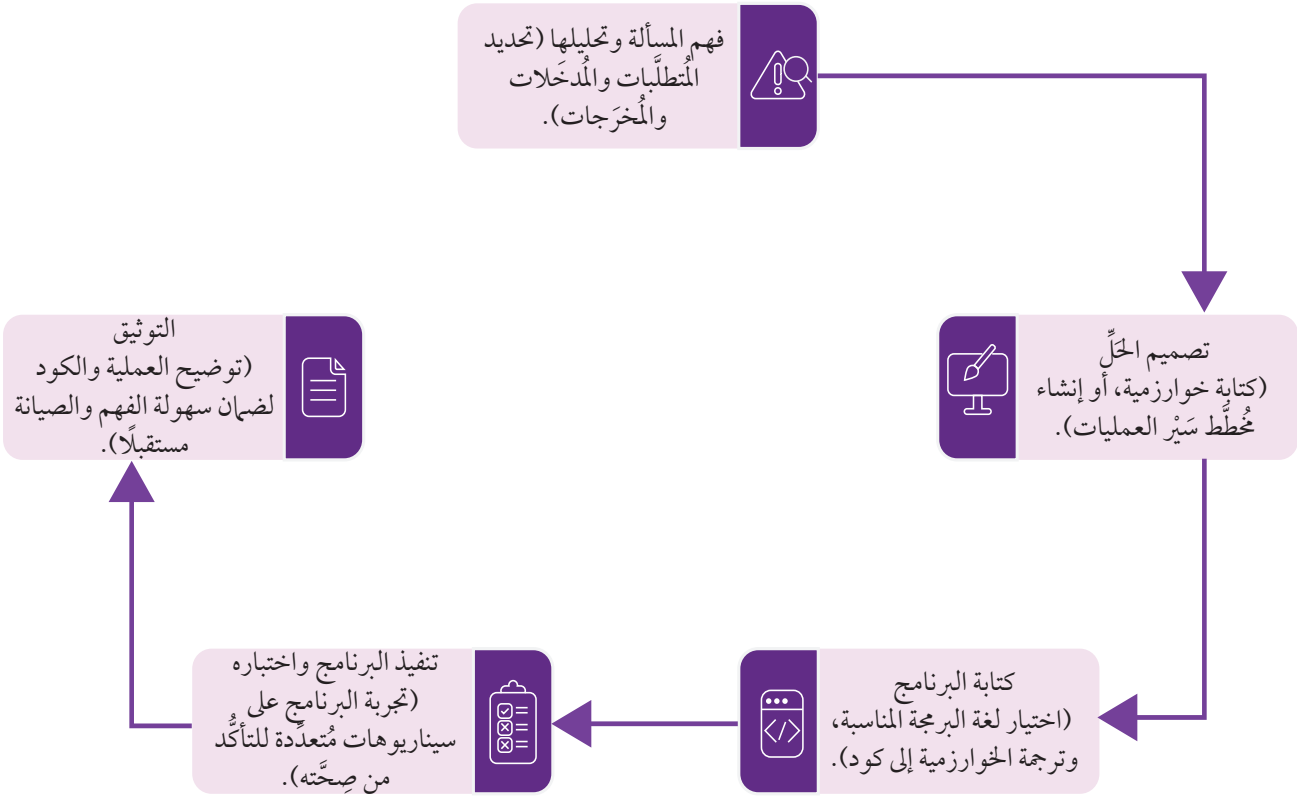
- تمثيل عملية المعالجة.
- تمثيل قرار أو شرط.
- تمثيل نقطة البداية أو نقطة النهاية في المُخطَّط.
- نقطة اتِّصال أو نقطة مرجعية داخل المُخطَّط.

أُناقِش زملائي/ زميلاتي في إجابتي، ونتبادل معًا الخبرات للتحقُّق من صِحَّة الحَلِّ.

حَلُّ المسألة (Problem Solving)

تُعرَّف المسألة (المشكلة) بأنَّها الناتج أو الهدف الذي يُراد الوصول إليه باتِّباع مجموعة من الخطوات التي قد تتطلَّب إجراء عمليات حسابية أو عمليات منطقية. لا شكَّ أنَّه يُمكِن التوصُّل إلى الناتج بأكثر من طريقة صحيحة، لكنَّ بعض هذه الطرائق تُعَدُّ أفضل من غيرها بناءً على الوقت المُستغرَق في عملية الحَلِّ، واستخدام الحَدِّ الأدنى من الموارد. وهذا كله يعتمد على كفاءة الشخص الذي يتولّى حَلَّ المسألة، ومدى فهمه لها.

يُبيِّن الشكل (2-1) الخطوات الأساسية التي يُمكِن اتِّباعها عند حَلِّ أيِّ مسألة برمجية.



الشكل (2-1): خطوات حَلِّ المسألة البرمجية.

في ما يأتي شرح مُفصَّل لكل خطوة من خطوات حَلِّ المسألة، مدعومًا بأمثلة عملية لتوضيح طريقة الحَلِّ:

الخطوة الأولى: فهم المسألة وتحليلها.

تتضمَّن هذه الخطوة تحديد المسألة، وتحديد المُدخَلات والمُخرَجات وطرائق المعالجة المختلفة لها.

مثال:

أُحلِّل المسألة الآتية:

ما حاصل ضرب العدد A في العدد B؟

الحَلُّ:

تحديد المسألة: إيجاد ناتج ضرب العددين A،B.

تحليل المسألة:

المُدخَلات: العدد A، والعدد B.

المُخرَجات: حاصل ضرب العددين، وتخزين الناتج في المُتغيِّر C.

طريقة المعالجة: C = A * B

نشاط جماعي

أُحلِّل – بالتعاون مع أفراد مجموعتي – كل مسألة من المسائل الآتية:

1. إيجاد المُتوسِّط الحسابي لثلاثة أعداد.
2. إدخال اسم العميل، ورقم حسابه، وفرع البنك الذي فَتحَ فيه حسابه، وطباعة هذه البيانات جميعها.
3. إيجاد ناتج قسمة خمسة أعداد على العدد 3.

الخطوة الثانية: كتابة الخوارزمية.

تتضمَّن هذه الخطوة تمثيل المسألة بطريقة مُنظَّمة ومنطقية باستخدام الخوارزمية. ويُمكِن في هذه الحالة استخدام خوارزمية شبه رمزية (Pseudocode)، أو مُخطَّط سَيْر العمليات (Flowchart).

مثال:

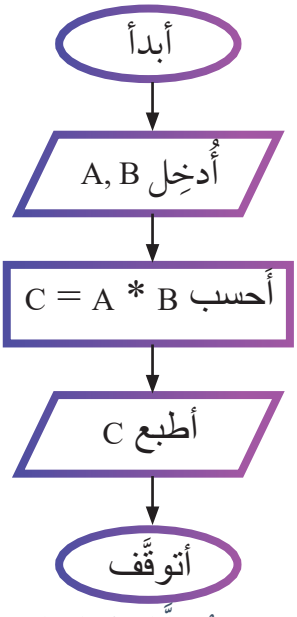
أكتب الخوارزمية شبه الرمزية للمسألة في المثال السابق، ثمَّ أرسم مُخطَّط سَيْر العمليات لها.

الحَلُّ:

- **الخوارزمية شبه الرمزية:**

1. أبدأ
2. أُدخِل A, B
3. أُحسب C = A * B
4. أطبع C
5. أتوقَّف

- **مُخطَّط سَيْر العمليات كما يظهر في الشكل (2-2):**



الشكل (2-2): مُخطَّط سَيْر العمليات لمسألة حاصل ضرب عددين.

أُناقِش

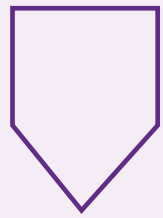
أتعاون مع أفراد مجموعتي على الإجابة عن الأسئلة الآتية:

1. **أيُّهما أسهل:** كتابة الخوارزمية شبه الرمزية أم رسم مُخطَّط سَيْر العمليات؟ أُبرِّر إجابتي.
   هل يُمكِن إجراء تعديلات على الخوارزمية السابقة؟
2. إذا كانت إجابة السؤال السابق بالإيجاب، فما هذه التعديلات؟
3. ما تأثير هذه التعديلات في حَلِّ المسألة؟

إثراء

يُبيِّن الجدول (2-1) رموزًا قياسيةً إضافيةً تُستخدَم في رسم مُخطَّطات سَيْر العمليات؛ ما يساعد على تقديم تمثيل أكثر شمولًا ووضوحًا للخوارزمية.

الجدول (2-1): رموز قياسية تُستخدَم في رسم مُخطَّطات سَيْر العمليات.

الرمز	الاستخدام
	التعليق: يُستخدَم هذا الرمز في كتابة ملاحظات أو تعليقات لتوضيح إحدى العمليات أو شيء ما.
	مستند إدخال/ إخراج: يُستخدَم هذا الرمز عند الحاجة إلى استيراد مُدخَلات من أحد الملفات أو تصدير مُخرَجات إلى ملف ما.
	**رابط الصفحات (Off- Page Connector):** يُستخدَم هذا الرمز في ربط الأجزاء المختلفة من مُخطَّط سَيْر العمليات عندما تكون هذه الأجزاء في صفحات مُنفصِلة.

يُمكِن استخدام تطبيقات ومواقع إلكترونية في رسم مُخطَّط سَيْر العمليات بدلًا من رسمه يدويًّا.
ومن هذه المواقع والتطبيقات: (Cacoo)، و(Miro)، و(Creately)، و(Draw.io)، و(Microsoft Visio)، و(Lucidchart)، علمًا بأنَّ بعضها مجّاني، وبعضها الآخر محدود الاستخدام للحساب المجّاني.

الخطوة الثالثة: كتابة البرنامج باستخدام إحدى لغات البرمجة.

بعد تمثيل الخوارزمية، تُترجَم خطواتها إلى جمل برمجية حقيقية باستخدام لغات البرمجة المختلفة،
مثل: لغة البرمجة جافا (Java)، ولغة البرمجة سي ++ (C++)، ولغة البرمجة بايثون (Python).

مثال:

أُترجِم الخوارزمية في المثال السابق إلى لغة البرمجة بايثون (Python).

الحَلُّ: يُبيِّن الشكل (2-3) الخوارزمية بعد ترجمتها إلى لغة البرمجة بايثون (Python).

```
A = int(input("Enter the first number: "))
B = int(input("Enter the second number: "))
C = A * B

print("The product of A * B =", C)
```

الشكل (2-3): برنامج كُتِب بلغة البرمجة بايثون لإيجاد حاصل ضرب عددين.

الخطوة الرابعة: تنفيذ البرنامج واختبار صِحَّته.

تتضمَّن هذه الخطوة تنفيذ البرنامج باستخدام جهاز الحاسوب؛ بُغْيَةَ اختباره، والتحقُّق من صِحَّته، وهو ما يتطلَّب إدخال معطيات مختلفة تخصُّ (تُغطّي) الحالات المطلوبة جميعها للتأكُّد أنَّ البرنامج يعمل بصورة صحيحة، وأنَّه يُقدِّم النتائج المُتوقَّعة في كل مَرَّة. يتمثَّل الهدف الأساسي من ذلك في اكتشاف أيَّة أخطاء أو تحسين الأداء؛ لضمان تحقيق الأهداف المنشودة من البرنامج.

مثال:

أُنفِّذ البرنامج السابق باستخدام أعداد أعرف مُسبَقًا ناتج ضربها (يُمكِن استخدام الآلة الحاسبة للتحقُّق من الإجابة الصحيحة)، ثمَّ أختبر صِحَّة البرنامج.

الحَلُّ:

أُنفِّذ البرنامج بإدخال العدد 4 والعدد 5:

```
Enter the first number: 4
Enter the second number: 5
The product of A * B = 20
```

أُلاحِظ من التطبيق العملي أنَّ الناتج صحيح، وهو 20.

أُنفِّذ البرنامج مَرَّة أُخرى بإدخال العدد 3- والعدد 7:

```
Enter the first number: -3
Enter the second number: 7
The product of A * B = -21
```

أُلاحِظ من التطبيق العملي أنَّ الناتج صحيح، وهو 21-.

ملحوظة: يُمكِن تنفيذ البرنامج مَرّات أُخرى للتحقُّق من صِحَّته باستخدام معطيات جديدة وحالة مختلفة، مثل: إدخال عددين سالبين.

الخطوة الخامسة: التوثيق.

تُعَدُّ مرحلة التوثيق خطوة أساسية في جميع مراحل حَلِّ المسألة منذ البَدْء بعملية الحَلِّ؛ لِما تُمثِّله من أهمية كبيرة في تسهيل فهم البرنامج، وإجراء التعديلات اللازمة عليه مستقبلًا. وتشمل هذه الخطوة كتابة تقرير يُبيِّن تفاصيل البرنامج، ومبدأ عمله، وكيفية تشغيله، إضافةً إلى جمل التعليقات داخل الكود، والتوثيق الخارجي (مثل دليل المُستخدِم).

مثال:

أُوثِّق البرنامج في المثال السابق، وذلك بإضافة جمل التعليقات في كل خطوة كما في الشكل (2-4).

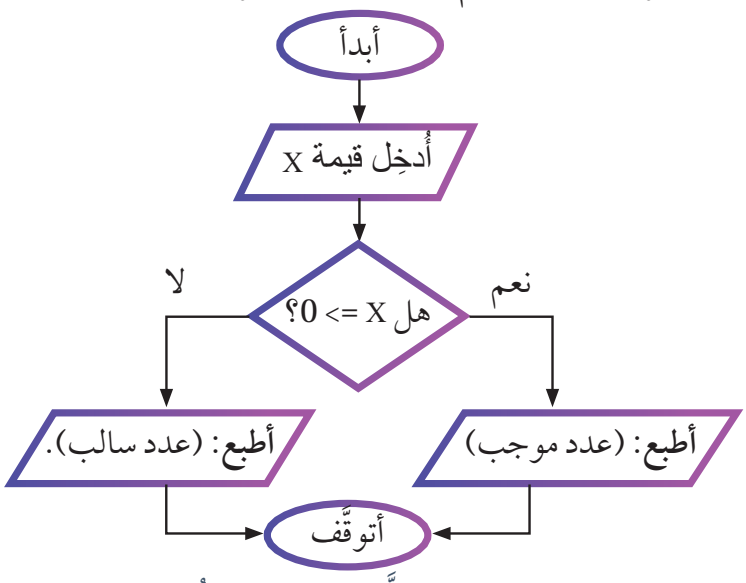
```
# A is the first number
A = int(input("Enter the first number"))

# B is the second number
B = int(input("Enter the second number"))

# C is the product
C = A * B

# Print the product
print("The product of A * B =", C)
```

الشكل (2-4): توثيق برنامج حاصل ضرب عددين باستخدام لغة البرمجة بايثون.

الخوارزمية للجمل الشرطية:

تؤدّي الجمل الشرطية دورًا مُهِمًّا في عملية اتِّخاذ القرار؛ إذ يتطلَّب تنفيذ أمر مُعيَّن أو مجموعة من الأوامر تحقُّق بعض الشروط. فعلى سبيل المثال، تُستخدَم الجمل الشرطية في تحديد إذا كان العدد المُدخَل موجبًا أم سالبًا، أو تحديد إذا كان العدد يقع ضمن فئة مُعيَّنة من القِيَم وغيرها.

يُستخدَم شكل المَعين في مُخطَّطات سَيْر العمليات لتمثيل الشرط كما هو مُبيَّن في الشكل (2-5)؛ إذ إنَّه يستقبل مُدخَلًا مُعيَّنًا من المرحلة التي تسبقه، ثمَّ يخرج منه مساران، أحدهما يُنفَّذ إذا كان جواب الشرط صحيحًا (نعم)، والآخر يُنفَّذ إذا كان جواب الشرط غير صحيح (لا). ويُطلَق على هذه المُخطَّطات - كما تعرَّفنا سابقًا- اسم المُخطَّطات التفرُّعية.



الشكل (2-5): مُخطَّط سَيْر عمليات تفرُّعي.

نشاط جماعي

اعتمادًا على خطوات حَلِّ المسألة، أَحُلُّ – بالتعاون مع أفراد مجموعتي – المسألة الآتية:

أُدخِل العدد X والعدد Y، ثمَّ أطبع العبارة الآتية: "X is greater than Y" إذا كانت قيمة X أكبر من قيمة Y، وأطبع العبارة الآتية: "Y is greater than X"إذا كانت قيمة Y هي القيمة الكبرى.

بعد ذلك أُعدِّل على خطواتي لإضافة خيار: في حال كان العددان متساويين، فإنَّني أطبع العبارة الآتية: "They are equal"، ثمَّ أطبع العددين.

أُشارِك أفراد المجموعات الأُخرى في الإجابة التي توصَّلنا إليها في المجموعة، ثمَّ نتبادل معًا التغذية الراجعة.

قد يتضمَّن البرنامج جملًا شرطيةً مُتداخِلةً، بحيث تُنفَّذ الجملة الشرطية الداخلية بناءً على صِحَّة الشرط أو عدم صِحَّته في الجملة الشرطية الخارجية.

مثال:

أُطبِّق خطوات حَلِّ المسألة على المسألة الآتية:

أُدخِل عددًا صحيحًا، وليكن X، ثمَّ أتأكَّد إذا كان هذا العدد يقبل القسمة على العدد 2 والعدد 3 معًا:

- **إذا كان العدد X يقبل القسمة على العدد 2 والعدد 3، فإنَّني أطبع العبارة الآتية:**

"Divisible by 2 and 3, then it's divisible by 6".

- **إذا كان العدد X يقبل القسمة على العدد 2 فقط، فإنَّني أطبع العبارة الآتية:**

"Divisible by 2 only".

- **إذا كان العدد X يقبل القسمة على العدد 3 فقط، فإنَّني أطبع العبارة الآتية:**

"Divisible by 3 only".

- **إذا كان العدد X لا يقبل القسمة على أيٍّ من العدد 2 والعدد 3، فإنَّني أطبع العبارة الآتية:**

"Not divisible by 2 and not divisible by 3, then not divisible by 6".

الحَلُّ:

- **الخطوة الأولى**: فهم المسألة، والعمل على تحليلها، بما في ذلك تحديد المسألة، والمُدخَلات، والمُخرَجات، وطريقة المعالجة.

تحديد المسألة: التحقُّق من قابلية قسمة عدد على العدد 6 بناءً على قابليته القسمة على العدد 2 والعدد 3، ثمَّ طباعة جملة تُبيِّن ذلك لكل حالة.

المُدخَلات: قيمة العدد X.

المُخرَجات: عبارة مُحدَّدة تعتمد على قيمة X:

إذا كانت قيمة X تقبل القسمة على العدد 2 والعدد 3 من دون باقٍ، فإنَّني أطبع العبارة الآتية:

"Divisible by 2 and 3, then it's divisible by 6".

إذا كانت قيمة X تقبل القسمة على العدد 2 فقط، فإنَّني أطبع العبارة الآتية:

"Divisible by 2 only".

إذا كانت قيمة X تقبل القسمة على العدد 3 فقط، فإنَّني أطبع العبارة الآتية:

"Divisible by 3 only".

إذا لم يتحقَّق أيٌّ من الشروط السابقة، فإنَّني أطبع العبارة الآتية:

"not divisible by 2 and not divisible by 3, then not divisible by 6".

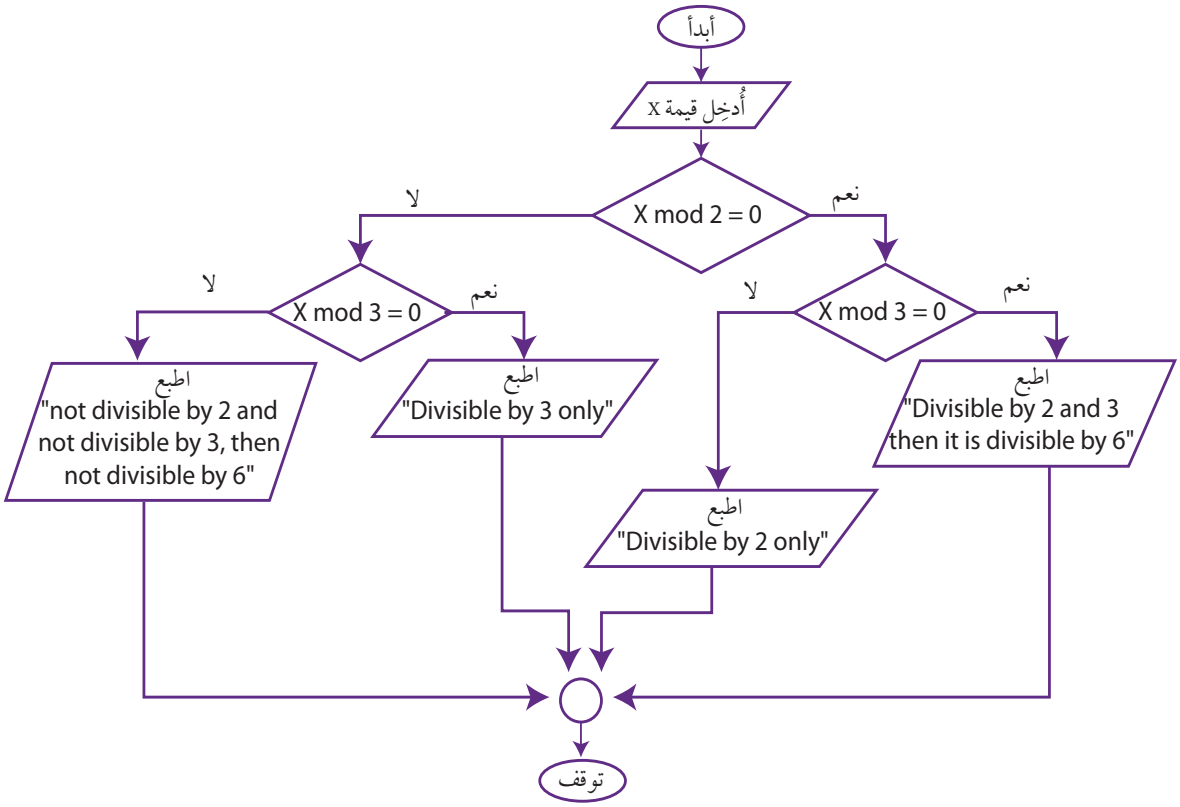
- **الخطوة الثانية**: كتابة الخوارزمية.

أ- الخوارزمية شبه الرمزية:

1. أبدأ.
2. أُدخِل قيمة X.
3. **أختبر**: هل تقبل قيمة X القسمة على العدد 2؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (4)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (7).
4. **أختبر**: هل تقبل قيمة X القسمة على العدد 3؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (5)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (6).
5. **أطبع العبارة الآتية**: "Divisible by 2 and 3, then it's divisible by 6"، ثمَّ أذهب إلى الخطوة رقم (10).
6. **أطبع العبارة الآتية**: "Divisible by 2 only"، ثمَّ أذهب إلى الخطوة رقم (10).
7. **أختبر**: هل تقبل قيمة X القسمة على العدد 3؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (8)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (9).
8. **أطبع العبارة الآتية**: "Divisible by 3 only"، ثمَّ أذهب إلى الخطوة رقم (10).

9. **أطبع العبارة الآتية:** "not divisible by 2 and not divisible by 3, then not divisible by 6"، ثمَّ أذهب إلى الخطوة رقم (10).
10. أتوقَّف.

ب- **مُخطَّط سَيْر العمليات:** يُمكِن رسم مُخطَّط سَيْر العمليات يدويًّا باستخدام القلم والورقة، ويُمكِن رسمه أيضًا باستخدام تطبيقات مُخصَّصة لهذا الغرض، مثل تطبيق draw.io الذي يتيح رسم مُخطَّطات سَيْر العمليات من الموقع الإلكتروني (الويب)، ثمَّ تخزينها في تطبيق (Drive) أو في جهاز الحاسوب الخاص بالمُستخدِم. كذلك يُمكِن تحميل هذا التطبيق في جهاز الحاسوب. أنظر الشكل (2-6) الذي يُبيِّن مُخطَّط سَيْر العمليات للمسألة المذكورة آنفًا.



الشكل (2-6): مُخطَّط سَيْر العمليات للتحقُّق من قابلية قسمة عدد على العدد 6.

إضاءة

يُمكِن الدخول إلى تطبيق draw.io عن طريق الرابط الإلكتروني الآتي:
https://www.drawio.com/، ثمَّ اختيار (Flowchart) من القائمة التي على يسار الشاشة.

- **الخطوة الثالثة**: كتابة البرنامج.

أُحوِّل مُخطَّط سَيْر العمليات في الخطوة السابقة إلى برنامج مكتوب بلغة البرمجة بايثون كما في الشكل (2-7).

```
X = int(input("Enter a number: "))

if X % 2 == 0:
    if X % 3 == 0:
        print("Divisible by 2 and 3, then it's divisible by 6")
    else:
        print("Divisible by 2 only")
else:
    if X % 3 == 0:
        print("Divisible by 3 only")
    else:
        print("Not divisible by 2 and not divisible by 3, then not
divisible by 6")
```

الشكل (2-7): برنامج كُتِب بلغة البرمجة بايثون للتحقُّق من قابلية قسمة عدد على العدد 6.

نشاط جماعي

أكتب - بالتعاون مع أفراد مجموعتي- البرنامج الوارد في الشكل (2-7) بلغة البرمجة بايثون، ثمَّ نُنفِّذ البرنامج معًا باستخدام مُحرِّر بايثون للتحقُّق من صِحَّة النتائج، فضلًا عن فحص المُخرَجات لضمان دِقَّتها، وإضافة جمل التعليقات والملاحظات المناسبة داخل الكود لتوثيقه، بما يُسهِّل فهم الخطوات، ويساعد على توضيح الوظائف.

أُشارِك أفراد المجموعات الأُخرى في ما توصَّلنا إليه في المجموعة، ونتبادل معًا الأفكار والتغذية الراجعة لتحسين الحَلِّ.

نشاط فردي

أدرس المسألة الآتية، ثمَّ أُجيب عن الأسئلة التي تليها:

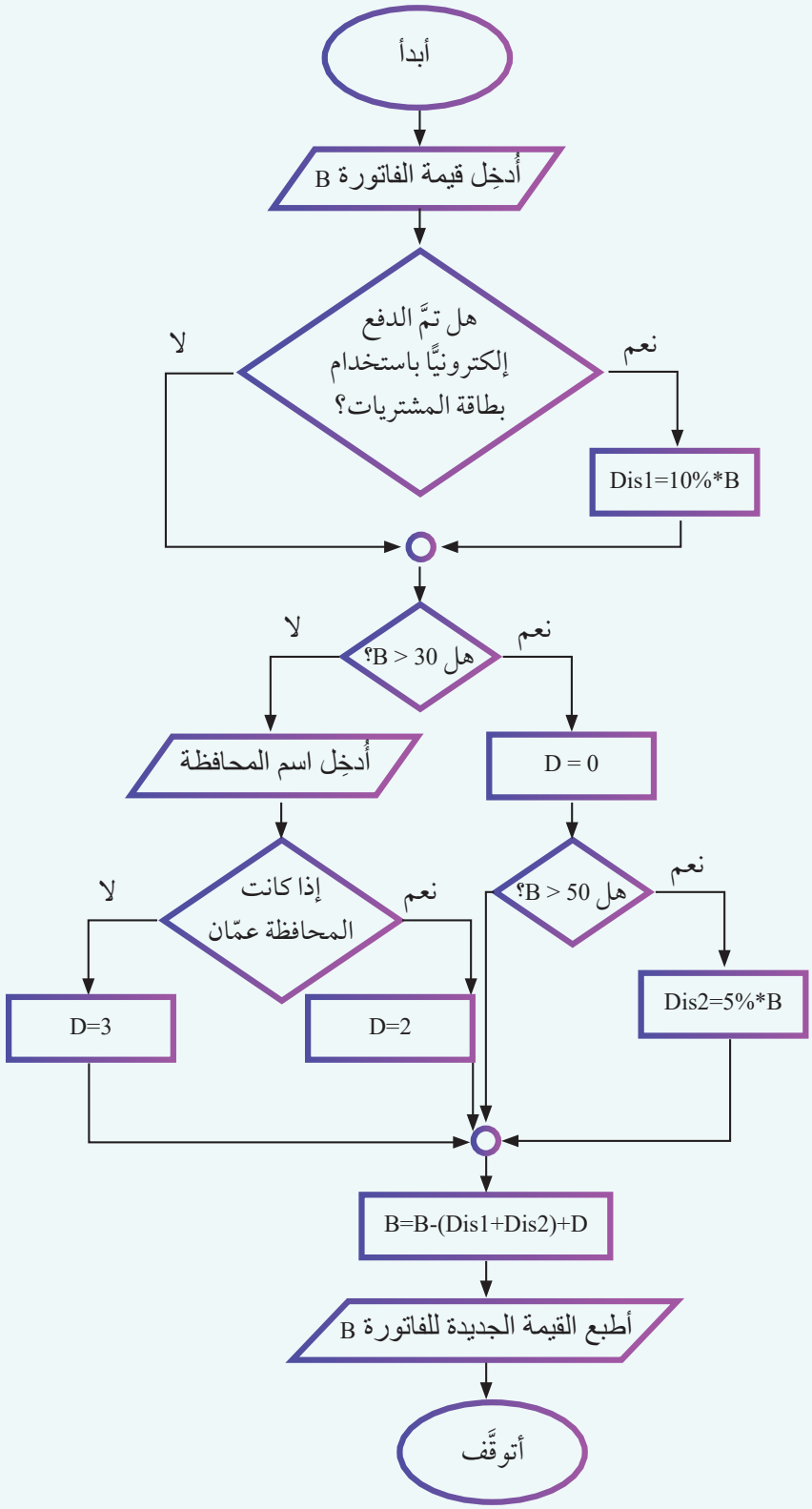
أعلن متجر إلكتروني عن حملة للمبيعات تتضمَّن خصمًا على فواتير الشراء بحسب الشروط الآتية:

- إذا استخدم المُستهلِك بطاقة المشتريات في عملية الدفع الإلكتروني، فإنَّه يحصل على خصم نسبته 10٪ من قيمة الفاتورة، ويعفى من رسوم التوصيل.
- إذا تجاوزت قيمة الفاتورة (50) دينارًا، فإنَّ المُستهلِك يحصل على خصم إضافي بما نسبته 5٪.
- إذا تراوحت قيمة الفاتورة بين (30) دينارًا و(50) دينارًا، فإنَّ المُستهلِك يعفى فقط من رسوم التوصيل. في الحالات الأُخرى، تُفرَض على المُستهلِك رسوم توصيل، مقدارها ديناران داخل مدينة عمّان، وثلاثة دنانير في بقيَّة المحافظات.

أ- بناءً على الشروط السابقة، أتتبَّع الخوارزمية شبه الرمزية المُرفَقة؛ لتحديد النتيجة، علمًا بأنَّ المُستهلِك سيدفع ثمن مشترياته إلكترونيًّا باستخدام بطاقة المشتريات، وقيمة فاتورته (25) دينارًا، وهو يقيم في محافظة الزرقاء.

1. أبدأ.
2. أُدخِل قيمة الفاتورة الأصلية B.
3. هل تمَّ الدفع إلكترونيًّا باستخدام بطاقة المشتريات؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (4)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (5).
4. الخصم الأوَّل Dis1 نسبته 10٪ من قيمة الفاتورة الأصلية.
5. هل كانت قيمة الفاتورة الأصلية أكثر من (30) دينارًا؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (6)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (7).
6. قيمة التوصيل D=0، إذن أذهب إلى الخطوة رقم (8).
7. أُدخِل اسم المحافظة، ثمَّ أذهب إلى الخطوة رقم (10).
8. هل كانت قيمة الفاتورة الأصلية أكثر من (50) دينارًا؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (9)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (13).
9. **قيمة الخصم الثاني**: Dis2 = 5% * B، إذن أذهب إلى الخطوة رقم (13).
10. هل المحافظة = عمّان؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (11)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (12).
11. قيمة التوصيل D=2، إذن أذهب إلى الخطوة رقم (13).
12. قيمة التوصيل D=3، إذن أذهب إلى الخطوة رقم (13).
13. **قيمة الفاتورة الجديدة تساوي قيمة الفاتورة القديمة مطروحًا منها الخصمان، ومضافًا إليها قيمة التوصيل**: B= B – (Dis1+Dis2) + D.
14. أطبع القيمة الجديدة للفاتورة.
15. توقَّف.

ب- أتتبَّع مُخطَّط سَيْر العمليات في الشكل (2-8)، وأُبيِّن النتيجة إذا علمْتُ أنَّ المُستهلِك سيدفع عند التسلُّم، وأنَّ قيمة فاتورته بلغت (70) دينارًا، وأنَّه يقيم في محافظة الزرقاء.



الشكل (2-8): مُخطَّط سَيْر العمليات لحساب فاتورة الشراء من متجر إلكتروني.

ج- **أُقارِن**: أيُّهما أسهل: تتبُّع مُخطَّط سَيْر العمليات أم الخوارزمية شبه الرمزية؟ أُبرِّر إجابتي.

نشاط فردي

تُحسَب علامة الطالب النهائية في إحدى الجامعات بناءً على مجموع ثلاث قِيَم، هي: علامة منتصف الفصل (Mid Grade)، وعلامة المشروع (Project's Grade)، وعلامة الامتحان النهائي (Final Grade). إذا كان المجموع النهائي (Total):

- 85 أو أكثر، فإنَّ الطالب يحصل على الرمز (A).
- 75 أو أكثر، وأقلَّ من 85، فإنَّ الطالب يحصل على الرمز (B).
- 65 أو أكثر، وأقلَّ من 75، فإنَّ الطالب يحصل على الرمز (C).
- 50 أو أكثر، وأقلَّ من 65، فإنَّ الطالب يحصل على الرمز (D).
- أقلَّ من 50، فإنَّ الطالب يحصل على الرمز (F).

يُبيِّن مُخطَّط سَيْر العمليات في الشكل (2-9) الخطوات التفصيلية لحساب الرمز بناءً على المجموع.

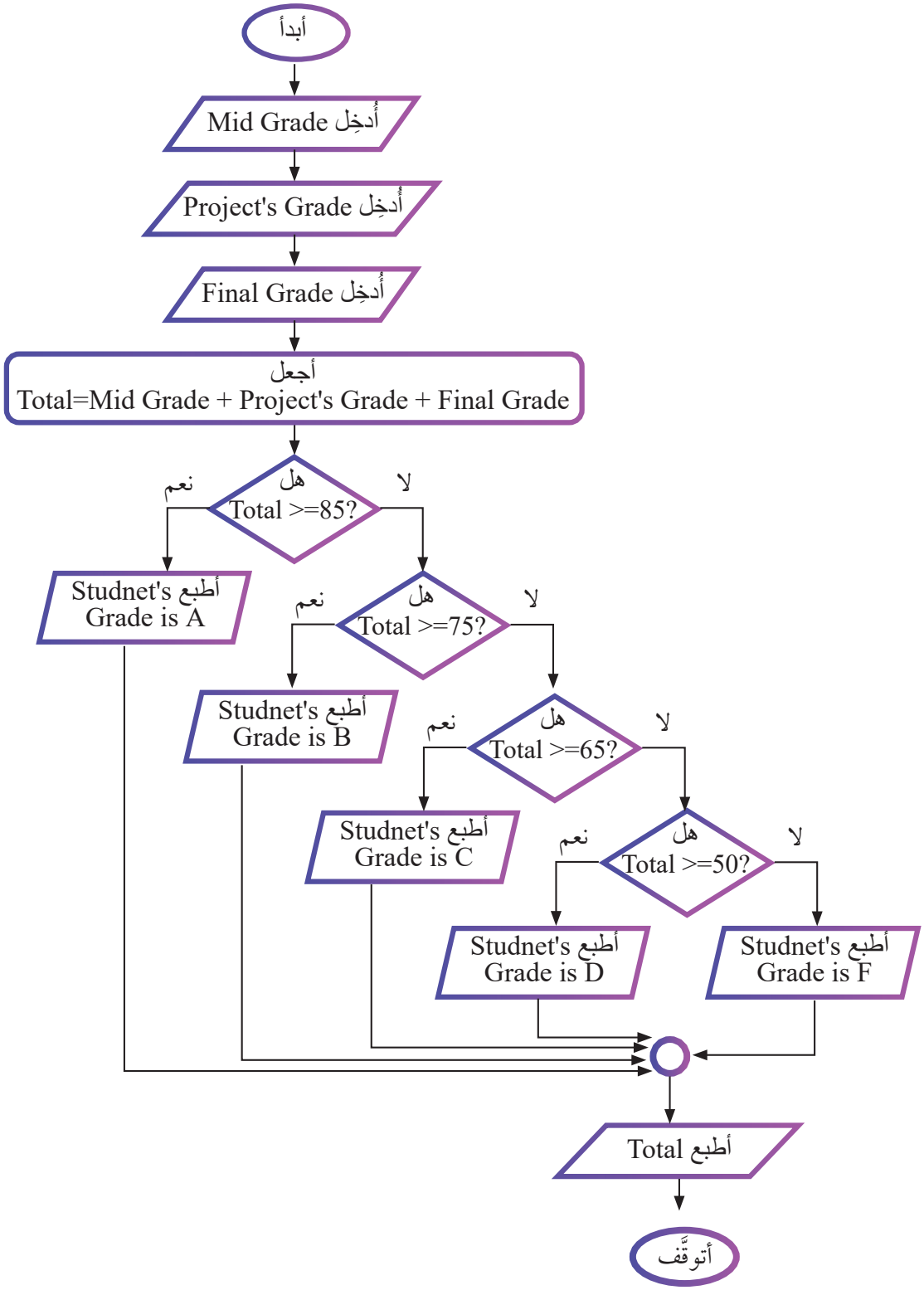
أ- بناءً على مُخطَّط سَيْر العمليات المُبيَّن في الشكل (2-9)، أتتبَّع الخطوات الواردة في المُخطَّط لإيجاد الناتج لكلٍّ من الحالات الآتية، ثمَّ أملأ الفراغ في الجدول بما هو مناسب.

	Mid Grade	Project's Grade	Final Grade	Total	المخرجات (الرموز)
-1	25	28	35		
-2	15	20	30		
-3	20	25	25		

ب- أكتب خوارزمية شبه رمزية للتعبير عن مُخطَّط سَيْر العمليات السابق.

ج- هل يُمكِن اختصار الخطوات الواردة في الخوارزمية؟ أُوضِّح إجابتي، وأُعدِّل الخوارزمية لتحقيق ذلك.

أُشارِك زملائي/ زميلاتي في ما أتوصَّل إليه من نتائج، ثمَّ نتبادل معًا التغذية الراجعة.



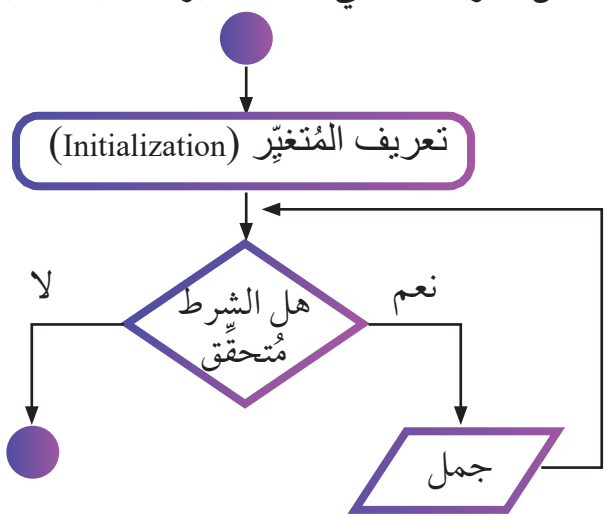
الشكل (2-9): مُخطَّط سَيْر العمليات لحساب علامات الطلبة في إحدى الجامعات.

نشاط جماعي

أُحوِّل – بالتعاون مع أفراد مجموعتي – مُخطَّط سَيْر العمليات السابق إلى برنامج مكتوب بلغة البرمجة بايثون، ثمَّ نُجرِّبه معًا باستخدام مُحرِّر بايثون للتحقُّق من صِحَّته، ثمَّ نعرض البرنامج المكتوب أمام أفراد المجموعات الأُخرى، ونُقارِنه بالبرامج التي كتبوها.

الخوارزمية لجمل التكرار:

تعرَّفْتُ سابقًا الحلقات (أيْ جمل التكرار)، واستخدمتها في برمجية سكراتش (Scratch)، وفي لغة البرمجة بايثون (Python). تُستخدَم جمل التكرار في كتابة البرامج التي تتطلَّب تنفيذ مجموعة من الأوامر والتعليمات عددًا من المَرّات كما في مُخطَّط سَيْر العمليات المُبيَّن في الشكل (2–10).



الشكل (2–10): مُخطَّط سَيْر العمليات لجملة التكرار.

مثال:

أُحلِّل المسألة الآتية باستخدام خطوات حَلِّ المسألة:

"يُدخِل المُستخدِم مجموعة من الأعداد ، ثمَّ يُصنِّفها إلى أعداد موجبة وأعداد سالبة ، ويتوقَّف عند إدخال العدد 0".

الحَلُّ:

الخطوة الأولى: فهم المسألة، والعمل على تحليلها، بما في ذلك تحديد المسألة، والمُدخَلات، والمُخرَجات، وطريقة المعالجة.

تحديد المسألة: تكرار إدخال مجموعة من الأرقام، وتحديد الأعداد الموجبة والأعداد السالبة، والتوقُّف إذا أدخل المُستخدِم الرقم 0

- **المُدخَلات:** مجموعة من الأعداد. ولأنَّ الخوارزمية ستستخدم عملية التكرار؛ فإنَّني أحتاج إلى مُتغيِّر واحد (Num) لاستخدامه في إدخال العدد كل مَرَّة.

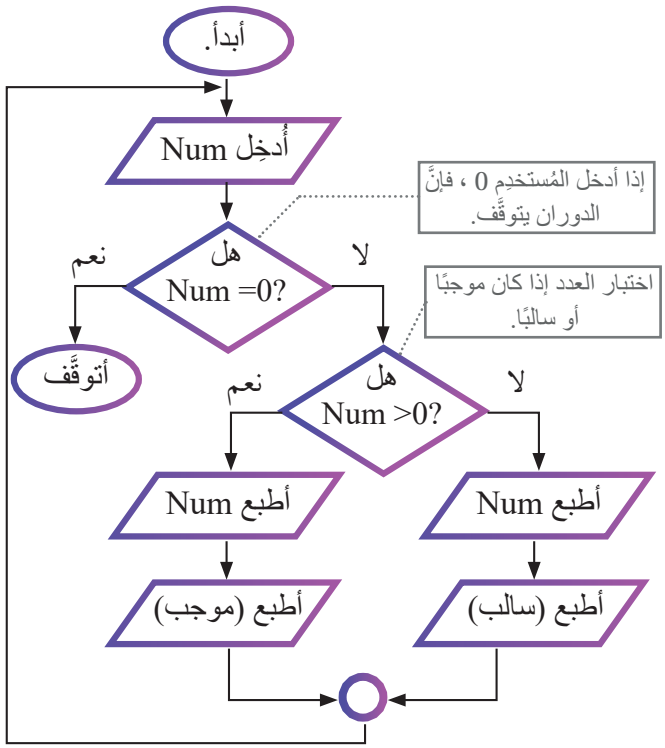
- **المُخرَجات:**
  - إذا كان العدد موجبًا،فإنَّ العدد يُطبَع، ثمَّ تُطبَع كلمة (موجب).
  - إذا كان العدد سالبًا، فإنَّ العدد يُطبَع، ثمَّ تُطبَع كلمة (سالب).
- **طريقة المعالجة:** عملية مقارنة منطقية تتضمَّن مقارنة العدد المُدخَل بالعدد 0 لتحديد إذا كان هذا العدد موجبًا أو سالبًا.

الخطوة الثانية: كتابة الخوارزمية.

- **الخوارزمية شبه الرمزية:**

1. أبدأ.
2. أُدخِل قيمة Num.
3. هل Num=0؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (8)، وإذا كانت الإجابـــة (لا)، فإنَّني أذهب إلى الخطوة رقم (4).
4. هل Num >0؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (5)، وإذا كانت الإجابـــة (لا)، فإنَّني أذهب إلى الخطوة رقم (6).
5. أطبع العدد، ثمَّ أطبع كلمة (موجب)، ثمَّ أذهب إلى الخطوة رقم (7).
6. أطبع العدد، ثمَّ أطبع كلمة (سالب)، ثمَّ أذهب إلى الخطوة رقم (7).
7. أذهب إلى الخطوة رقم (2).
8. أتوقَّف.

- مُخطَّط سَيْر العمليات كما هو مُبيَّن في الشكل (2-11)



الشكل (2-11): مُخطَّط سَيْر العمليات لإدخال مجموعة أعداد، وتصنيفها إلى موجبة وسالبة.

الخطوة الثالثة: كتابة البرنامج عن طريق ترجمة الخوارزمية إلى إحدى لغات البرمجة.
يُمكِن تحويل المُخطَّط – كما تعلَّمْتُ سابقًا كيفية كتابة جمل الدوران أو التكرار– باستخدام لغة البرمجة بايثون (Python)، أو باستخدام برمجية سكراتش (Scratch).

نشاط إثرائي

أُفكِّر في التعديل الذي يجب إجراؤه على المُخطَّط السابق إذا بلغت الأعداد المُدخَلة (20) عددًا، ثمَّ أُشارِك الزملاء/ الزميلات بالصف في ما أتوصَّل إليه من نتائج.

نشاط عملي

أتعاون مع أفراد مجموعتي على كتابة البرنامج الوارد في الشكل (2-11) بلغة البرمجة بايثون، ثمَّ نُنفِّذ البرنامج معًا باستخدام مُحرِّر بايثون للتحقُّق من صِحَّة النتائج، فضلًا عن فحص المُخرَجات لضمان دِقَّتها، وإضافة جمل التعليقات والملاحظات المناسبة داخل الكود لتوثيقه، بما يُسهِّل فهم الخطوات، ويساعد على توضيح الوظائف.
أُشارِك أفراد المجموعات الأُخرى في ما توصَّلْتُ إليه من خلال مجموعتي، ونتبادل معًا الأفكار والتغذية الراجعة لتحسين الحَلِّ.

نشاط فردي

أُفكِّر في المثال السابق، ثمَّ أُحدِّد خطوة واحدة أو مجموعة خطوات يُمكِن اختصارها، ثمَّ أُعدِّل على مُخطَّط سَيْر العمليات.
أُشارِك الزملاء/ الزميلات بالصف في ما أتوصَّل إليه من نتائج.

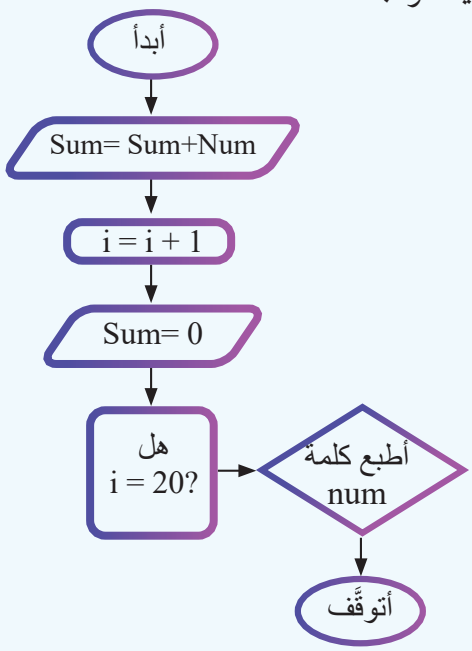
إضاءة

عند مقارنة الخوارزميات بعضها ببعض، لا بُدَّ من مراعاة العوامل الآتية:

1. **وقت التنفيذ:** يُقصَد بذلك الوقت اللازم لتنفيذ البرنامج المُعتمِد على إحدى الخوارزميات.
2. **عدد العمليات التي يُراد تنفيذها:** يعتمد ذلك على لغة البرمجة المُستخدَمة، وطريقة المُبرمِج في تحويل الخوارزمية إلى برنامج.
3. **الحَلُّ المثالي:** يُقصَد بذلك أفضل حَلٍّ يُمكِن تقديمه للمسألة.
4. **حجم الذاكرة المُستخدَم:** كلَّما قَلَّ استهلاك حجم الذاكرة كانت الخوارزمية أفضل من غيرها.

استكشاف الأخطاء وتصحيحها

أُحلِّل - بالتعاون مع أفراد مجموعتي- مُخطَّط سَيْر العمليات الذي يُبيِّنه الشكل (2-12)، والذي يُمثِّل إيجاد مجموع الأعداد الزوجية المحصورة بين (1) و (20)، ثمَّ أعمل مع أفراد مجموعتي على اكتشاف الأخطاء وتصحيحها، ثمَّ نُناقِش أفراد المجموعات الأُخرى في هذه الأخطاء، ونتبادل معًا التغذية الراجعة.



الشكل (2-12): مُخطَّط سَيْر العمليات لحساب مجموع أوَّل (10) أرقام زوجية باستخدام حلقة تكرار.

المواطنة الرقمية

- **الأمان الرقمي** (Digital Security): أستخدم خوارزميات التشفير لضمان حماية البيانات الرقمية وخصوصية الأفراد، وأتأكَّد أنَّ البيانات التي تخضع للمعالجة بواسطة الخوارزميات لا تُستخدَم بشكل غير آمن ، أو لا تنتهك الخصوصية.
- **المسؤولية الرقمية** (Digital Responsibility): أُراعي الشفافية وأخلاقيات التعامل الرقمي عند استخدام الخوارزميات وتطويرها. كذلك أختار الخوارزمية التي تستهلك أصغر حيِّز من الذاكرة، وتستغرق أقلَّ وقت للتنفيذ.

أُقيِّم تعلُّمي:

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أكتب خوارزمية شبه رمزية للتعبير عن كل مسألة من المسائل الآتية:

1. إدخال علامات طالب في (7) مواد، وطباعة مُعدَّلها والعلامة الكبرى من بينها.
2. حساب مضروب العدد n.
3. إدخال مُكوِّنات بريد إلكتروني؛ كلٍّ على حِدَة، ثمَّ طباعتها في صورة بريد إلكتروني صحيح.

السؤال الثاني: أدرس الخوارزمية شبه الرمزية الآتية، ثمَّ أُجيب عن الأسئلة التي تليها:

1. أبدأ.
2. أجعل Sum=0
3. أجعل i = 1
4. أجعل Sum = Sum + i
5. هل i = 20 ؟ إذا كانت الإجابة (نعم)، فإنَّني أذهب إلى الخطوة رقم (8)، وإذا كانت الإجابة (لا)، فإنَّني أذهب إلى الخطوة رقم (6).
6. أجعل i=i +1
7. أذهب إلى الخطوة رقم (4).
8. أطبع قيمة (Sum).
9. أتوقَّف

- أُحدِّد وظيفة الخوارزمية ومُخرَجاتها.
- أُترجِم الخوارزمية إلى برنامج مكتوب بلغة البرمجة بايثون.
- أَجِد الناتج عمليًّا.

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُقارِن الخوارزمية التي كتبْتُها في السؤال الأوَّل (من بند المعرفة) بالخوارزميات التي كتبها أفراد مجموعتي، ثمَّ نُحدِّد معًا الخوارزمية التي يُمكِن كتابتها بأقلِّ عدد مُمكِن من الخطوات وفي أقصر وقت.

السؤال الثاني: تدفع إدارة أحد المصانع رواتب أساسية للعُمّال بناءً على فئاتهم، إضافةً إلى أَجْر عن ساعات العمل الإضافي وَفق المعايير الآتية:

- **العُمّال من الفئة** (A): الراتب الأساسي (500) دينار، و(7) دنانير عن كل ساعة عمل إضافية.
- **العُمّال من الفئة** (B): الراتب الأساسي (400) دينار، و(5) دنانير عن كل ساعة عمل إضافية.
- **العُمّال من الفئة** (C): الراتب الأساسي (300) دينار، و(3) دنانير عن كل ساعة عمل إضافية.

شرط ألّا يتجاوز عدد ساعات العمل الإضافي لكل عامل (30) ساعة في الشهر.

1. أُحلِّل المسألة وَفقًا لخطوات حَلِّ المسألة التي تعلَّمْتُها.
2. أرسم مُخطَّط سَيْر العمليات لحساب إجمالي ما ستدفعه إدارة المصنع من رواتب في نهاية كل شهر، علمًا بأنَّ العُمّال في المصنع مُوزَّعين على النحو الآتي: (15) عاملًا من الفئة (A)، و(17) عاملًا من الفئة (B)، و(10) عُمّال من الفئة (C).

السؤال الثالث: أكتب خوارزمية شبه رمزية لطباعة جداول الضرب للأعداد من (3) إلى (9)؛ على أنْ تتضمَّن حاصل ضرب كل عدد في الأرقام من (1) إلى (10) على شكل جدول مُنظَّم، ثمَّ أرسم مُخطَّط سَيْر العمليات الذي يُمثِّل هذه المسألة. بعد ذلك أُقارِن بين الخوارزمية شبه الرمزية ومُخطَّط سَيْر العمليات الذي رسمته، وأُوضِّح إذا كانت هذه الخوارزمية مُتطابِقة مع مُخطَّط سَيْر العمليات أم لا، وأُبرِّر إجابتي.

الدرس الثالث

هياكل تراكيب البيانات (Data Structures)

الفكرة الرئيسة:

سأتعلَّم في هذا الدرس كيف أَصِف تراكيب البيانات المختلفة، وكيف أختار التركيب المناسب لتمثيل مُدخَلات البرامج ومُخرَجاتها. كذلك سأستكشف الطرائق التي تُستخدَم فيها هذه التراكيب استخدامًا فاعلًا لتحسين كفاءة البرامج وضمان ملاءمتها لطبيعة البيانات المطلوبة.

المفاهيم والمصطلحات:

هياكل البيانات (Data Structures)، القائمة (List)، الفهرسة (Index)، الصفوف (Tuples)، القواميس (Dictionaries)، المجموعة (Set)، البيانات المُجرَّدة (Abstract Data Type)، القوائم المُترابِطة (Linked Lists)، المكدس (Stack).

نتاجات التعلُّم (Learning Outcomes):

- أُحدِّد نوع البيانات المناسبة لتمثيل مُدخَلات مشكلة مُحدَّدة ومُخرَجاتها، مثل: النصوص (String)، والقوائم (Lists)، والمصفوفات (Arrays)، والمكدسات (Stacks).
- أُقارِن بين تراكيب البيانات المختلفة من حيث الخصائص والاستخدامات.
- أختار أنسب الهياكل لتراكيب البيانات من بين مجموعة خيارات مُحدَّدة بما يُوائِم أهداف البرنامج.

تعرَّفْتُ سابقًا مفهوم القائمة في لغة البرمجة بايثون وأهميتها في إدارة تنظيم البيانات والتعامل معها واسترجاعها بسهولة، وسأتعرَّف في هذا الدرس أنواعًا أُخرى من هياكل البيانات.

نشاط تمهيدي

اعتمادًا على ما تعلَّمْتُه في الصف الحادي عشر عن القوائم (Lists) في لغة البرمجة بايثون (Python) وكيفية تعريفها، أكتب مقطعًا برمجيًّا لتعريف قائمة تحتوي على أسماء زملائي/ زميلاتي في الصف، ثمَّ أُدخِل هذه الأسماء باستخدام البرنامج. بعد ذلك أُجرِّب تشغيل البرنامج بواسطة مُفسِّر لغة البرمجة بايثون للتحقُّق من صِحَّة النتائج.

تُعرَّف هياكل البيانات (Data Structure) بأنَّها طريقة لتنظيم البيانات وتخزينها داخل جهاز الحاسوب على نحوٍ يُمكِّن من استخدامها بكفاءة. تتيح هذه الهياكل تخزين بيانات أساسية من أنواع مُتعدِّدة (مثل: int, float,boolean char and string)، أو بيانات مُركَّبة (مثل Objects)، أو تراكيب بيانات أُخرى يُمكِن للمُستخدِم تعريفها (مثل: List, Tuple, Set and Dictionary).

البيانات المُركَّبة في لغة البرمجة بايثون:

البيانات المُركَّبة هي أحد أنواع البيانات التي قد تحتوي على أكثر من قيمة، وتكون مُرتَّبة بطريقة تُسهِّل التعامل معها. تُستخدَم البيانات المُركَّبة في تخزين البيانات وتنظيمها على نحوٍ مُتَّسِق يتيح للمُبرمِج تنفيذ عمليات مُعقَّدة بكفاءة.
يمتاز كل نوع من البيانات المُركَّبة بخصائص ووظائف تُناسِب أغراضًا مُحدَّدةً.

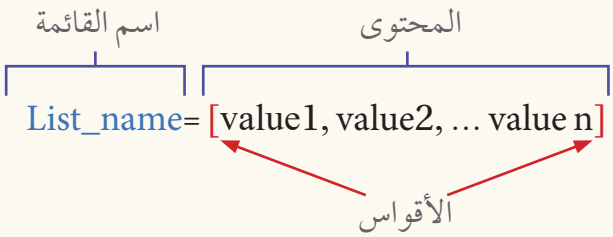
وهذه بعض أنواع البيانات المُركَّبة الشائعة في لغة البرمجة بايثون وكيفية استخدامها:

١- القوائم (Lists):

تعرَّفْتُ في الصف الحادي عشر أنَّ القائمة هي مجموعة من القِيَم التي يُخزَّن بعضها مع بعض، والتي ترتبط عادةً في ما بينها بمعنى وظيفي مُشترَك. فمثلًا، يُمكِن تخزين أسماء شوارع مدينة ما في قائمة، وتخزين أسعار البضائع التي اشتراها عميل ما في قائمة، وتخزين إحداثيات تحرُّك سيّارة ما في قائمة، وهكذا. تعرَّفْتُ أيضًا أنَّه لا يُشترَط في القائمة أنْ تكون جميع عناصرها من النوع نفسه؛ إذ يُمكِن تخزين أرقام صحيحة وأرقام عشرية وسلاسل حروف في قائمة واحدة، حتّى إنَّه يُمكِن تخزين قوائم أُخرى في هذه القائمة.

إضاءة

يُمكِن تعريف قائمة ما باستخدام الأقواس المُربَّعة، وكذا الفصل بين عناصر القائمة بفواصل على النحو الآتي:



حيث:

- **List_name**: اسم القائمة.
- **[ ]**: الأقواس المُربَّعة (تُستخدَم في حصر عناصر القائمة).
- **Value1**: العنصر الأوَّل في القائمة.
- **Value n**: العنصر الأخير في القائمة.

مثال (1):

يُبيِّن البرنامج الآتي كيف تُستخدَم الأوامر مع القوائم في لغة البرمجة بايثون.

```
# تعريف قائمة أولية
my_list = [1, 2, 3]

# إضافة عناصر
my_list.append(4)
my_list.insert(1, 99)
my_list.extend([5, 6, 7])

# حذف عناصر
del my_list[2]
removed = my_list.pop(6)
my_list.remove(99)

# طباعة القائمة النهائية
print("القائمة النهائية:", my_list)
```

بعد تنفيذ البرنامج، سيظهر الناتج الآتي:

```
القائمة النهائية: [1, 3, 4, 5, 6]
```

مثال (2):

يُبيِّن البرنامج الآتي كيفية تعريف إحدى القوائم في لغة البرمجة بايثون، ثمَّ طباعة كلٍّ من قِيَمها ونوعها.

```
# تعريف قائمة تحتوي على أرقام
L = [2, 4, 6, 8]

# طباعة محتويات القائمة
print(L)

# طباعة نوع البيانات للقائمة
print(type(L))
```

بعد تنفيذ البرنامج، سيظهر الناتج الآتي:

```
[2, 4, 6, 8]
<class 'list'>
```

نشاط فردي

أَجِد النمط الخاص بعناصر القائمة في المثال السابق.

أكتب البرنامج في بيئة بايثون، ثمَّ أُضيف عنصرين آخرين إلى هذه القائمة باستخدام لغة البرمجة بايثون.

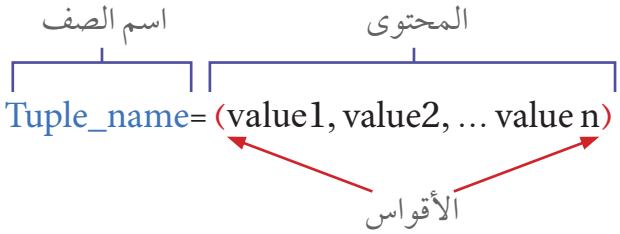
أطبع العنصر الثالث من القائمة، ثمَّ أُشارِك الزملاء/ الزميلات بالصف في ما أتوصَّل إليه من نتائج.

2- الصفوف (Tuples):

تُعرَّف الصفوف بأنَّها مجموعة مُتسلسِلة مُرتَّبة من القِيَم، وهي تمتاز عن القوائم بثباتها وعدم قابليتها للتعديل والتغيير. وهذا يعني إمكانية الوصول إلى العناصر داخل الصفوف، ولكنْ من دون تغيير العناصر الموجودة أو إضافة عناصر أُخرى. عند إنشاء الصفوف، توضَع العناصر المُكوِّنة لها داخل أقواس، ويُفصَل بينها بفواصل.

الصيغة العامة لتعريف الصف (Tuple):

تُعرَّف الصفوف باستخدام الأقواس الدائرية، ويُفصَل بين عناصر القائمة بفواصل كما يأتي:



حيث:

- Tuple_name: اسم الصف (Tuple).
- ( ): الأقواس الدائرية التي تُستخدَم في حصر عناصر الصف (Tuple).

مثال:

يُبيِّن البرنامج الآتي كيفية تعريف الصف (Tuple)، ثمَّ طباعة كلٍّ من القِيَم والنوع.

```
# tuple تعريف متغير يحتوي على
T = (2, 4, 6, 8)

# tuple طباعة عناصر
print(T)

#  طباعة نوع المتغير
print(type(T))
```

بعد تنفيذ البرنامج، سيظهر الناتج الآتي:

```
(2, 4, 6, 8)
<class 'tuple'>
```

نشاط جماعي

أُنفِّذ – بالتعاون مع أفراد مجموعتي – المهمة الآتية:

- فتح (IDLE) الخاص بلغة البرمجة بايثون، ثمَّ إدخال البرنامج الوارد ذكره في المثال السابق.
- حفظ البرنامج، ثمَّ تنفيذه. هل يوجد تشابه بين هذه النتائج والنتائج التي ظهرت في المثال؟
- إضافة عنصر إلى الصف (tuple) السابق باستخدام أمر (append). ماذا نُلاحِظ؟
- **التعديل على العنصر الثاني بحيث تصبح قيمته:** T [1]=A. ماذا نُلاحِظ؟

نشاط فردي

أُنفِّذ البرنامج الآتي باستخدام مُفسِّر لغة البرمجة بايثون.

```
# تعريف ال (Tuple) بعنصر واحد.
Tuple_elements = (5,)

# tuple طباعة عناصر
print(Tuple_elements)

# طباعة نوع المتغير
print(type(Tuple_elements))
```

- ما مُخرَجات البرنامج؟
- أحذف الفاصلة بعد الرقم (5) عند تعريف الصف، ثمَّ أعمل على حفظ البرنامج، ثمَّ أُنفِّذه. ما مُخرَجات البرنامج؟ ماذا أستنتج؟

أُشارِك الزملاء/ الزميلات بالصف في ما أتوصَّل إليه من نتائج.

نشاط فردي

أُنفِّذ البرنامج الآتي باستخدام مُفسِّر لغة البرمجة بايثون.

```
# تعريف Tuple تحتوي على أسماء وارقام.
Names = (‹Hmad›, ‹Omar›, ‹Saleh›, ‹Ali›, ‹Zaid›, ‹Yazan›, 10, 12, 14, 16)

# طباعة محتويات الـ Tuple بالكامل.
print(Names)

# طباعة جزء محدد من عناصر الـ Tuple من الفهرس 3 إلى 6
print(Names[3:7])

# طباعة نوع البيانات للمتغير Names
print(type(Names))
```

- ما مُخرَجات البرنامج؟
- كيف تتعامل لغة البرمجة بايثون مع الصفوف لتعرُّف عناصرها؟

أُشارِك الزملاء/ الزميلات بالصف في ما أتوصَّل إليه من نتائج.

3- القواميس (Dictionaries):

تُخزَّن محتويات القاموس على شكل أزواج من المفاتيح (Keys) والقِيَم (Values)، بحيث يُمكِن الوصول إلى القيمة المُرتبِطة بالمفتاح بسهولة عند معرفة المفتاح.

فعلى سبيل المثال، يُمكِن تعرُّف اسم الطالب عن طريق الرقم الجامعي الذي يحمله؛ فهذا الرقم هو المفتاح، واسم الطالب هو القيمة. كذلك يُمكِن استخدام الرقم الجامعي مفتاحًا، واستخدام المواد التي يَدْرسها الطالب بوصفها قيمةً تُخزَّن في قائمة ما.

الصيغة العامة لتعريف القاموس (Dictionary):

يُمكِن تعريف القاموس كما هو مُبيَّن في الآتي.

```
Dictionary_name = dict ( )

Dictionary_name [Key1] = Value 1
Dictionary_name [Key2] = Value 2
Dictionary_name [Key3] = Value 3
```

حيث:

Dictionary_name: اسم القاموس الذي نرغب في إنشائه.

dict (): كلمة محجوزة تُستخدَم في تعريف قاموس فارغ في لغة البرمجة بايثون، وتوضَع أقواس دائرية بعدها.

Key1: الكلمة المفتاحية الأولى في القاموس، وهي المفتاح الذي يُمكِن عن طريقه الوصول إلى القيمة.

Value1: القيمة المُرتبِطة بالمفتاح الأوَّل في القاموس.

يُذكَر أنَّ مفاتيح القاموس تمتاز بأنَّها فريدة، ولا يُمكِن تكرارها؛ أيْ لا يُمكِن للقاموس أنْ يحتوي على مفتاحين يحملان الاسم نفسه. أمّا القِيَم المُرتبِطة بالمفاتيح فقد تتكرَّر، ولكنْ لا يُشترَط فيها أنْ تكون فريدة.

مثال:

يُبيِّن البرنامج الآتي كيف يُمكِن إنشاء قاموس خاص بمجموعة من كلمات اللغة الإنجليزية ومعانيها باللغة العربية، ثمَّ طباعتها.

```
# تعريف القاموس
My_dictionary = dict()
My_dictionary["eat"] = "أكل"
My_dictionary["sleep"] = "نام"
My_dictionary["swim"] = "سبح"

# طباعة القاموس
print(My_dictionary)

# طباعة نوع القاموس
print(type(My_dictionary))
```

بعد تنفيذ البرنامج، سيظهر الناتج الآتي:

```
{'eat': 'أكل', 'sleep': 'نام', 'swim': 'سبح'}
<class 'dict'>
```

يُلاحَظ أنَّ البرنامج عرض القاموس باستخدام أقواس المجموعة { } (curly parenthesis)، وأنَّ كل كلمة مفتاحية طُبِعت بين إشارات اقتباس فردية متبوعة بنقطتين رأسيتين، تلتها القيمة المُرتبِطة بتلك الكلمة، وأنَّه فُصِل بين العناصر داخل القاموس بفواصل عادية؛ إذ طُبِع كل عنصر بالصيغة الآتية: (القيمة المُرتبِطة بها: الكلمة المفتاحية)، وهكذا.

نشاط فردي

أكتب البرنامج الآتي باستخدام مُفسِّر لغة البرمجة بايثون، ثمَّ أعمل على تنفيذه، وأُلاحِظ الناتج.

```
Dic = {"eat": "أكل", "sleep": "نام", "swim": "سبح"}

print(Dic)
print(type(Dic))
```

أستخرج الصيغة العامة الأُخرى لتعريف القاموس.

4- المجموعات (Sets):

تُعرَّف المجموعات بأنَّها هياكل بيانات في لغة البرمجة بايثون تحتوي على مجموعة من الكائنات.

مثال:

يُبيِّن البرنامج الآتي كيف يُمكِن إنشاء مجموعة، ثمَّ طباعتها وحذف العناصر المُتكرِّرة.

```
# تعريف المجموعة
sports = {"Football", "Basketball", "Rugby", "Volyball", "Rugby",
"Tennis", "Swimming"}

# تطبع المجموعة وتحذف العناصر المتشابهة
print(sports)

# اختبار لوجود عنصر ضمن المجموعة
print("Tennis" in sports)
print("Diving" in sports)

# تعريف مجموعة جديدة من نصوص تحتوي على تكرار
a = set("abaabcd")
print(a)
```

بعد تنفيذ البرنامج، سيظهر الناتج الآتي:

```
{'Football', 'Tennis', 'Basketball', 'Volyball', 'Swimming', 'Rugby'}
True
False
{'d', 'c', 'b', 'a'}
```

مزايا المجموعات (Sets):

أ. **عدم الترتيب**: لا يُشترَط ترتيب العناصر في المجموعات؛ ما يُفسِّر سبب اختلاف ترتيب العناصر أثناء عملية الطباعة.

ب. **الثبات وعدم القابلية للتكرار**: لا يُسمَح بوجود عناصر مُكرَّرة في المجموعات؛ إذ تُحذَف التكرارات تلقائيًّا عند إضافة العناصر.

ج. **الثبات وعدم القابلية للتعديل** (Immutable): يُمكِن إضافة عناصر إلى أيَّة مجموعة أو حذف عناصر منها، ولكنْ يتعذَّر تعديل العناصر الفردية داخل المجموعة.

د. **دعم العمليات الرياضية**: تُعَدُّ المجموعات خيارًا مثاليًّا لدعم العمليات الرياضية والحسابية، مثل: التقاطع، والاتحاد، والفَرْق بين المجموعات.

ه. **سرعة الأداء**: تُعَدُّ المجموعات أسرع من القوائم في ما يخصُّ عمليات البحث والمقارنة؛ نظرًا إلى عدم وجود ترتيب لعناصرها.

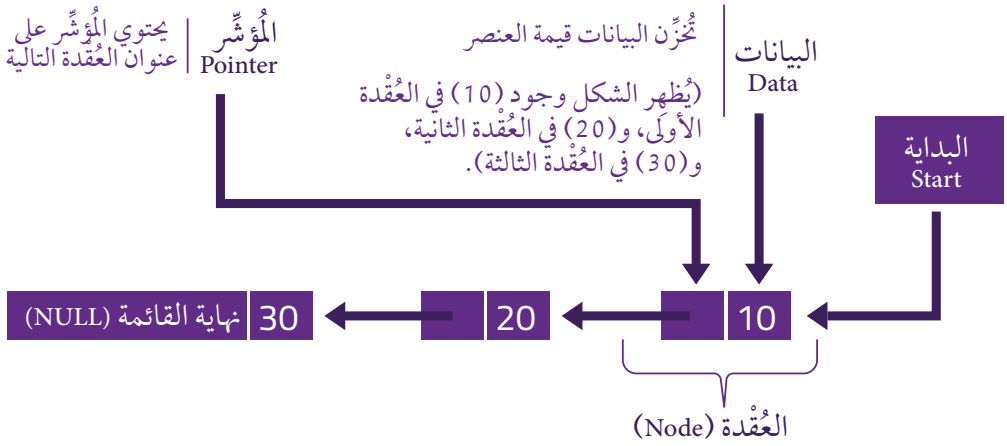
5- نوع البيانات المُجرَّدة (Abstract Data Type: ADT):

تعرَّفْتُ آنفًا كيف تُخزَّن البيانات في بعض التراكيب بأشكال مختلفة تبعًا لنوع هذه البيانات، والغرض من استخدامها. ولكنْ تعتمد طريقة تخزين البيانات على مجموعة من العوامل، أبرزها: الحاجة إلى تعديل البيانات، وإمكانية تكرار العناصر، وطبيعة العمليات التي يُراد تنفيذها. فكل نوع من تراكيب البيانات (مثل: القوائم، والصفوف، والقواميس، والمجموعات) يمتاز بخصائص مُعيَّنة تُناسِب مُتطلَّبات مُحدَّدة في البرمجة.

يُعرَّف نوع البيانات المُجرَّدة (ADT) بأنَّه نموذج يُركِّز على السلوكات، أو العمليات (مثل: القراءة، والتعديل)، أو العمليات الحسابية الخاصة بالبيانات، بدلًا من التركيز على كيفية تخزين البيانات.
من الأمثلة على نوع البيانات المُجرَّد (ADT): **القوائم المُترابِطة** (Linked List)، **والمكدس** (Stack).

أوَّلًا- القوائم المُترابِطة (Linked List):

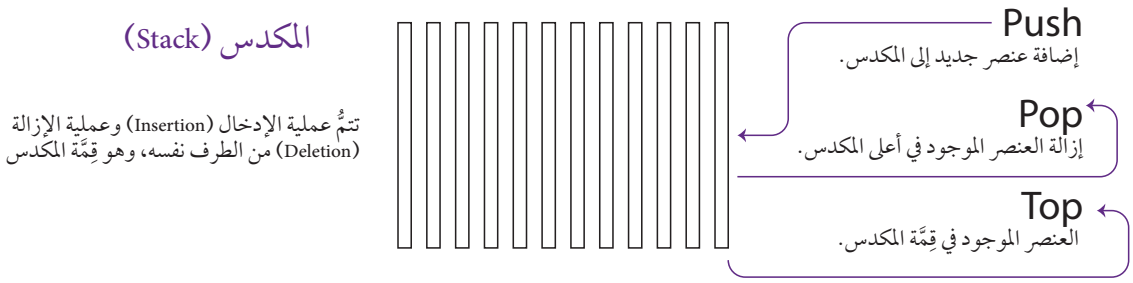
تُخزَّن البيانات في القوائم المُترابِطة على هيئة عُقَد، وتحتوي كل عُقْدة (Node) من هذه العُقَد على مرجعية العُقْدة التي تليها كما هو مُبيَّن في الشكل (3-1). تُستخدَم القوائم المُترابِطة في التطبيقات التي تتغيَّر حجوم بياناتها، وفي عمليات التخزين المُؤقَّت. كذلك تُستخدَم هذه القوائم في تتبُّع العمليات الخاصة بأنظمة التشغيل، وفي تتبُّع الصفحات المُحرَّرة، إضافةً إلى الرسوم التي تُستخدَم في تمثيل الأشكال الهندسية، مثل: النقاط، والخطوط، وحدود الكائنات ذات الأبعاد الثنائية.



الشكل (3-1): بِنية القوائم المُترابِطة (قائمة مُرتبِطة أُحادية).

ثانيًا- المكدس (The Stack ADT):

نوع من أنواع البيانات المُجرَّدة، يتعامل مع البيانات بطريقة (Last-In /First-Out: LIFO)؛ أيْ إنَّ آخر عنصر يدخل في المكدس هو أوَّل عنصر يخرج منه. وهذا يعني أنَّ لهذه القائمة بوّابة واحدة فقط تُستخدَم مُدخَلًا ومُخرَجًا كما هو مُبيَّن في الشكل (3- 2).



الشكل (3-2): بِنية المكدس.

عند تنفيذ أكثر من أمر في برنامج مُحرِّر النصوص (Word)، فإنَّ العمليات تُخزَّن واحدة تلو الأُخرى. وفي حال أُعطِي أمر بالتراجع، فإنَّ آخر عملية مُنفَّذة سيتمُّ التراجع عنها. وهذا الأمر مُشابِه لِما يحدث عند استخدام المُتصفِّحات حين تُخزَّن الصفحات السابقة والصفحات اللاحقة. أمّا في البرمجة فيُستخدَم المكدس عند استدعاءات الدوال. فمثلًا، إذا كان تنفيذ الدالَّة يعتمد على قِيَم محلية، فإنَّ المكدس يحتفظ بهذه القِيَم مثلما يحتفظ بالتعليمات التي يجب أنْ يعود إليها عند انتهاء عملية التنفيذ، وكذا الحال بالنسبة إلى السجلّات. وبعد الانتهاء من تنفيذ الدالَّة، تُحذَف المعلومات الخاصة بهذه الدالَّة لاستئناف العمل. والشيء نفسه ينطبق على الألعاب الإلكترونية؛ إذ يتتبَّع اللاعب جميع الخطوات خطوة خطوة، ثمَّ يتراجع عن آخر خطوة نفَّذها إذا واجه إحدى المشكلات أثناء اللعب، وهكذا.

في لغة البرمجة بايثون، يُمكِن استخدام المكدس (Stack) بواسطة القوائم، أو مكتبة (Collections)، أو أحد تطبيقات (Deque).

في ما يأتي توضيح للعمليات التي يُمكِن تنفيذها على المكدس:

`push`: يُستخدَم الأمر append (item) بدلًا من الدالَّة ()push لإضافة عنصر جديد إلى المكدس (Stack)، وذلك في نهاية القائمة التي تُمثِّل المكدس (Stack).

```
stack = []
stack.push( )
print(stack)
```

`pop`: تُستخدَم الدالَّة ()pop لحذف العنصر الأخير من المكدس (Stack).

```
stack.pop()
print(stack)
```

`peek or top` : تُســـتخدَم هذه الدالَّة في المكدس (Stack) لإرجاع العنصر الموجود في أعلى المكدس (آخر عنصر مضاف) دون إزالته من المكدس (Stack).

```
top_element=stack[]
print(top_element)
```

`size`: تُستخدَم الدالَّة ()len لإرجاع عدد العناصر في المكدس (Stack).

```
size=len(stack)
print(size)
```

`Is_empty`: تُســـتخدَم هذه الدالَّة مع الشـــرط not. وفي حال كانت القائمة فارغة، يُرجَع الشرط (True)، أمّا إذا كانت القائمة تحتوي على عناصر، فإنَّه يُرجَع (False).

```
stack = []
#  التحقق باستخدام الشرط
is_empty = not stack
print ("Is the stack empty?", is_empty)
```

المواطنة الرقمية:

- **محو الأُمِّية الرقمية** (Digital Literacy): أتعلَّم كيف أختار هيكل البيانات المناسب (مثل: القوائم للقِيَم المُتكرِّرة، والقواميس للعلاقات بين المفاتيح والقِيَم)، وأعمل على توعية المُستخدِمين بأهمية تنظيم البيانات جيِّدًا لتحسين مستوى الكفاءة والأداء.
- **المسؤولية الرقمية** (Digital Responsibility): أعي أهمية الاستخدام الصحيح للبيانات المُخزَّنة وما قد يحدث إذا أُسيء استخدامها، وأحرص على استخدام نوع البيانات المناسب لمشروعي؛ كي أستفيد من الموارد المُتوافِرة أيَّما استفادة.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أُوضِّح المقصود بكلٍّ ممّا يأتي:

1. هياكل البيانات.
2. البيانات المُجرَّدة.

السؤال الثاني: أُحدِّد نوع البيانات في كل جملة ممّا يأتي:

1. تُخزَّن فيها البيانات على هيئة عُقَد.
2. مجموعات مُتسلسِلة ومُترتِّبة من الكائنات، يُمكِن الوصول إلى العناصر داخلها من دون تغييرها أو إضافة عناصر جديدة إليها، وتُكتَب هذه العناصر بين أقواس دائرية.
3. تُخزَّن محتوياته في صورة أزواج من المفاتيح والقِيَم.
4. تسلسلات من القِيَم مُرتَّبة ومُرتبِطة معًا بمعنى وظيفي مُشترَك.
5. مجموعة من الكائنات غير مُرتَّبة، وغير مُكرَّرة، وغير قابلة للتعديل.
6. أوَّل عنصر يدخل إلى القائمة هو أوَّل عنصر يخرج منها.

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أستخدم لغة البرمجة بايثون في تعريف قاموس يحتوي على أنواع البيانات التي تناولها هذا الدرس، وتعريف كلٍّ منها.

السؤال الثاني: أكتشف الخطأ في كل مقطع برمجي ممّا يأتي، ثمَّ أُحدِّد طريقة تعديله:

-1

```
Time_dictionary = dict()
Time_dictionary(time) = وقت
Time_dictionary {second} = ثانية
Time_dictionary <minute> = دقيقة
prtn Time_dictionary
```

-2

```
b = [Amman, Jarash, Zarqa, Aqaba, Mafraq] # b is list
b.append (Ajloun)
print (b)
```

الدرس الرابع

الدالَّة الراجعة (Recursion Function)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس الدالَّة الراجعة، وأتعلَّم كيف أستخدمها في إيجاد ناتج برنامج يتكوَّن من حلقات مُتكرِّرة من الاستدعاء نفسه.

المفاهيم والمصطلحات:

الدالَّة الراجعة (Recursion)، مضروب العدد (Factorial).

نتاجات التعلُّم (Learning Outcomes):

- أُعرِّف الدالَّة الراجعة (الدالَّة التي تستدعي نفسها).
- أُحلِّل المشكلات التي يحتاج تمثيلها إلى دالَّة راجعة.
- أستخدم لغة البرمجة بايثون في إيجاد حلول تتناسب مع الدالَّة الراجعة.

تعرَّفْتُ سابقًا مضروب العدد، وكيف يُمكِن إيجاده باستخدام الحلقات في لغة البرمجة بايثون. كذلك تعرَّفْتُ طريقة تعريف الدوال، وكيفية استدعائها في البرنامج. والآن سأتعرَّف نوعًا جديدًا من أنواع الدوال، هو الدالَّة الراجعة.

نشاط تمهيدي

أدرس جيِّدًا البرنامج الظاهر في الشكل (4-1)، ثمَّ أُجيب عن الأسئلة التي تليه:

```
def Rectangle_Area(height, width):
    Area = height * width
    return Area

print(Rectangle_Area(7, 9))
```

الشكل (4-1): برنامج كُتِب بلغة البرمجة بايثون.

- ما وظيفة هذا البرنامج؟
- ماذا نتج من تنفيذ البرنامج؟
- ما التعديل الذي يجب عمله لتمكين المُستخدِم من إدخال قِيَم للمُتغيِّر (width) والمُتغيِّر (height)؟

أُقارِن إجابتي بإجابات زملائي/ زميلاتي في الصف.

الدالَّة الراجعة (Recursion):

تُعرَّف الدالَّة الراجعة بأنَّها دالَّة قادرة على حَلِّ مسألة مُعيَّنة عن طريق استدعاء نفسها مَرّات عديدة، وأداء المهمة نفسها كل مَرَّة باستخدام عناصر (Parameters) مختلفة.

يشيع استخدام الاستدعاء الذاتي في حَلِّ المشكلات التي يُمكِن تقسيمها إلى مشكلات فرعية صغيرة ذات هيكل مُشابِه لهيكل المشكلة الأصلية، ومن ثَمَّ تُطبَّق العملية نفسها على جزء أصغر من البيانات والمعاملات. وفي جميع الأحوال، يجب أنْ تحتوي الدالَّة على حالة أساسية يتوقَّف عندها الاستدعاء الذاتي؛ لكيلا يدخل هذا الاستدعاء في سلسلة لانهائية من الاستدعاءات.

يُستخدَم الاستدعاء الذاتي في حَلِّ بعض المشكلات البرمجية بدلًا من الدوران والتكرار؛ نظرًا إلى سهولة كتابة البرنامج ووضوحه. ومن أشهر الأمثلة على هذا النوع من الاستدعاءات: حساب المضروب، حيث إنَّ:

$5! = 5 * 4!$

$4! = 4 * 3!$

$3! = 3 * 2!$

$2! = 2 * 1!$

$1! = 1$

أيْ إنَّ مضروب العدد 5 يساوي حاصل ضرب (5) في مضروب العدد (4)، ومضروب العدد (4) يساوي حاصل ضرب العدد (4) في مضروب العدد (3)، وهكذا وصولًا إلى الحالة الأساسية، وهي مضروب العدد (1)، وهي حالة لا تتطلَّب استدعاءً آخرَ للدالَّة.

الصيغة العامة للدالَّة الراجعة (Recursive Function) في لغة البرمجة بايثون (Python):

يُبيِّن البرنامج الآتي الصيغة العامة للدالَّة الراجعة (Recursive Function) في لغة البرمجة بايثون (Python):

```
def recursive_call(N):
    if base_case:
        return base_case_value
    else:
        return recursive_call(N - 1)
```

حيث:

def: كلمة محجوزة تُستخدَم في تعريف دالَّة جديدة.

recusive_call: اسم الدالَّة التي يجب أنْ يكون لها مُدخَلات موضوعة بين أقواس.

base_case: القيمة الأساسية التي تجعل الدالَّة تتوقَّف، ومن دونها يصبح الدوران لانهائيًّا.

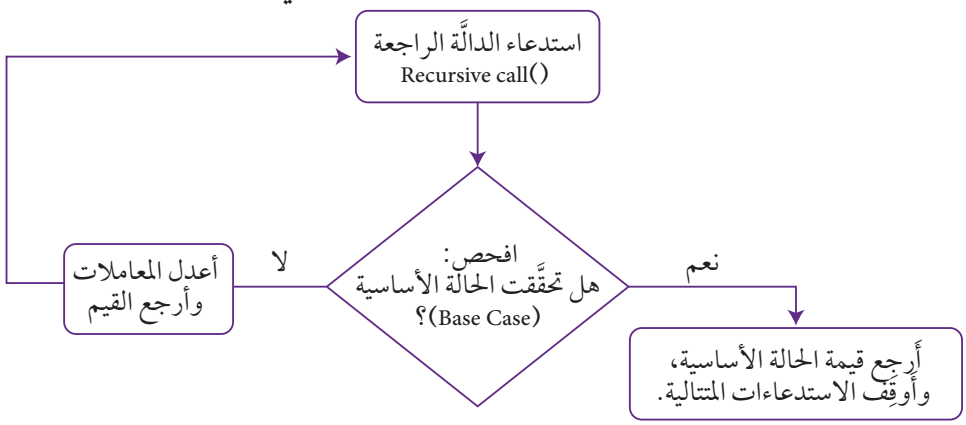
base_case_value: القيمة التي تعود إلى البرنامج عند تحقُّق شرط الوصول إلى القيمة الأساسية.

أُناقِش

هل يُمكِن التبديل بين جزأي (If) بحيث تصبح القيمة الراجعة عند الوصول إلى القيمة الأساسية (base_case_ value) في جزئية (else)؟ ما التعديل الذي يجب إجراؤه على البرنامج إنْ كان ذلك مُمكِنًا؟

أُناقِش زملائي/ زميلاتي في إجابتي، ونتبادل معًا الخبرات للتحقُّق من صِحَّة الحَلِّ.

يُمكِن تمثيل مُخطَّط سَيْر العمليات للدالَّة الراجعة كما هو مُبيَّن في الشكل (4-2).

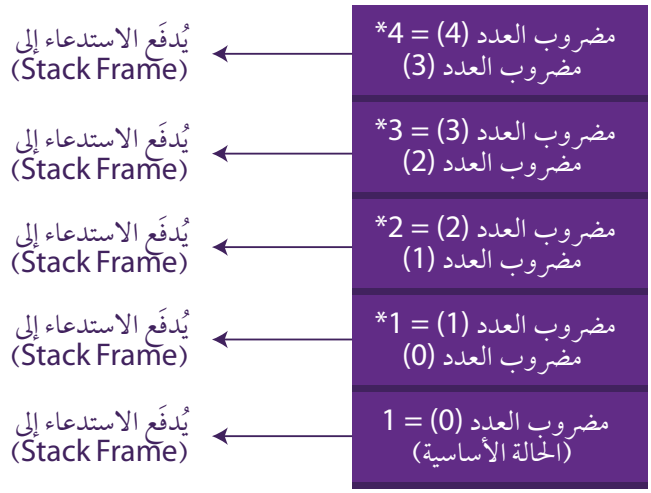


الشكل (4-2): مُخطَّط سَيْر العمليات للدالَّة الراجعة (Recursive_call).

استخدام المكدس في تتبُّع حالة كل استدعاء للدالَّة الراجعة:

تعرَّفْتُ أنَّ المكدس نوع من هياكل البيانات، يعتمد في عمله على مبدأ مفاده أنَّ آخر قيمة تدخل هي أوَّل قيمة تخرج. وعند تنفيذ الاستدعاء الذاتي (Recursion) يُستخدَم المكدس في تتبُّع حالة كل استدعاء للدالَّة الراجعة، بحيث يتمكَّن البرنامج من العودة إلى النقطة الصحيحة بعد انتهاء عملية التنفيذ. عند استدعاء البرنامج لنفسه، يتمُّ إنشاء إطار جديد يحتوي على المعطيات وقِيَم المُتغيِّرات المحلية والموضع الذي تُتابَع منه عملية التنفيذ. وبعد إنهاء الدالَّة، يضاف الإطار إلى المكدس كما هو مُبيَّن في الشكل (4-3). وما إنْ تصل الدالَّة إلى الحالة الأساسية أو تنتهي عملية التنفيذ، حتّى يُزال الإطار من المكدس، وتُعاد مهمة التحكُّم إلى الإطار السابق في المكدس.

يُذكَر أنَّ استدعاء الدالَّة لنفسها عددًا كبيرًا من المَرّات يُفْضي إلى وضع عدد كبير من الإطارات في المكدس؛ ما يزيد من استخدام حجم الذاكرة. وفي بعض الحالات، قد يتسبَّب ذلك في تجاوز حجم الذاكرة المُتوافِرة؛ ما يؤدّي إلى توقُّف البرنامج عن العمل.



الشكل (4-3): مثال على آليَّة عمل الدوال الراجعة (Recursive Functions)، واستخدام (Stack Frame) في عملية التنفيذ.

نشاط جماعي

- أتعاون مع أفراد مجموعتي على تنفيذ ما يأتي:
- زيارة الموقع الإلكتروني الآتي، أو مسح الرمز سريع الاستجابة المجاور:
- https: / /pythontutor.com /python-compiler.html#mode=edit
- كتابة المقطع البرمجي الخاص بمثال المضروب في شاشة بايثون.
- تنفيذ البرنامج، ومشاهدة التمثيل المرئي لتنفيذ البرنامج، وكيف يتمُّ الاستدعاء الذاتي لدالَّة المضروب.
- أُدوِّن الملاحظات، ثمَّ أُشارِك أفراد المجموعات الأُخرى في ما توصَّلْتُ إليه من خلال مجموعتي، ونتبادل معًا الأفكار والتغذية الراجعة.

مثال:

يُمثِّل الشكل (4-4) برنامجًا يُمكِنه إيجاد مجموع الأعداد من 1 إلى N باستخدام دالَّة (recursion).

```
N = int(input("Enter a positive number: "))

def sum_recursion(N):
    if (N > 0):
        result = N + sum_recursion(N - 1)
        print(result)
    else:
        result = 0
    return result


print("The summation of the first", N, "numbers =", sum_recursion(N))
```

الشكل (4-4): برنامج لحساب مجموع الأعداد من 1 إلى N.

بتتبُّع البرنامج في المثال السابق، يتبيَّن ما يأتي:

إدخال المُستخدِم قيمة: يُدخِل المُستخدِم رقمًا موجبًا (لنفترض أنَّ الرقم المُدخَل هو (7)).

1. استدعاء الدالَّة sum_recursion (7):
   يُمرَّر الرقم(7) إلى الدالَّة sum_recursion؛ لإيجاد مجموع الأعداد من (1) إلى (7).
   يضاف إطار جديد إلى المكدس (Stack Frame) يُمثِّل استدعاء الدالَّة sum_recursion (7).
2. اختبار الشرط داخل الدالَّة: يُختبَر الشرط (0<7)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:
   result = sum_recursion (6) +(7)
3. استدعاء الدالَّة sum_recursion (6)، وإضافة إطار جديد إلى المكدس.
4. اختبار الشرط (0<6)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:
   result = sum_recursion (5) +(6)

5. استدعاء الدالَّة sum_recursion (5)، وإضافة إطار جديد إلى المكدس.
6. اختبار الشرط (0 < 5)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:

result = sum_recursion (4) + 5

7. استدعاء الدالَّة sum_recursion (4)، وإضافة إطار جديد إلى المكدس.
8. اختبار الشرط (0 < 4)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:

result = sum_recursion (3) + 4

9. استدعاء الدالَّة sum_recursion (3)، وإضافة إطار جديد إلى المكدس.
10. اختبار الشرط (0 < 3)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:

result = sum_recursion (2) + 3

11. استدعاء الدالَّة sum_recursion (2)، وإضافة إطار جديد إلى المكدس.
12. اختبار الشرط (0 < 2)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:

result = sum_recursion (1) + 2

13. استدعاء الدالَّة sum_recursion (1)، وإضافة إطار جديد إلى المكدس.
14. اختبار الشرط (0 < 1)، وإذا كانت الإجابة (نعم)، فإنَّه يتمُّ حساب النتيجة:

result = sum_recursion (0) + 1

15. استدعاء الدالَّة sum_recursion (0)، وإضافة إطار جديد إلى المكدس.
16. الاستدعاء الأساسي: sum_recursion (0).
17. اختبار الشرط (0 < 0)، وإذا كانت الإجابة (لا)، فإنَّ الكود يُنفَّذ داخل فرع (else).

result = 0

أُلاحِظ أنَّ هذه القيمة هي قيمة الأساس.

18. إرجاع القيمة 0، وإزالة إطار sum_recursion (0) من المكدس.
19. يعمل المكدس وَفق مبدأ أنَّ آخر عملية تدخله هي أوَّل عملية ستُنفَّذ، ومن ثَمَّ سيعود كل مَرَّة خطوة إلى الوراء. أنظر الجدول (4-1).

عند العودة إلى sum_recursion (1)، فإنَّه يتمُّ حساب النتيجة:

result = 0 + 1 = 1

تتمُّ طباعة النتيجة 1.

إزالة إطار sum_recursion (1) من المكدس.

عند العودة إلى sum_recursion (2)، فإنَّه يتمُّ حساب النتيجة:

result = 1 + 2 = 3

تتمُّ طباعة النتيجة 3.

إزالة إطار (2) sum_recursion من المكدس.

وهكذا، حتّى نصل إلى النتيجة: (7) sum_recursion.

الجدول (4-1): خطوات حساب القيمة النهائية باستخدام الدالَّة الراجعة.

الخطوة	العملية الحالية	القيمة الراجعة من الدالَّة	النتيجة النهائية (result)
1	sum_recursion (0) + 1 = result	0	1
2	sum_recursion (1) + 2 = result	1	3
3	sum_recursion (2) + 3 = result	3	6
4	sum_recursion (3) + 4 = result	6	10
5	sum_recursion (5) + 5 = result	10	15
6	sum_recursion (5) + 6 = result	15	21
7	sum_recursion (6) + 7 = result	21	28

بتنفيذ الأوامر السابقة الواردة في الشكل (4-4) تكون الدالَّة الراجعة قد أنهت إيجاد مجموع الأعداد من (1) إلى (7)، فتعود إلى البرنامج الأصلي لتبدأ عملية التنفيذ، وتعود تحديدًا إلى جملة:

```
print("The summation of the first", N, "numbers =", sum_recursion(N))
```

يُبيِّن الشكل (4-5) النتيجة النهائية بعد تنفيذ البرنامج في شاشة بايثون:

```
Enter a positive number: 7
1
3
6
10
15
21
28
The summation of the first 7 numbers = 28
```

الشكل (4-5): ناتج تنفيذ sum_recursion().

نشاط جماعي

أرسم - بالتعاون مع أفراد مجموعتي - مُخطَّط سَيْر العمليات للمثال السابق، ثمَّ أُشارِك أفراد المجموعات الأُخرى في ما توصَّلْتُ إليه من خلال مجموعتي، ونتبادل معًا الأفكار والتغذية الراجعة.

نشاط فردي

أتتبَّع البرنامج الآتي، وأتوقَّع ناتج تنفيذه، ثمَّ أُجيب عن الأسئلة التي تليه:

```
1  def fib(n):
2      if n == 0:
3          return 0
4      elif n == 1:
5          return 1
6      return fib(n - 1) + fib(n - 2)
7
8  for i in range(10):
9      print(fib(i))
10
```

- ما وظيفة هذا البرنامج؟
- أُنفِّذ البرنامج باستخدام مُفسِّر لغة البرمجة بايثون.
- أُقارِن ناتج التنفيذ بما توقَّعته.
- أُقارِن إجابتي بإجابات زملائي/ زميلاتي في الصف.

إضاءة

() Factorial: دالَّة جاهزة ضمن مكتبة (math)، وهي تُستخدَم في إيجاد مضروب العدد، لكنَّها ليست دالَّة راجعة.

نشاط فردي

أكتب برنامجًا بلغة البرمجة بايثون لحساب مضروب عدد باستخدام الدالَّة الراجعة.

أَحسُب مضروب العدد (8) يدويًّا: أتتبَّع خطوات تنفيذ البرنامج يدويًّا خطوة خطوة، بَدْءًا بالحالة الأساسية (n = 0)، وأُوضِّح كل خطوة أثناء تنفيذ الدالَّة الراجعة.

أُنفِّذ البرنامج المكتوب بلغة البرمجة بايثون، ثمَّ أُقارِن النتائج التي توصَّلْتُ إليها يدويًّا بالنتائج التي توصَّلْتُ إليها من تنفيذ البرنامج.

أتحقَّق من إجابتي بطريقة أُخرى، وذلك بتنفيذ البرنامج الآتي:

```
import math
print(math.factorial(8))
```

هل الإجابتان مُتطابِقتان؟ أُبرِّر إجابتي.

نشاط جماعي

أُنفِّذ – بالتعاون مع أفراد مجموعتي – المهمة الآتية:

- كتابة برنامج تُستخدَم فيه الدالَّة الراجعة لإيجاد أكبر عدد في قائمة.
- تتبُّع البرنامج يدويًّا خطوة خطوة، وتوضيح كيفية عمل الدالَّة الراجعة في كل خطوة، وصولًا إلى النتيجة النهائية.
- تنفيذ البرنامج باستخدام مُفسِّر لغة البرمجة بايثون؛ لاستكشاف النتيجة، والتحقُّق من صِحَّتها.

نُناقِش معًا: أيُّهما أسهل لحَلِّ هذه المسألة: استخدام الدالَّة الراجعة مع الجملة الشرطية أم استخدام جملة الدوران مثل: (for)، و(while)؟ نُبرِّر إجابتنا.

نُشارِك أفراد المجموعات الأُخرى في ما نتوصَّل إليه من نتائج، ونتبادل معًا التغذية الراجعة.

المواطنة الرقمية:

- **المسؤولية الرقمية** (Digital Responsibility): أحرص على عدم استخدام الدوال الراجعة في الحالات التي قد تُسبِّب استهلاكًا مُفرِطًا للذاكرة أو الوقت، وبخاصة في الأنظمة المُشتَرَكة أو الأنظمة الحسّاسة.
- **محو الأُمِّية الرقمية** (Digital Literacy): أُوضِّح أهمية الدوال الراجعة وكيفية استخدامها بطرائق صحيحة وفاعلة، وأُشجِّع تعلُّم مبادئ البرمجة المسؤولة لتطوير مهارات البرمجة المستدامة.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: ما المقصود بالدالَّة الراجعة؟ وما الحالة الأساسية في الدوال الراجعة؟

السؤال الثاني: أستخلص إيجابيات استخدام الدالَّة الراجعة وسلبياتها.

السؤال الثالث: أكتب الصيغة العامة للدوال الراجعة في لغة البرمجة بايثون.

السؤال الرابع: ما الفَرْق بين الدالَّة الراجعة وجملة الدوران، مثل: (for)، و(while)؟

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أَجِد ناتج تنفيذ البرنامج الآتي عند إدخال كلمة (Hello):

```
def reverse_string(s):
    if len(s) == 0:
        return s
    return s[-1] + reverse_string(s[:-1])

Word = input("Enter a word: ")

print(reverse_string(Word))
```

السؤال الثاني: أكتب دالَّة راجعة جديدة لإيجاد مجموع الأرقام الفردية فقط بين 1 وN.

السؤال الثالث: أتتبَّع يدويًّا الخطوات التي يَمُرُّ بها المكدس عند تنفيذ الدالَّة الراجعة لحساب مضروب العدد (2).

السؤال الرابع: أكتب برنامجًا لإيجاد أقلِّ عدد في قائمة ما باستخدام الدوال الراجعة، وأُوضِّح كيف يُمكِن معالجة كل خطوة يدويًّا.

السؤال الخامس: أكتب برنامجًا لإيجاد ناتج ضرب الأرقام في قائمة ما باستخدام دالَّة راجعة، وأُوضِّح كيف يُمكِن معالجة كل خطوة يدويًّا.

السؤال السادس: أَجِد الخطأ في المقاطع البرمجية الآتية، ثمَّ أُصحِّحه، علمًا بأنَّ هذه المقاطع تحتوي على دالَّة راجعة:

-1

```
def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n - 1)

print(factorial(-5))
```

-2

```
def linear_search(lst, target, index=0):
    if index == len(lst):
        return -1
    if lst[index] == target:
        return index
    return linear_search(lst, target)

print(linear_search([1, 2, 3, 4, 5], 3))
```

-3

```
def sum_list(lst):
    if len(lst) == 0:
        return 0
    return lst[0] + sum_list(lst)

print(sum_list([1, 2, 3, 4, 5]))
```

الدرس الخامس

الحُزَم (Packages)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس ماهيَّة الحُزَم البرمجية وأنواعها، وأتعلَّم كيف أستخدمها وكيفية إنشاء حُزْمة برمجية خاصة.

المفاهيم والمصطلحات:

الحُزَم (Packages)، الوحدة (Module)، الحُزَم الخارجية (Third-Party Packages)، الحُزَم التي طوَّرها المُستخدِم (User-Defined Packages)، الوحدة المستدعاة (Imported Module)، الوحدة المُستدعِية (Calling Module)، الدليل (Directory)، الأسماء المستعارة (Alias) ، الدالَّة (Function)، الروتين الفرعي (Subroutine).

نتاجات التعلُّم (Learning Outcomes):

- أختار الطريقة المثلى لتمثيل أجزاء المشكلة في صورة جمل برمجية، أو روتين فرعي، أو وحدات، أو كائنات.
- **أستخدم لغة البرمجة بايثون في تطوير حلول برمجية للمشكلات باستخدام جمل برمجية، مثل:** الروتينات الفرعية، والوحدات، والكائنات.
- أُحدِّد الأماكن المناسبة لاستخدام الدوال والروتينات الفرعية في الحَلِّ البرمجي.
- أستخدم لغة البرمجة بايثون في تعريف مجموعات من معاملات لمُخرَجات الروتين الفرعي.
- أُنشِئ دوال وروتينات فرعية باستخدام لغة البرمجة بايثون.

- أكتب جملًا برمجيةً وروتيناتٍ فرعيةً ووحداتٍ باستخدام لغة البرمجة بايثون.
- أُضيف مكتبات برمجية إلى البرامج، وأُوظِّفها في أغراض مُحدَّدة.
- أُوضِّح مفهوم الأنماط البرمجية.
- أستخدم لغة البرمجة بايثون في إعداد حُزْمة برمجية.

تعرَّفْتُ سابقًا أنَّ الدالَّة البرمجية (Function) تُمثِّل مقطعًا برمجيًّا له اسم يؤدّي وظيفة ما، ويُمكِن استدعاؤه باستخدام اسمه. كذلك تعرَّفْتُ أنَّ كل مجموعة من الدوال التي تشترك معًا في الغرض والاستخدام نفسيهما قد جُمِعت في وحدات (Modules).

نشاط تمهيدي

أتأمَّل المقطع البرمجي الظاهر في الشكل (5-1)، ثمَّ أُوضِّح الجمل الموجودة فيه. ما ناتج تنفيذ البرنامج؟ ماذا تُسمّى (math) في لغة البرمجة بايثون؟

```
import math

print(math.sqrt(16))
print(math.pi)
print(math.factorial(5))
```

الشكل (5-1): برنامج في لغة البرمجة بايثون.

الوحدة (Module) والحُزَم (Packages):

الوحدة (Module) في لغة البرمجة بايثون هي ملف يحتوي على مقاطع برمجية (Code) مكتوبة مُسبَّقًا، ومن المُمكِن استدعاؤها وإعادة استخدامها في برامج أُخرى. من الأمثلة على الوحدات التي استخدمْتُها سابقًا: وحدة (math)، ووحدة (time).

أمّا الحُزْمة (Package) فهي طريقة لتنظيم الوحدات المُرتبِطة معًا في تسلسل هرمي، وهي تُعَدُّ دليلًا (Directory) يحتوي على وحدات مُتعدِّدة، وحُزَم فرعية، وموارد إضافية، مثل: ملفات البيانات، والوثائق. ويُمكِن تعرُّف الدليل - بوصفه حُزْمة- من خلال وجود ملف (init.py).

أنواع الحُزَم:

توجد أنواع مختلفة من الحُزَم، أبرزها:

1. **حُزَم جاهزة مُدمَجة** (Built-in Packages) **تُوفِّرها لغة البرمجة بايثون، مثل**: (math)، و (time).
2. **حُزَم خارجية** (Third-Party Packages) تُطوِّرها أطراف ثالثة، وهي تُوفِّر مزايا إضافية لمُطوِّري البرامج. وهذا النوع من الحُزَم يُثبَّت باستخدام مدير الحُزَم (pip)، **مثل**: (numpy)، و (pandas).
3. **حُزَم مُخصَّصة** (**User-Defined Packages**) يُطوِّرها المُستخدِم لتُلائِم مُتطلَّبات خاصة في برنامجه.

تُعَدُّ الحُزَم في لغة البرمجة بايثون أداة فاعلة لتنظيم المقطع البرمجي (Code)؛ إذ تتيح تقسيم الكود إلى أجزاء صغيرة ومستقلة ومُنظَّمة؛ ما يُسهِّل عملية تصحيح الأخطاء، واختبار البرنامج وصيانته، إضافةً إلى تسهيل إعادة استخدام الكود في مشروعات مُتعدِّدة عن طريق استيرادها؛ ومن ثَمَّ يُمكِن للحُزَم المختلفة أنْ تحتوي على وحدات تحمل الاسم نفسه من دون أيِّ مشكلات. كذلك، فإنَّ الحُزَم تجعل المشروعات أكثر قابلية للتوسُّع من خلال دعم التطوير التعاوني بين الفِرَق المختلفة، وتُسهِّل عملية التعامل مع التعديلات المستقبلية. وبذلك، تُسهِم الحُزَم في تحسين كفاءة المشروع، وتزيد من مرونته.

إنشاء الوحدات (Modules Construction):

يُمكِن للمُستخدِم إنشاء وحدة برمجية خاصة به باستخدام لغة البرمجة بايثون عن طريق كتابة الدوال والوظائف داخل ملف بايثون (Python File)، ثمَّ حفظه في صورة ملف يكون امتداده (.py)، ويحمل اسمًا يُمثِّل الوحدة. بعد ذلك يُمكِن إنشاء ملف آخر أساسي، ثمَّ حفظه في المُجلَّد نفسه، واستدعاء الوحدة باستخدام الأمر (import.). وبهذه الطريقة، يُمكِن استخدام الدوال والوظائف المُعرَّفة في الوحدة داخل الملف الأساسي؛ ما يُسهِّل تنظيم الكود، ويتيح إعادة استخدامه في مشروعات عديدة.

مثال:

يُبيِّن هذا المثال خطوات إنشاء وحدة برمجية لحساب مساحة مستطيل وكيفية استدعائها.

إنشاء وحدة لحساب مساحة مستطيل:

- أفتح مُحرِّر بايثون، ثمَّ أكتب المقطع البرمجي الآتي الذي يُعرِّف وحدة باسم (area) لحساب مساحة المستطيل.

```
1 def area (H, W):
2     return H*W
```

- أحفظ الملف باسم (Rarea.py) في مُجلَّد خاص.

- **أفتح برنامج بايثون جديدًا، ثمَّ أكتب البرنامج الآتي:**

```
import Rarea

a=Rarea.area (8, 9)
print("Rectange Area=",a)
```

- أحفظ الملف باسم (area_main.py) في المُجلَّد نفسه الذي حفظْتُ الكود السابق فيه، ثمَّ أعمل على تنفيذه.
- **يظهر الناتج الآتي في شاشة** (IDLE Shell):

```
Rectange Area= 72
```

أُلاحِظ أنَّ ملف (Rarea.py) يُسمّى **الوحدة المستدعاة** (Imported Module)، وأنَّ الملف (area_main.py) الذي يتضمَّن جملة الاستدعاء يُسمّى **الوحدة المُستدعِية** (Calling Module).

نشاط جماعي

أُنفِّذ – بالتعاون مع أفراد مجموعتي – المهمة الآتية:

- فتح مُفسِّر لغة البرمجة بايثون، ثمَّ تعريف وحدة خاصة بالجمع، وإطلاق اسم (Add) عليها، وحفظها باسم (Addition).
- تعريف وحدة خاصة بالطرح، وإطلاق اسم (Subtract) عليها، وحفظها باسم (Subtraction) في المُجلَّد نفسه الذي حُفِظت وحدة (Addition) داخله.
- تعريف وحدة خاصة بالضرب، وإطلاق اسم (Multiply) عليها، وحفظها باسم (Multiplication) في المُجلَّد نفسه الذي حُفِظت وحدة (Addition) داخله.
- تعريف وحدة خاصة بالقسمة، وإطلاق اسم (Divide) عليها، وحفظها باسم (Division) في المُجلَّد نفسه الذي حُفِظت وحدة (Addition) داخله.
- **كتابة الكود المستدعي بحيث يحتوي على ما يأتي:**

أ- جمل إدخال لإدخال الأعداد التي يُراد إجراء العمليات الحسابية عليها.

ب- جملة إدخال لإدخال الرمز الخاص بالعملية الحسابية التي يُراد إجراؤها على الأعداد.

ج- جملة شرطية (if statement) لتنفيذ العملية المطلوبة عن طريق استدعاء الوحدة الخاصة بها.

د- حفظ الملف باسم (main) في المُجلَّد نفسه الذي حُفِظت فيه الوحدات السابقة.

- تنفيذ البرنامج (main)، وإدخال عملية جديدة وأرقام مختلفة كل مَرَّة.

نُشارِك أفراد المجموعات الأُخرى في ما نتوصَّل إليه من نتائج، ونتبادل معًا التغذية الراجعة.

نشاط فردي

أُنفِّذ الخطوات التي تعلَّمْتُها في المثال السابق لعمل وحدات خاصة بإيجاد مساحة شكل رباعي ومحيطه واستدعاء كلٍّ منهما.

أُشـــارِك الزملاء/ الزميلات في البرنامج، ونتبادل معًا ما يتعلَّق بالبرنامج من أفكار وتحدِّيات وكيفية التغلُّب عليها.

تثبيت الحُزَم الخارجية ومكتبة (NumPy):

تُثبَّت الحُزَم الخارجية في لغة البرمجة بايثون باســـتخدام مدير الحُزَم الافتراضي (pip) واستخدام الأمر:

`(pip install package_name)` .

أمّا مكتبة (NumPy) فهي مكتبة جاهزة تُستخدَم مع المصفوفات، وتحتوي على دوال للعمل في مجال الجبر الخَطِّي، وهي ترمز إلى (Numerical Python). تُعالَج البيانات المُخزَّنة في كائن المصفوفة أسرع بنحو (50) مَرَّة تقريبًا من قوائم بايثون التقليدية؛ ما يجعلها أكثر استخدامًا في علم البيانات، ويُطلَق على كائن المصفوفة في المكتبة (Numpy) اسم (ndarray). ومن الجدير يُذكَر أنَّ جزءًا من هذه المكتبة مكتوب بلغة البرمجة بايثون، خلافًا للأجزاء التي تتطلَّب سرعة عالية؛ فهي مكتوبة بلغة C أو لغة C++.

مثال:

لتثبيت مكتبة (numpy) في جهاز الحاسوب، أَتَّبِع ما يأتي:

أفتح (command prompt) من زِرِّ البَدْء.

أكتب الأمر (pip install numpy).

```
C:\Users\user1>pip install numpy
```

قد تظهر المشكلة الآتية:

```
'pip' is not recognized as an internal or external command,
operable program or batch file.
```

يُمكِن تنزيل هذه المكتبة عن طريق الأمر الآتي:

Python –m pip install numpy

```
C:\Users\user1>python -m pip install numpy
```

ستظهر رسائل تُؤكِّد تثبيت المكتبة.

نشاط فردي

أكتب الصيغة العامة لتثبيت مكتبات بايثون، ثـــمَّ أبحث عن هذه المكتبات، وأعمل على تثبيتها باستخدام الصيغة العامة. بعد ذلك أُشارِك الزملاء/ الزميلات في النتائج التي توصَّلْتُ إليها.

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن وظائف المكتبات الآتية:

- مكتبة (pillow).
- مكتبة (emoji).
- مكتبة (PyPDF2).

أُدوِّن النتائج التي أتوصَّل إليها، ثمَّ أُشارِكها مع الزملاء/ الزميلات في الصف.

استخدام المكتبات:

يتطلَّب استخدام المكتبات في برامج بايثون استيرادها أوَّلًا. ويُمكِن ذلك باستخدام الأمر (import) متبوعًا باسم المكتبة كما هو مُوضَّح في المثال الآتي.

مثال:

يعمل البرنامج الآتي على اســـتيراد مكتبة (numpy)، وتعريف مُتغيِّر يحمل اسم (five_elements)، ويحتوي على خمسة عناصر، ثمَّ طباعته.

```
import numpy
five_elements=numpy.array([1,2,3,4,5])

print(five_elements)
```

بعد تنفيذ البرنامج، سيظهر الناتج الآتي:

```
[1 2 3 4 5]
```

تتيح برامج بايثون كتابة أسماء مستعارة (Alias) بدلًا من كتابة الأسماء الحقيقية للمكتبات. فمثلًا، يُمكِن التعديل على المثال السابق كما يأتي:

```
import numpy as np
five_elements=np.array([1,2,3,4,5])

print(five_elements)
```

إضاءة

يُمكِن تعرُّف النسخة المُستخدَمة في العمل بكتابة الجملة الآتية:

```
import numpy
print (numpy.__version__)
```

نشاط فردي

أُدخِل كل مقطع برمجي ممّا يأتي في مُفسِّر لغة البرمجة بايثون، ثمَّ أُنفِّذه، وأُلاحِظ الناتج:

أ-

```
import numpy as np
five_elements=np.array([1,2,3,4,5])

print(five_elements)
print(type(five_elements))
```

ب-

```
import numpy as np
five_elements=np.array((1,2,3,4,5))

print(five_elements)
print(type(five_elements))
```

أُشارِك الزملاء/ الزميلات في النتائج التي أتوصَّل إليها، ثمَّ أُقارِنها بما توصَّل إليه الزملاء/ الزميلات من نتائج، ونتبادل معًا التغذية الراجعة.

إنشاء الحُزَم (User-Defined Packages):

يُعَدُّ إنشاء الحُزَم باستخدام لغة البرمجة بايثون واحدًا من الجوانب المُهِمَّة في تطوير البرمجيات الحديثة؛ إذ إنَّه يتيح لمُطوِّري البرامج تنظيم المقاطع البرمجية (الأكواد) بطريقة منهجية يُمكِن بها إعادة الاستخدام. ومن ثَمَّ، فإنَّ إنشاء الحُزَم يسمح بتقسيم المشروعات الكبيرة إلى أجزاء صغيرة ومُنفصِلة؛ ما يُسهِّل إدارة الكود. كذلك تُمكِّن الحُزَم مُطوِّري البرامج من مشاركة الآخرين في برامجهم؛ إمّا بتثبيتها محليًّا، وإمّا بنشرها في مستودعات، مثل مستودع بايثون الرسمي (PyPI).

مثال:

يُبيِّن المثال الآتي كيف يُمكِن إنشاء حُزْمة وحدات:

تتولّى الوحدة الأولى اختبار إذا كان العدد موجبًا أو سالبًا، ثمَّ طباعة النتيجة، في حين تعمل الوحدة الثانية على اختبار إذا كان العدد زوجيًّا أو فرديًّا، ثمَّ طباعة النتيجة.

يُمكِن إنشاء الحُزْمة بتنفيذ الخطوات الآتية:

- إنشاء ملف بايثون فارغ، ثمَّ حفظه باسم (__init__.py) في مُجلَّد الحُزْمة، وليكن اسم المُجلَّد (Package_G12). أتذكَّر أنَّه يجب الانتباه إلى حالة الأحرف؛ لأنَّ لغة البرمجة بايثون حسّاسة تجاه حالة الأحرف.
- **إنشاء ملف بايثون جديد، ثمَّ كتابة الدالَّة الآتية فيه:**

```
def even_odd (x):
    if x%2 ==0:
        print ("The number is even number")
    else:
        print ("The number is odd number")
    return
```

- **حفظ الملف في مُجلَّد الحُزْمة باسم** (even_number.py). **أُلاحِظ أنَّ** (even_number) **يُعَدُّ وحدة نمطية** (module):

يُمكِن إنشاء الوحدة النمطية بتنفيذ الخطوتين الآتيتين:

- **إنشاء ملف بايثون آخر، ثمَّ كتابة الدالَّة الآتية فيه:**

```
def positive_or_negative(x):
    if x > 0:
        print("The number is positive")
    elif x < 0:
        print("The number is negative")
    else:
        print("The number is equal to 0")
    return
```

- حفظ الملف في مُجلَّد الحُزْمة باسم (positive_negative.py).

إنشاء ملف بايثون جديد باسم (main)، ثمَّ كتابة المقطع البرمجي الآتي فيه:

```
from G12_package import even_number
from G12_package import positive_negative

number= int (input ("Enter a number="))
even_number. even_odd (number)

positive_negative.positive_or_negative (number)
```

- حفظ الملف خارج ملف الحُزْمة، وفي مكان حفظ المُجلَّد نفســـه. فمثلًا، إذا حفظْتُ الحُزْمة على سطح المكتب، فإنَّني أحفظ هذا الملف أيضًا على سطح المكتب.
- **أُنفِّذ البرنامج في الملف (main)، فأُلاحِظ النواتج الآتية بناءً على المُدخَلات:**

.1

```
Enter a number=211
The number is odd number
The number is positive
```

.2

```
Enter a number= -45
The number is odd number
The number is negative
```

.3

```
Enter a number= -24
The number is even number
The number is negative
```

من المُلاحَظ أنَّني استخدمْتُ الأمر البرمجي الآتي لاستدعاء وحدة نمطية داخل حُزْمة مُعيَّنة:

```
from package_name import module_name
```

ثمَّ استخدمْتُ الأمر البرمجي الآتي لاستدعاء دالَّة مُعيَّنة داخل الوحدة النمطية:

```
module_name.function_name
```

يُمكِن استخدام هذه الحُزْمة في برامج ومشروعات أُخرى بعد حفظها في المُجلَّد المُخصَّص للحُزَم داخل ملف بايثون في جهاز الحاسوب.

نشاط عملي

أُنشِئ – بالتعاون مع أفراد مجموعتي – إحدى الحُزَم البرمجية الآتية ضمن مشروع خاص بإدارة المهام، يساعد الطلبة على تنظيم أوقاتهم ومهامهم الدراسية:

- حُزْمة لإنشاء المهام وتحديثها وحذفها (tasks.management).
- حُزْمة لتنبيه الطلبة بخصوص المواعيد النهائية (tasks.notifications).

- حُزْمة لتوليد تقارير عن المهام المُكتمِلة (tasks.report).

نُشارِك أفراد المجموعات الأُخرى في تنفيذ النشاط، ثمَّ نجمع الحُزَم بعضها مع بعض في ملف يحمل اسم (main).

نشاط فردي

أقرأ المقطع البرمجي الآتي بتدبُّر ورَوِيَّة، ثمَّ أُجيب عن الأسئلة التي تليه:

```
# main.py
from mymathpackage import basic_operations
print("Square root of 16=", basic_operations.sqrt(16))
print("2 to the power of 3=", basic_operations.power(2, 3))
print("Sin of 30 degrees=", basic_operations.sin(30))
print("Cos of 60 degrees=", basic_operations.cos(60))
print("Tan of 45 degrees=", basic_operations.tan(45))
```

- أُناقِش ما يقوم به المقطع البرمجي بوجه عام.
- أُحدِّد اسم الحُزْمة البرمجية المُستخدَمة في المقطع البرمجي.
- أُنشِئ الحُزْمة البرمجية المطلوبة، وأحرص على تضمين جميع الدوال البرمجية المُشار إليها في المقطع البرمجي.
- أحفظ الحُزْمة البرمجية في مكانها الصحيح؛ لأتمكَّن من استدعائها عن طريق البرنامج الرئيس.
- أُنفِّذ البرنامج باستخدام مُفسِّر لغة البرمجة بايثون، ثمَّ أُلاحِظ النتائج.
- أُدوِّن النتائج التي أتوصَّل إليها، ثمَّ أُشارِكها مع الزملاء/ الزميلات في الصف.

المواطنة الرقمية:

- **نشر المعرفة**: أُقسِّم البرامج الكبيرة إلى حُزَم، ثمَّ إلى وحدات برمجية ودوال برمجية؛ لكي يتمكَّن مُطوِّرو البرامج من استخدامها.
- **الأمان والموثوقية**: أُراعي حقوق الملكية الفكرية؛ بأنْ أستخدم الحُزَم بصورة قانونية، وألّا أتعدّى على حقوق الملكية الفكرية لأصحاب الحُزَم. كذلك أتجنَّب استخدام الحُزَم التي تُوزَّع بطريقة غير قانونية أو دون إذن من أصحاب الحقوق.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُعرِّف الحُزَم البرمجية، ثمَّ أُوضِّح أهمية استخدامها في المشروعات.

السؤال الثاني: ما الفَرْق بين الوحدة (Module) والحُزْمة (Package)؟

السؤال الثالث: ما الخطوات اللازمة لإنشاء وحدة برمجية (Module) في بايثون؟

السؤال الرابع: كيف يُمكِن تثبيت مكتبة خارجية باستخدام (pip)؟

السؤال الخامس: أملأ الفراغ بما هو مناسب في ما يأتي:

1- الهدف الرئيس لإنشاء الحُزَم في بايثون هو ..... .

2- الملف الذي يجب أنْ تحتويه كل حُزْمة في بايثون هو..... .

3- يُمكِن استيراد ملف مُعيَّن من حُزْمة في بايثون باستخدام التعليمة ..... .

4- الخطوة الأولى لإنشاء حُزْمة في بايثون هي ..... .

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل: ماذا تعني عبارة: "الاتِّساق في ممارسات الترميز الخاصة بالمُبرمِج"؟

السؤال الثاني: أُنشِئ حُزْمة تساعد على معالجة النصوص؛ بأنْ تعمل على حساب عدد الكلمات، وعكس النصوص، وتحويل النصوص من أحرف كبيرة إلى أحرف صغيرة والعكس.

السؤال الثالث: أُنشِئ وحدة برمجية لحساب محيط دائرة ومساحتها، ثمَّ أحفظها في ملف مُنفصِل.

أسئلة الوحدة

السؤال الأوَّل: أختار رمز الإجابة الصحيحة في كلٍّ ممّا يأتي:

1. **من مراحل دورة حياة تطوير النظام التي تتضمَّن وضع المُخطَّط الزمني للتنفيذ:**
   - التطوير.
   - التصميم.
   - التخطيط وتحليل المُتطلَّبات.
2. **الوثيقة التي تنتج من مرحلة تحديد المُتطلَّبات هي:**
   - (Software Requirement Specification).
   - (Design Document Specification).
   - (Development Document Specification).
3. **من مراحل دورة حياة تطوير النظام التي تتضمَّن إنتاج صيغة برمجية قابلة للاستخدام:**
   - التصميم.
   - التطوير.
   - الاختبار.
4. **أحد أنواع البيانات المُجرَّدة يُستخدَم فيه الأمر (pop) لحذف عنصر من عناصره:**
   - (List).
   - (Stack).
   - (Set).
5. **نوع من البيانات يدعم العمليات الرياضية، مثل عمليات التقاطع والاتحاد:**
   - القاموس.
   - الصف.
   - المجموعة.
6. **الرمز (numpy) هو اختصار لـ:**
   - (Numerical Platform).
   - (Numerical Python).
   - (Number Picker).

السؤال الثاني: أُعرِّف المقصود بدورة حياة تطوير الأنظمة (SDLC)، وأذكر مراحلها الأساسية.

السؤال الثالث: أذكر المزايا والعيوب لكلٍّ من نموذج الشلّال (Waterfall)، والنموذج الرشيق (Agile)

السؤال الرابع: أقرأ الحالة الآتية، ثمَّ أختار نموذج (SDLC) المناسب، وأُبرِّر إجابتي:
تعمل شركة طيران على تطوير نظام حجز تذاكر يَتَّسِم بالتعقيد.

السؤال الخامس: أرسم مُخطَّط سَيْر العمليات للمسألة الآتية:
حساب الراتب الإجمالي لمندوب مبيعات، راتبه الأساسي (300) دينار أردني، والعمولة الإضافية التي يأخذها مُوزَّعة على النحو الآتي:

1. إذا كانت مبيعاته أقلَّ من (1000) دينار أو تساويها، فإنَّ نسبة عمولته تبلغ 6٪.
2. إذا كانت مبيعاته أكثر من (1000) دينار، وأقلَّ من (3000) دينار أو تساويها، فإنَّ نسبة عمولته تبلغ 7,5٪.
3. إذا كانت مبيعاته أكثر من (3000) دينار، وأقلَّ من (5000) دينار أو تساويها، فإنَّ نسبة عمولته تبلغ 9٪.
4. إذا كانت مبيعاته أكثر من (5000) دينار، فإنَّ نسبة عمولته تبلغ 10٪.

السؤال السادس: ما ناتج تنفيذ كل مقطع برمجي ممّا يأتي:

أ-

```
def draw_triangle(n):
    if n == 0:
        return
    draw_triangle(n - 1)
    print("*" * n)

draw_triangle(5)
```

ب-

```
def repeat_string(s, n):
    if n == 0:
        return ""
    return s + repeat_string(s, n - 1)

print(repeat_string("Hello", 3))
```

تقويم ذاتي (Self-Checklist)

بعد دراستي هذه الوحدة، اقرأ الفقرات الواردة في الجدول الآتي، ثمَّ أضع إشارة (✔) في العمود المناسب:

مُؤشِّرات الأداء	نعم	لا	لستُ مُتأكِّدًا
1- أُعرِّف دورة حياة تطوير النظام (SDLC).	○	○	○
2- أَصِف دورة حياة تطوير النظام باستخدام أحد النماذج الآتية؛ (Spiral, Agile, or Waterfall).	○	○	○
3- أستخدم المُخطَّطات (Flow Charts) والخوارزمية شبه الرمزية (Pseudocode) في تمثيل الخوارزميات المُركَّبة.	○	○	○
4- أستخدم أدوات تمثيل المُخطَّطات.	○	○	○
5- أُقارِن بين هياكل تراكيب البيانات المناسبة أثناء حَلِّ مشكلة مُحدَّدة.	○	○	○
6- أُوضِّح كيفية عمل البرامج الراجعة.	○	○	○
7- أَجِد حلولًا برمجيةً باستخدام جمل برمجية مُعَدَّة مُسبَّقًا (روتين فرعي).	○	○	○
8- أستخدم لغة البرمجة بايثون في إيجاد حلول برمجية.	○	○	○
9- أستخدم لغة البرمجة بايثون في إعداد مكتبة برمجية (Library).	○	○	○
10- أُعرِّف الأنماط البرمجية	○	○	○
11- أُجزِّئ المشكلة إلى مشكلات صغيرة يُمكِن حَلُّها عن طريق البرمجيات أو أجزاء من برامج مُعَدَّة مُسبَّقًا.	○	○	○

تعليمات للمراجعة والتحسين:

إذا اخترْتُ (لا) أو (لسْتُ مُتأكِّدًا) لأيٍّ من الفقرات السابقة، فأَتَّبع الخطوات الآتية لتجنُّب ذلك:

- ُراجِع المادة الدراسية؛ بأنْ أُعيد قراءة المحتوى المُتعلِّق بالمعيار.
- أطلب المساعدة؛ بأنْ أُناقِش مُعلِّمي/ مُعلِّمتي أو زملائي/ زميلاتي في ما تعذَّر عليَّ فهمه.
- أستخدم مراجع إضافية؛ بأنْ أبحث عن مراجع أُخرى مثل الكتب، أو أستعين بالمواقع الإلكترونية الموثوقة التي تُقدِّم شرحًا وافيًا للموضوعات التي أَجِد صعوبة في فهمها.

تأمُّلات ذاتية

عزيزي الطالب، عزيزتي الطالبة:

التأمُّلات الذاتية هي فرصة لتقييم عملية التعلُّم، وفهم التحدِّيات، وتطوير اســـتراتيجيات لتحسين عملية التعلُّم مستقبلًا.

أملأ الفراغ في ما يأتي بالأفكار والتأمُّلات الشخصية التي يُمكِن بها تحقيق أفضل استفادة من التجربة التعليمية:

تعلَّمْتُ في هذه الوحدة:

يُمكِنني أنْ أُطبِّق ما تعلَّمْتُه في:

الصعوبات التي واجهْتُها أثناء عملية التعلُّم:

ذلَّلْتُ هذه الصعوبات عن طريق:

يُمكِنني مستقبلًا تحسين:

(self, **kwargs):
.name = kwargs.
f.damage = kwargs.
lf.armor = kwargs.
elf.hit_points = kwargs.
self.current_hit_points
self.level = kwargs.
attack(self, enemy:
ttack enemy unit.
amage_top_lim
return

الوحدة
2

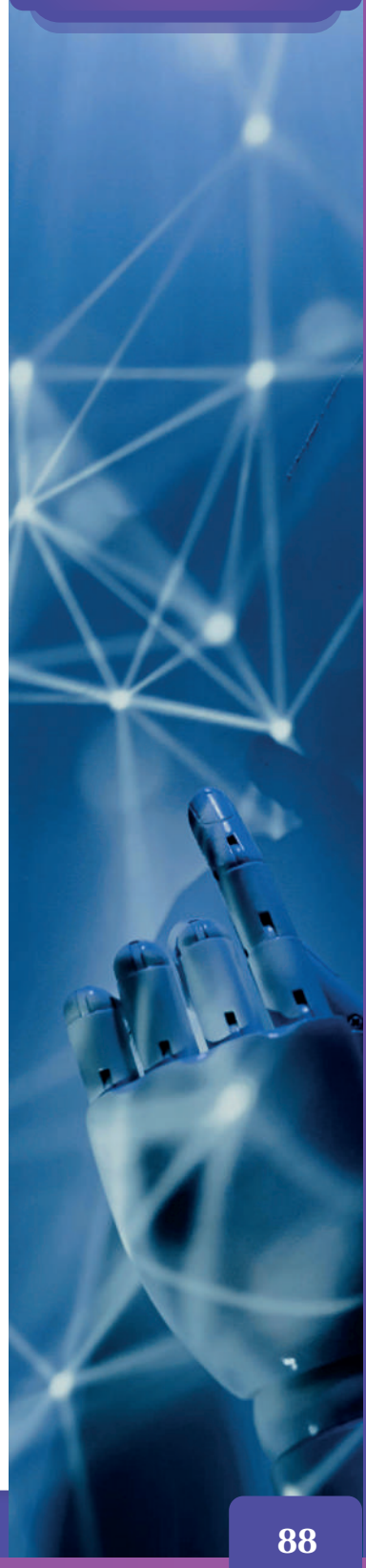
تحليل البيانات
(Data Analysis)

نظرة عامة على الوحدة:

سأتعلَّم في هذه الوحدة كيف يُمكِن تمثيل البيانات وفهم طرائق تحليلها، وأتعرَّف أهمية تحليل البيانات في استخلاص النتائج واتِّخاذ القرارات. كذلك سأُطوِّر مهارات جمع البيانات وتحليلها لتحديد الأنماط والعلاقات المُعقَّدة، وأستخدم الرسوم البيانية أداةً فاعلةً لتوضيح المعلومات والبيانات، وأتعرَّف علاقة السبب والنتيجة في تفسير البيانات، إضافةً إلى دراسة مفهوم الفرضية، وتعلُّم كيفية بنائها واختبارها أثناء عملية التحليل.

يُتوقَّع منّي في نهاية الوحدة أنْ أكون قادرًا على:

- توضيح طرائق تحليل البيانات.
- بيان أهمية تحليل البيانات.
- تحديد أدوات جمع البيانات وتقنيات تحليلها.
- استخدام الأدوات والتقنيات الخاصة بتحليل البيانات في استخلاص النتائج الإحصائية واستكشاف الأنماط المُتكرِّرة فيها.
- توضيح طرائق جمع البيانات باستخدام أدوات وتقنيات معروفة.
- المفاضلة بين الأدوات والتقنيات المتوافرة اعتمادًا على طبيعة المشكلة التي يُراد إيجاد حَلٍّ لها.
- **جمع بيانات باستخدام أدوات برمجية (مثل**: برامج المحاكاة، ولغة البرمجة بايثون) في مجالات مُحدَّدة.
- تقديم الاستنتاجات واستكشاف البيانات عن طريق تحليل الأنماط باستخدام الرسوم البيانية.
- استنتاج علاقة السبب والنتيجة من البيانات وتفسيرها.
- التنبُّؤ بالنتائج عن طريق الأسباب أو العكس.
- استخدام البيانات في إيصال فكرة مُحدَّدة بوضوح ودِقَّة.
- وصف طرائق فحص الفرضيات باستخدام نماذج المحاكاة ضمن سياقات واقعية.



المهارات الرقمية: التفكير الحاسوبي، التفكير الإبداعي التحليلي، التعاون الرقمي، المواطنة الرقمية، المسؤولية الرقمية.

فهرس الوحدة:

- **الدرس الأوَّل:** مُقدِّمة في تحليل البيانات (Introduction to Data Analysis).
- **الدرس الثاني:** تقنيات تحليل البيانات (Data Analysis Techniques).
- **الدرس الثالث:** جمع البيانات وتحليلها باستخدام برمجية بايثون (Python Data Collection & Analyses Methods).
- **الدرس الرابع:** علاقة السبب والنتيجة في البيانات (Data Cause–and– Effect Relation).
- **الدرس الخامس:** الفرضيات في تحليل البيانات (Hypotheses in Data Analysis).

الدرس الأوَّل

مُقدِّمة في تحليل البيانات
(Introduction to Data Analysis)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس مفهوم البيانات، وأنواعها المختلفة، وأهمية تحليلها في فهم العلاقات واستخلاص النتائج. كذلك سأتعرَّف أهمَّ طرائق تحليل البيانات التي تُستخدَم في تفسير الأنماط واتِّخاذ القرارات بصورة فاعلة.

المفاهيم والمصطلحات:

البيانات (Data)، أنواع البيانات (Data Types)، تحليل البيانات (Data Analysis)، الأنماط (Pattern)، الاتِّجاه (Trend)، تحليل النص (Text Analysis)، التحليل الوصفي (Descriptive Analysis)، التحليل التشخيصي (Diagnostic Analysis)، التحليل التنبُّؤي (Predictive Analysis)، التحليل التوجيهي (Prescriptive Analysis).

نتاجات التعلُّم (Learning Outcomes):

- أُعرِّف البيانات، وأذكر أنواعها.
- أُبيِّن أهمية تحليل البيانات.
- أُوضِّح أهمَّ طرائق تحليل البيانات.

نعيش اليوم في عصر استحوذت عليه البيانات؛ إذ أصبحت عنصرًا أساسيًّا تعتمد عليه المؤسسات والأفراد في اتِّخاذ القرارات المُهِمَّة. ففي هذا العصر، أصبح كَمُّ البيانات

هائلًا أكثر من أيِّ وقت مضى، وغَدَت البيانات رأسَ مالِ الشركات الكبرى ومفتاحًا للولوج إلى عالَم النجاح والتطوُّر. ومن ثَمَّ أخذت البيانات تعيد تشكيل فهمنا للعالَم وكيفية التعامل معه، وقد تزايدت أهميتها في ظلِّ الانتشار الواسع لوسائل التكنولوجيا الحديثة، التي أتاحت للناس كافَّةً جمع كَمٍّ ضخم من البيانات واستخدامها.

نشاط تمهيدي

بناءً على معرفتي السابقة، أُصنِّف كُلًّا ممّا يأتي إلى بيانات ومعلومات:

- علامات الطلبة في مبحث العلوم.
- قائمة تحوي أسماء الطلبة مُرتَّبةً ترتيبًا هجائيًّا.
- نسب هطل الأمطار خلال أسبوع كامل.
- نتائج الطلبة النهائية في امتحان قَبول (ناجح أو راسب).

أُدوِّن إجاباتي، ثمَّ أُشارِكها مع الزملاء/ الزميلات في الصف.

البيانات: مفهومها، وأنواعها (Data : Concept and Types)

تُعرَّف البيانات بأنَّها أرقام، أو حروف، أو صور، أو تسجيل صوتي، أو تسجيل مرئي يُمكن الاستفادة منها بعد تحليلها ومعالجتها في اتِّخاذ قرارات مُهِمَّة بخصوص إجراء مُعيَّن. ويعتقد كثير من الناس أنَّ البيانات عامةً ليس لها معنى. غير أنَّ البيانات تكتسب معنى عند تفسيرها وتحويلها إلى معلومات. وبالمثل، فإنَّ فحص البيانات ومعالجتها يُمكِّننا من إيجاد أنماط مُحدَّدة لهذه البيانات، تساعدنا على اتِّخاذ قرارات مُهِمَّة. كذلك توحي البيانات بصور مفيدة ودلالات مُتعدِّدة عند تحليلها. وفي جميع الأحوال، فإنَّ الهدف من تحليل البيانات هو تحويل البيانات الخام إلى معارف ومعلومات تساعد الفرد على اتِّخاذ القرار.

إضاءة

كلمة (بيانات) في اللغة العربية هي ترجمة لكلمة (Data) في اللغة الإنجليزية، التي يعود أصلها إلى الكلمة اللاتينية (Datum)، والتي تعني الشيء المعطى، أو المعلومة المفردة، وهي مُشتَقَّة من الفعل اللاتيني (Dare) الذي يعني يعطي أو يُسلِّم.

أنواع البيانات (Data Types):

تأخذ البيانات أشكالًا عديدةً وأنواعًا مختلفةً؛ لذا يُعَدُّ فهم خصائص كل نوع من أنواع البيانات أمرًا مُهِمًّا يساعد على تعرُّف نوع العمليات التي يُمكِن إجراؤها على هذا النوع من البيانات.

على سبيل المثال، يُمكِن التعبير عن درجة حرارة الجو بطريقتين أو شكلين، هما:

- الأرقام التي تُعبِّر عن درجات الحرارة بالوحدة المئوية، مثل 31 درجة مئوية، والأرقام التي تُعبِّر عن درجات الحرارة بوحدة الفهرنهايت، مثل 100 درجة فهرنهايت على مقياس كلفن. ويُنظَر إلى هذه الأرقام بوصفها بيانات رقمية؛ لأنَّه يُمكِن إجراء العمليات الحسابية عليها، مثل؛ الجمع، والطرح، والمقارنة.

- المستويات، مثل: حار، ومعتدل، وبارد. وهذا النوع من البيانات يُعرَف بالتصنيفات الترتيبية؛ لِما تُمثِّله البيانات من ترتيب مُعيَّن بعيدًا عن الأرقام.

ومن ثَمَّ، فإنَّ الأرقام والمستويات تشير إلى درجة الحرارة في منطقة مُعيَّنة، وتُمثِّل أنواعًا مختلفةً من البيانات. وفي بعض الأحيان، يُمكِن تحويل مجموعة من أنواع البيانات إلى أنواع أُخرى بحسب الحاجة أو سياق التحليل.

تعلَّمْتُ في صفوف سابقة أنَّه توجد أنواع مختلفة من البيانات يُمكِن تخزينها داخل المُتغيِّرات. وفي ما يأتي ذكر لأبرز أنواع البيانات المتوافرة في قواعد البيانات والبرمجيات:

1. **البيانات العددية (Numeric):** تُمثِّل الأعداد هذا النوع من البيانات، وهي تُستخدَم في إجراء العمليات الحسابية (مثل: الجمع، والطرح)، وتشمل ما يأتي:
   - **الأعداد الصحيحة** (Integer)؛ **وهي الأعداد الصحيحة من دون كسور، مثل:** 10، و0، و5-
   - **الأعداد العشرية** (Float)؛ **وهي الأعداد ذات الفواصل العشرية، مثل:** 3.14، و2.8-، و0.05
2. **البيانات النصية (Text):** تُصنَّف هذه البيانات إلى نوعين، هما:
   - **الحرف** (Character: CHAR) الذي يُمثِّل وحدة بيانات فردية قد تكون حرفًا أبجديًّا (مثل: A، وأ)، أو رقمًا (مثل: 1، و9)، أو علامةَ ترقيم (مثل: . و،)، أو رمزًا خاصًّا (مثل: *، و@)، أو مسافةً فارغةً؛ أيْ فراغًا بين النصوص.

- **السلاسل النصية** (Strings) **التي تُستخدَم في تخزين سلسلة من النصوص** (مثل: الأسماء، والجمل، والرموز)، ويُمكِن أنْ تحتوي على الحروف (مثل: A–Z ، a–z، أ - ي)، والأرقام (بالرغم من أنَّها لا تُعامَل بوصفها قِيَمًا رقميةً في هذا السياق)، والرموز الخاصة، مثل: @، و#، و$، و!.

3. **البيانات المنطقية (Boolean):** تُستخدَم هذه البيانات في اتِّخاذ القرارات، أو تحديد الحالات المنطقية. وهي تُمثِّل حالتين فقط، هما: صواب (True)، وخطأ (False).
4. **قوائم البيانات (Lists/Arrays):** تُستخدَم هذه القوائم في تخزين مجموعة من القِيَم المُرتَّبة، مثل: [1, 2, 3]، و ["أحمر"، "أصفر"، "أخضر"].
5. **البيانات الترتيبية أو البيانات التصنيفية (Categorical):** تُمثِّل هذه البيانات فئات أو مجموعات، مثل: ("ذكر"، "أنثى")، و ("حار"، "بارد").
6. **البيانات المُعقَّدة (Complex):** تُمثِّل هذه البيانات الهياكل (Structures)، والكائنات (Objects) التي تحتوي على أنواع مُتعدِّدة من البيانات في آنٍ معًا.
7. **المجموعات (Sets):** هي مجموعات غير مُرتَّبة من العناصر الفريدة (أيْ لا تتكرَّر)، مثل: {1, 2.3}، و{"تفاحة"، "موز"، "برتقال"}.
8. **الصفوف (Tuples):** هي مجموعات مُرتَّبة من العناصر التي لا يُمكِن تعديلها بعد تعريفها، مثل: ("1"، أحمر، True)، و("تفاحة"، 3 ,5، False ").
9. **القواميس (Dictionaries):** توصَف القواميس بأنَّها هيكل بيانات يُخزِّن القِيَم بوصفها مفاتيحَ وقِيَمًا مُرتبِطةً بها، مثل: "اسم": "محمد" ، "عمر": "25"، و"البلد": "الأردن"، "العاصمة": "عمّان".
10. **الوقت والتاريخ:** يُستخدَم الوقت والتاريخ في تخزين بيانات عن كلٍّ من التاريخ (سنة، شهر، يوم) والوقت (ساعة، دقيقة، ثانية) في حقل واحد. وهذا النوع من البيانات يفيد في تسجيل الأحداث الزمنية.

نشاط جماعي

أُحدِّد – بالتعاون مع أفراد مجموعتي – نوع البيانات الواردة في كل صف من صفوف الجدول الآتي:

البيانات (Data)	نوعها (Data Type)
"Hello World"	
20	
20 , 5	
[20,32,45,90]	
("apple", 'Banana", "Cherry")	
{36 : "name" : "John" , " age"}	
False	
– 56	
hello@gmail.com	
"5763_5432_1235"	

تحليل البيانات (Data Analysis):

يُعرَّف تحليل البيانات بأنَّه عملية منهجية لجمع البيانات، وتنظيمها، وتنظيفها، وتفسيرها، وتمثيلها؛ بُغْيَةَ استخراج رؤى ذات معنى، وتحديد الأنماط، ودعم اتِّخاذ القرارات المستنيرة.

تعتمد الأساليب المُستخدَمة في تحليل البيانات على طبيعة البيانات وأهداف التحليل. وفي هذا السياق، تُستخدَم تقنيات إحصائية وحاسوبية ومنطقية للكشف عن الاتِّجاهات، واختبار الفرضيات، وحَلِّ المشكلات المُعقَّدة.

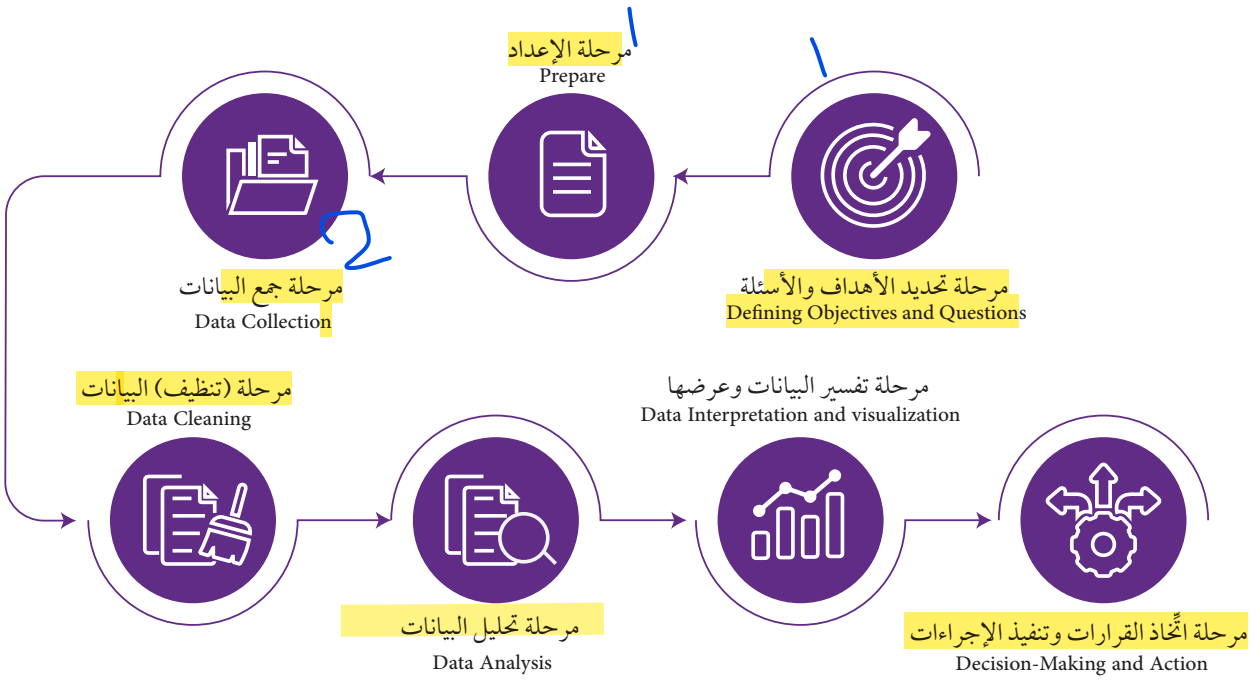
يستفاد من تحليل البيانات على المستوى الشخصي والمستوى المُؤسَّسي، ويشمل ذلك جميع المجالات. فعلى المستوى الشخصي، يعتمد تحليل البيانات على بيانات بسيطة وتحليل مباشر بناءً على الخبرة أو المعرفة السابقة. فمثلًا، إذا لاحظْتُ أثناء التسوُّق أنَّ سعر الخبز الذي يحمل علامة تجارية مُفضَّلة لديَّ (هي بيانات معروفة بالنسبة إليَّ) قد ارتفع، فإنَّني أُحلِّل هذه البيانات لاتِّخاذ قرار بخصوص شراء هذا النوع من الخبز، أو البحث عن بديل له.

أمّا على المستوى المُؤسَّسي، فقد يكون تحليل البيانات أكثر شمولًا وتعقيدًا، ويتطلَّب التعامل مع كَمٍّ هائل من البيانات (Big Data) باستخدام أدوات تحليل مُتقدِّمة ومهارات مُتخصِّصة. وهذا التحليل يساعد المؤسسات على فهم الأنماط، وتوقُّع الاتِّجاهات، والتوجُّه نحو اتِّخاذ قرارات استراتيجية تُؤثِّر في الأعمال تأثيرًا كبيرًا.

مراحل عملية تحليل البيانات:

بوجه عام، تَمُرُّ عملية تحليل البيانات بالمراحل التكرارية السبع الآتية: تحديد الأهداف والأسئلة، الإعداد، جمع البيانات، تنظيف البيانات، تحليل البيانات، تفسير البيانات وعرضها، اتِّخاذ القرارات وتنفيذ الإجراءات. أنظر الشكل (1-1).

يُذكَر أنَّ كل مرحلة من تلك المراحل تعتمد على المرحلة التي تسبقها؛ لضمان أنْ يكون التحليل فاعلًا وواضحًا وقابلًا للتنفيذ.



الشكل (1-1) : مراحل عملية تحليل البيانات.

في ما يأتي توضيح لهذه المراحل:

1. **مرحلة تحديد الأهداف والأسئلة (Defining Objectives and Questions):** تُركِّز هذه المرحلة على فهم المشكلة، وتحديد الأهداف وتوقُّعات الأطراف المَعْنِيَّة.

   من الأسئلة التي تُطرَح في هذه المرحلة:

   - ما المشكلة التي أُحاوِل حَلَّها؟
   - ما الذي يَلزم الأطراف المَعْنِيَّة؟

2. **مرحلة الإعداد (Prepare)**: بعد تحديد المشكلة، يجب تحديد البيانات المطلوبة وكيفية تنظيمها.

من الأسئلة التي تُطرَح في هذه المرحلة:

- ما البيانات التي أحتاج إليها؟
- كيف أُنظِّم هذه البيانات تنظيمًا فاعلًا؟
- ما المصادر التي سأجمع منها البيانات التي تَلزمني؟

3. **مرحلة جمع البيانات (Data Collection)**: يتمُّ في هذه المرحلة جمع البيانات ذات الصلة بالإجابة عن الأسئلة السابقة، وذلك باستخدام طرائق مُتنوِّعة تتناسب مع طبيعة المشكلة وأهداف التحليل (مثل: الاستبانات، والمقابلات، والملاحظات)؛ لمراقبة السلوكات والعمليات وقت حدوثها، أو استخراج البيانات من قواعد البيانات الداخلية أو من المصادر الخارجية. وقد تكون البيانات المُجمَّعة كَمِّية، مثل: الأرقام، والنتائج الإحصائية. وقد تكون نوعية، مثل: النصوص، والصور.

4. **مرحلة تنظيف البيانات (Data Cleaning)**: تُعَدُّ هذه المرحلة أساسية في عملية تحليل البيانات؛ إذ تتضمَّن فحص البيانات بصورة دقيقة لتحديد الأخطاء والتناقضات ومعالجتها. وقد يشمل ذلك إزالة الإدخالات المُكرَّرة، والتعامل مع القِيَم المفقودة، وتصحيح الأخطاء الناجمة عن إدخال البيانات، وضمان اتِّساق البيانات. ومن ثَمَّ يهدف تنظيف البيانات إلى تحسين جودة البيانات وزيادة درجة موثوقيتها، وهو أمر ضروري للحصول على نتائج دقيقة ذات دلالات مُحدَّدة.

5. **مرحلة تحليل البيانات (Data Analysis)**: يبدأ في هذه المرحلة تنفيذ العمل التحليلي الأساسي باستخدام تقنيات وأدوات خاصة بتحليل البيانات المختلفة (مثل: لغة البرمجة بايثون (Python)، ولغة البرمجة (R)، وبرمجية إكسل (Excel)، والبرامج المُتخصِّصة، مثل: (SPSS)، و (SAS))؛ بُغْيَةَ استكشاف البيانات، واكتشاف الأنماط، أو العلاقات، أو الاتِّجاهات، أو الارتباطات بين البيانات، التي تعالج الأهداف والأسئلة التي حُدِّدت مُقدَّمًا.

6. **مرحلة تفسير البيانات وعرضها (Data Interpretation and Visualization)**: بعد الانتهاء من مرحلة تحليل البيانات، تأتي مرحلة تفسير نتائج البيانات وعرضها بصورة واضحة وسهلة الفهم عن طريق الرسوم البيانية والمُخطَّطات والتقارير ولوحات التحكُّم. يُذكَر أنَّ تفسير

البيانات وتمثيلها مرئيًّا يُسهِم في جعل النتائج قابلة للفهم، ويساعد الأطراف المَعْنِيَّة على فهمها، ويُمكِّنهم من اتِّخاذ قرارات مستنيرة.

7. **مرحلة اتِّخاذ القرارات وتنفيذ الإجراءات (Decision-Making and Action):** تُركِّز هذه المرحلة على استثمار النتائج المُستخلَصة من التحليل في وضع استراتيجيات فاعلة، وصياغة سياسات مناسبة، وإجراء بحوث إضافية عند اللزوم، فضلًا عن تقديم التوصيات اللازمة، وتنفيذ الحلول المناسبة.

نشاط جماعي

أُناقِش - بالتعاون مع أفراد مجموعتي- العبارات الآتية، ثمَّ نُصنِّفها معًا إلى مراحل البيانات التي تُعبِّر عنها:

- "استبعاد الصفوف التي تحتوي على أخطاء في القِيَم أو في البيانات غير المنطقية".
- "إجراء استطلاع رأي لجمع وجهات نظر العملاء حيال المُنتَج الجديد".
- "تحديد القرارات الرئيسة بناءً على النتائج المُستخلَصة من التحليل".
- "استخدام جدول محوري لاكتشاف الأنماط في البيانات".
- "تحويل التواريخ إلى صيغة مُوحَّدة لسهولة التحليل".
- "تقديم عرض تقديمي للإدارة العليا يتضمَّن الرسوم البيانية والنتائج".
- "تنفيذ الاستراتيجية الجديدة الآن بناءً على تحليل البيانات".

أُشارِك أفراد المجموعات الأُخرى في الإجابة التي توصَّلنا إليها في المجموعة، ثمَّ نتبادل معًا التغذية الراجعة.

أهمية تحليل البيانات:

يُعَدُّ تحليل البيانات أداة أساسية للكشف عن المعلومات المُهِمَّة، ودعم عملية اتِّخاذ القرارات المُستنِدة إلى الأدلَّة. تتمثَّل أهمية تحليل البيانات في أربعة مناحٍ رئيسة، هي: القدرة على التوصُّل إلى التنبُّؤات، وكشف الارتباطات بين المُتغيِّرات، وتعرُّف الأنماط المُتكرِّرة، ورصد الاتِّجاهات.

في ما يأتي بيان لتلك المناحي، إضافةً إلى تقديم أمثلة عملية على كلٍّ منها؛ لتوضيح دورها في تعزيز الفهم، واتِّخاذ القرارات المستنيرة:

1. **التوصُّل إلى التنبُّؤات (Predictions):**

يساعد تحليل البيانات على استقراء المستقبل بناءً على البيانات المؤرشفة. كذلك يُمكِن استخدام تطبيق تقنيات التحليل الإحصائي (مثل: الانحدار الخَطِّي، والنمذجة التنبُّؤية، والتعلُّم الآلي) في تقدير القِيَم المستقبلية بناءً على معطيات سابقة؛ ما يُمكِّننا من التخطيط المُسبَّق على نحوٍ أكثر دِقَّة، ويساعدنا على اتِّخاذ قرارات استراتيجية تدعم النمو المستدام، وتُقلِّل من المخاطر. على سبيل المثال، يُمكِن توقُّع النمو في الإيرادات أو تحديد التغيُّرات في الطلب على المُنتَجات والخدمات.

مثال:

يشير الجدول (1-1) إلى البيانات التي تتعلَّق بمُتوسِّط الرسوم الدراسية (بالدينار الأردني) للجامعات الخاصة التي تستغرق فيها مُدَّة الدراسة (4) سنوات. ومن المُلاحَظ على هذا الجدول تزايد الرسوم الدراسية كل سنة (2011- 2016م). ولكي نتمكَّن من التنبُّؤ بما سيطرأ على الرسوم الدراسية مستقبلًا؛ يتعيَّن علينا أنْ نفهم المُعدَّل الذي تتزايد فيه الأعداد المُمثِّلة للرسوم الدراسية، ويُمكِننا أنْ نستخدم لذلك الطريقة التي تقوم على حساب نسبة التغيُّر في الرسوم كل سنة.

الجدول (1-1): بيانات تُوضِّح مُتوسِّط الرسوم الدراسية للجامعات الخاصة.

السنة الدراسية	الرسوم (بالدينار الأردني)
(2020-2021)	15210
(2021-2022)	15970
(2022-2023)	16570
(2023-2024)	17140
(2024-2025)	18180

نشاط فردي

أدرس البيانات الواردة في الجدول (1-1)، ثمَّ أُحاوِل التنبُّؤ بقيمة الرسوم للسنتين الدراسيتين (2026-2027، 2025-2026)، ثمَّ أُجيب عن السؤالين الآتيين:

- ما القِيَم التي تنبَّأْتُ بها؟
- ما الطريقة التي استخدمْتُها في عملية التنبُّؤ؟

أُقارِن إجابتي بإجابات الزملاء/ الزميلات في الصف، ثمَّ نُناقِش معًا سبب الاختلاف في القِيَم.

2. كشف الارتباطات بين المُتغيِّرات (Correlations):

يساعد تحليل البيانات على فهم العلاقات الإحصائية بين المُتغيِّرات المختلفة وتحديدها؛ ما يُمكِّننا من اتِّخاذ قرارات دقيقة، علمًا بأنَّ العلاقة بين المُتغيِّرات تظهر غالبًا بأحد الأشكال الثلاثة الآتية:

- **الارتباط الموجب:** وفيه تزداد قيمة أحد المُتغيِّرات عند زيادة الآخر.
- **الارتباط السالب:** وفيه تنخفض قيمة أحد المُتغيِّرات عند زيادة الآخر.
- **عدم وجود ارتباط:** وفيه تنعدم العلاقة الواضحة بين المُتغيِّرات.

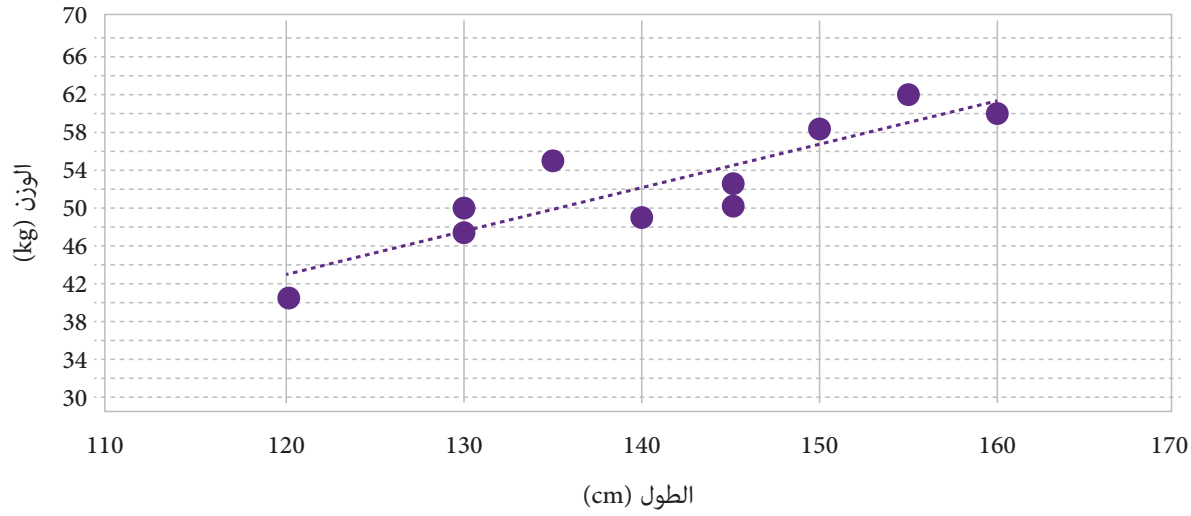
لتمثيل هذه العلاقات بصريًّا، وتحديد اتِّجاه العلاقة بين المُتغيِّرات، والكشف عن الأنماط أو القِيَم غير المنتمية، وتوفير فهم مرئي يساعدنا على تفسير البيانات بشكل أفضل؛ يُمكِننا استخدام مُخطَّط التشتُّت (Scatter Plot).

إثراء

مُخطَّط التشتُّت (Scatter Plot): هو تمثيل مرئي يُظهِر النقاط التي تُمثِّل القِيَم لمُتغيِّرين على المحور السيني (X–axis) والمحور الصادي (Y–axis).

مثال:

يُبيِّن الشكل (1–2) العلاقة بين الطول والوزن لبعض الأطفال من الفئة العمرية (10–15) عامًا. وفيه يُظهِر مُخطَّط التشتُّت وجود ارتباط إيجابي بين الطول والوزن؛ فكلَّما زاد الطول زاد الوزن في العيِّنة موضوع الدراسة.



الشكل (1–2): مُخطَّط التشتُّت للعلاقة بين الطول والوزن لمجموعة أطفال من الفئة العمرية (10–15) عامًا.

أبحث

أبحث في الحياة اليومية عن أمثلة على بيانات مُترابِطة (مثل: العلاقة بين عمر السيارة وعدد الكيلومترات التي تقطعها بالوقود، والعلاقة بين عدد ساعات استخدام الهاتف المحمول والنوم الصِّحِّي، والعلاقة بين درجة الحرارة ومبيعات المُثلَّجات)، ثمَّ أُمثِّلها باستخدام مُخطَّط تشتُّت، ثمَّ أُحلِّل المُخطَّط لتحديد نوع الارتباط بين المُتغيِّرات (موجب، سالب، عدم وجود ارتباط). بعد ذلك أعرض النتائج التي توصَّلْتُ إليها أمام مُعلِّمي/ مُعلِّمتي وزملائي/ زميلاتي لمناقشتها معًا.

3. تعرُّف الأنماط المُتكرِّرة (Patterns):

تشير الأنماط في البيانات إلى العناصر أو السلوكات التي تظهر بشكل مُتكرِّر داخل مجموعة البيانات، وتُعرَف بالنمط المُتكرِّر. يساعد اكتشاف هذه الأنماط على فهم السلوكات التي يُمكِن الإفادة منها في تحسين الأداء أو اتِّخاذ القرارات، علمًا بأنَّ هذه الأنماط ليست مُقتصِرة على عمليات البيع والشراء (مثل العلاقة بين الإعلانات وزيادة حجم المبيعات)، وإنَّما يُمكِن العثور عليها في مختلف المجالات، مثل: سلوك المُستخدِمين في تصفُّح المواقع الإلكترونية في شبكة الإنترنت، والعمليات الصناعية، وكذا الأنماط الصِّحِّية في بيانات المستشفيات؛ فهي تُعَدُّ مَطْلَبًا أساسيًّا في علم البيانات.

4. رصد الاتِّجاهات (Trends):

يشير الاتِّجاه (Trend) – في مجال تحليل البيانات– إلى التغيُّر العام في مُتغيِّر ما بمرور الوقت. على سبيل المثال، تُعَدُّ الزيادة التدريجية في مُتوسِّط درجة حرارة الأرض بسبب تغيُّر المناخ اتِّجاهًا.

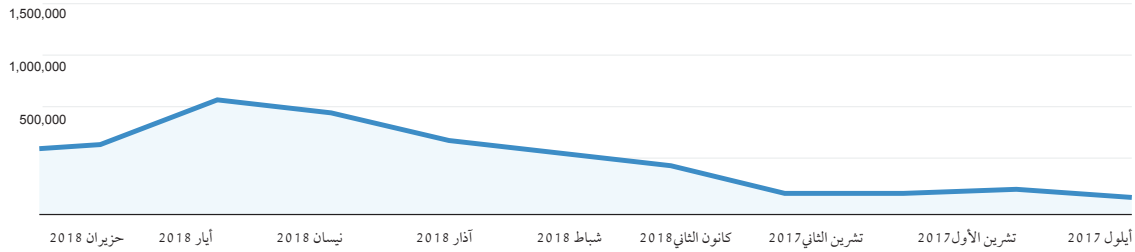
تُصنَّف الاتِّجاهات إلى ثلاثة أنواع، هي:

- **الاتِّجاه التصاعدي** (Uptrend): يشير هذا الاتِّجاه إلى حركة تصاعدية عامة في البيانات.
- **الاتِّجاه التنازلي** (Downtrend): يُظهِر هذا الاتِّجاه حركة تنازلية عامة.
- **الاتِّجاه الجانبي** (Sideways Trend): يَصِف هذا الاتِّجاه حالة يتذبذب فيها المُتغيِّر من دون أنْ يُظهِر حركة صافية مُحدَّدة.

يساعد تحليل الاتِّجاهات على التنبُّؤ بالتغيُّرات المستقبلية، ويُسهِم في تحديد المخاطر المُحتمَلة، وتعديل الاستراتيجيات وَفقًا لها.

مثال:

يُبيِّن الشكل (1-3) رسمًا بيانيًّا لعدد مَرّات زيارة صفحة إلكترونية مُعيَّنة، بَدْءًا بشهر تشرين الأوَّل عام 2017م، وانتهاءً بشهر حزيران عام 2018م. أُلاحِظ أنَّ اتِّجاه البيانات قد تزايد من شهر كانون الثاني إلى منتصف شهر أيّار، ثمَّ أخذ يتناقص من منتصف شهر أيّار إلى شهر حزيران؛ ما يعني أنَّ البيانات قد لا تسلك اتِّجاهًا واحدًا باستمرار، وإنَّما تتغيَّر وَفقًا لعدد من المُتغيِّرات.



الشكل (1-3): رسم بياني لعدد مَرّات زيارة صفحة إلكترونية مُعيَّنة خلال زمن مُحدَّد (June 2018–October 2017).

إثراء

يُعَدُّ الموقع الإلكتروني (https://trends.google.com/trends) أداة فاعلة لعرض البيانات واستكشافها؛ فهو يُمكِّن المُستخدِم من معرفة اتِّجاهات البحث في موضوع مُعيَّن على مدار الوقت، عن طريق عرض البيانات والإحصائيات المُتعلِّقة بما يبحث عنه الناس، وتعرُّف التوجُّهات العالمية والمحلية في مختلف المجالات. ومن ثَمَّ يُمكِن استخدام هذه الأداة في مجالات مُتنوِّعة ومُتعدِّدة، مثل: البحث العلمي، وتحليل البيانات، والتسويق.

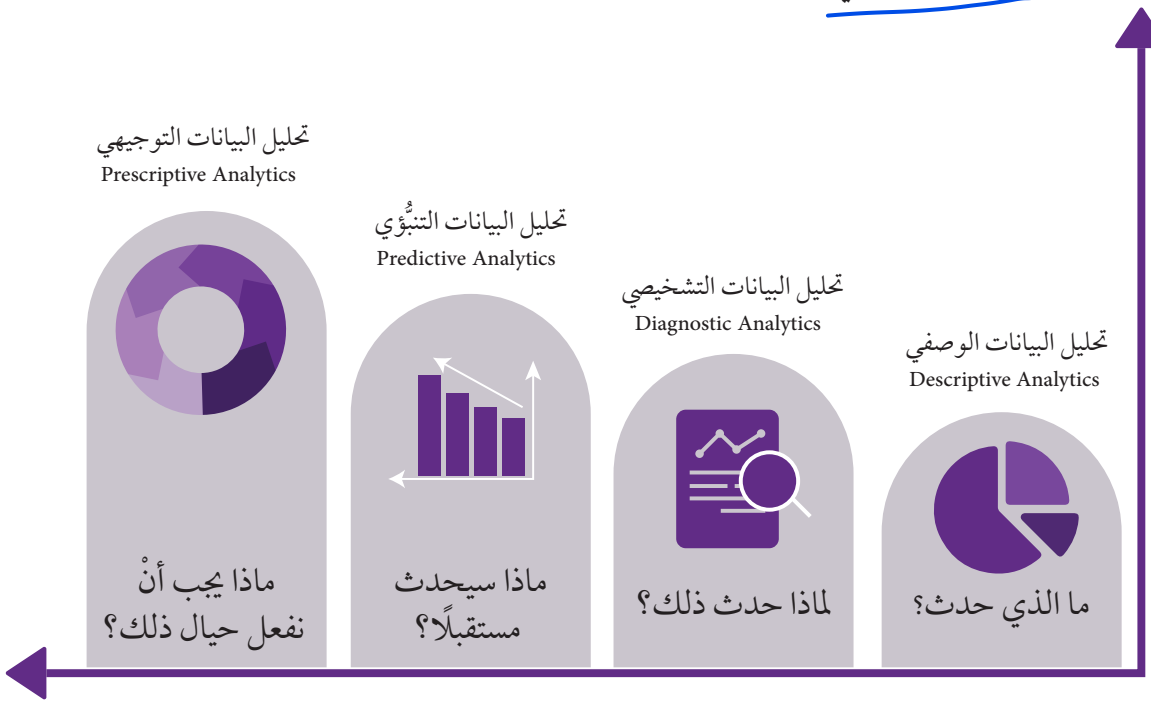
على سبيل المثال، عند استكشاف الاهتمام بعلم البيانات بين عام 2019م وعام 2024م، تبيَّن وجود زيادة ملحوظة في حجم البحث كما هو مُوضَّح في الشكل (1-4)؛ ما يشير إلى تزايد الاهتمام بهذا المجال خلال هذه الأعوام.

الشكل (1-4): رسم تخطيطي لاستكشاف الاهتمام بعلم البيانات خلال الأعوام (2019 – 2024 م).

أنواع تحليل البيانات (Types of Data Analysis):

تُعَدُّ البيانات أداة أساسية للإجابة عن الأسئلة التي تدعم اتِّخاذ قرارات مُهِمَّة، وهي تُسهِم أيضًا في إيجاد تساؤلات جديدة تُمثِّل وسيلة لاستكشاف الاتِّجاهات وفتح آفاق معرفية تُعزِّز مناحي الإبداع والابتكار. غير أنَّ تحقيق الفائدة القصوى من البيانات يتطلَّب فهم أنواع التحليل المختلفة للتمكُّن من اختيار الأنسب منها. أنظر الشكل (1-5).

- **تحليل البيانات الوصفي (Descriptive Analytics):** "ما الذي حدث؟".
- **تحليل البيانات التشخيصي (Diagnostic Analytics):** "لماذا حدث ذلك؟".
- **تحليل البيانات التنبُّؤي (Predictive Analytics):** "ماذا سيحدث مستقبلًا؟".
- **تحليل البيانات التوجيهي (Prescriptive Analytics):** "ماذا يجب أنْ نفعل حيال ذلك؟".



الشكل (1-5): أنواع تحليل البيانات مُرتَّبةً بحسب صعوبتها وقيمتها.

في ما يأتي بيان لكل نوع من أنواع تحليل البيانات:

- **تحليل البيانات الوصفي (Descriptive Analytics):**

يُستخدَم التحليل الوصفي في تلخيص الخصائص الرئيسة لمجموعة البيانات ووصفها، وهو يجيب عن السؤال الذي نصُّه: ما الذي حدث؟ بوجه عام، يبحث التحليل الوصفي في البيانات الرقمية، وهو يُعْنى بتقديم صورة واضحة وشاملة عن البيانات باستخدام تقنيات إحصائية ورسوم بيانية.

على سبيل المثال، يُمكِن لشركة تقنية تعليمية تعمل على تطوير نظام إدارة تعلُّم (LMS) للمدارس والجامعات أنْ تستخدم التحليل الوصفي في تلخيص البيانات التي تجمعها من الطلبة والمُعلِّمين/ المُعلِّمات المُستخدِمين للنظام. فهذا التحليل يساعد على فهم كيفية استخدام النظام، واكتشاف أنماط الاستخدام، مثل: أكثر المزايا شيوعًا، وأكثر الطلبة نشاطًا، وأفضل المُعلِّمين/ المُعلِّمات أداءً.

التقنيات المُستخدَمة في تحليل البيانات الوصفي:

أ. **مقاييس التكرار والجداول التكرارية:** تهدف هذه التقنية إلى تحديد أكثر القِيَم شيوعًا في البيانات.

ب. **مقاييس التشتُّت ومقاييس النزعة المركزية:** تهدف هذه التقنية إلى تلخيص البيانات. ومن أمثلتها: الوسط، والوسيط، والمنوال، والتباين، ومقاييس النزعة المركزية والانحراف المعياري.

ج. **تحليل توزيع البيانات:** تهدف هذه التقنية إلى فهم نمط توزيع البيانات، مثل تحديد إذا كانت تتبع التوزيع الطبيعي أم لا.

- **تحليل البيانات التشخيصي (Diagnostic Analytics):**

يُستخدَم هذا النوع من تحليل البيانات في استكشاف الأسباب الجذرية للمشكلات أو الظواهر المُكتشَفة، وهو يجيب عن السؤال الذي نصُّه: لماذا حدث ذلك؟ ما يساعد على تحديد الأسباب الرئيسة لاتِّخاذ خطوات تصحيحية.

على غرار المثال السابق عن الشركة التقنية التعليمية التي تعكف على تطوير نظام إدارة التعلُّم (LMS)، إذا أبلغ أحد المُعلِّمين/ المُعلِّمات مثلًا عن مشكلة في خصيصة مُعيَّنة بالنظام، فقد تستخدم الشركة التحليل التشخيصي في تحديد سبب المشكلة الجذري، مثل: الخطأ البرمجي، والنقص في تدريب المُعلِّمين/ المُعلِّمات على استخدام هذه الخصيصة. وفي مثال آخر، إذا أظهر التحليل الوصفي أنَّ مبيعات مُنتَج مُعيَّن انخفضت في شهر مُعيَّن، فإنَّ التحليل التشخيصي يسعى إلى معرفة أسباب هذا الانخفاض، مثل: التغيُّرات والتقلُّبات في الأسواق، وظهور منافسين جُدُد، والتغيُّر في سلوكات العملاء.

التقنيات المُستخدَمة في تحليل البيانات التشخيصي:

أ. **تحليل السلاسل الزمنية:** تهدف هذه التقنية إلى تحليل البيانات، وتقييم التغيُّرات بمرور الوقت، وتحديد الأحداث المُؤثِّرة.

ب. **تحليل الارتباط:** تهدف هذه التقنية إلى تحديد مدى قوَّة العلاقة بين المُتغيِّرات ونوعها؛ ما يساعد على كشف العوامل المُؤثِّرة.

- **تحليل البيانات التنبُّؤي (Predictive Analytics):**

يُستخدَم هذا النوع من تحليل البيانات في التنبُّؤ بالاتِّجاهات أو الأحداث المستقبلية، وهو يجيب عن السؤال الذي مفاده: ماذا يُمكِن أنْ يحدث مستقبلًا؟ يعتمد التحليل التنبُّؤي على البيانات المؤرشفة والنماذج الإحصائية وخوارزميات تعلُّم الآلة في استخلاص توقُّعات دقيقة تدعم اتِّخاذ قرارات استراتيجية.

على سبيل المثال، يُمكِن لشركة تقنية تعليمية تُعْنى بتطوير نظام إدارة التعلُّم (LMS) أنْ تستخدم التحليل التنبُّؤي في توقُّع أنماط الاستخدام المستقبلية. فقد يساعد هذا النوع من التحليل مثلًا على تحديد الطلبة الذين يواجهون صعوبات في محتوى مُعيَّن، أو تحديد أكثر المُعلِّمين والمُعلِّمات الذين يُتوقَّع تبنّيهم استراتيجيات تدريس مُحدَّدة. ومن ثَمَّ يُمكِن لهذه التوقُّعات أنْ تساعد الشركة على تحسين تصميم النظام وتسويقه، ودعم المُستخدِمين المُستهدَفين بشكل استباقي.

يستخدم التحليل التنبُّؤي أدوات مُتعدِّدة، مثل: أداة بايثون (Python) التي تعتمد على مكتبات مُتخصِّصة (مثل: (Pandas)، و(NumPy)، و(Scikit-Learn)) في تحليل البيانات وتطوير النماذج التنبُّؤية، وأداة (SQL) التي يُعتمَد عليها في استخراج البيانات من قواعد البيانات ومعالجتها. كذلك تُوفِّر بعض المِنصّات (مثل Microsoft Azure Machine Learning) بيئة مَرِنة لتطوير نماذج تعلُّم الآلة ونشرها.

- **تحليل البيانات التوجيهي (Prescriptive Analytics):**

يُعَدُّ هذا النوع خطوة مُتقدِّمة في تحليل البيانات؛ فهو يهدف إلى الإجابة عن السؤال الآتي: ماذا يجب أنْ نفعل حيال ذلك؟ يستخدم تحليل البيانات التوجيهي النتائج المُستخلَصة من الأنواع الأُخرى (مثل: التحليل الوصفي، والتحليل التشخيصي، والتحليل التنبُّؤي) في تقديم توصيات عملية وقابلة للتنفيذ حيال كيفية التصرُّف في مواجهة المشكلات، أو عند اتِّخاذ قرارات استراتيجية.
على سبيل المثال، إذا أظهر التحليل التنبُّؤي أنَّ الطلبة قد يواجهون صعوبة في تعلُّم مادة مُعيَّنة، فقد تستخدم شركة التقنية التعليمية التحليل التوجيهي في تحديد استراتيجيات مُلائِمة لدعم هؤلاء الطلبة، مثل: توفير موارد إضافية، وتقديم خدمات دعم مُخصَّصة.

التقنيات المُستخدَمة في تحليل البيانات التوجيهي:

ج. **الخوارزميات:** تُستخدَم الخوارزميات في تنفيذ المهام بناءً على قواعد مُحدَّدة؛ إذ تُستخدَم في البنوك مثلًا خوارزميات لمراقبة إنفاق العملاء، ويُمكِن لهذه الخوارزميات التوصية بإلغاء تنشيط بطاقات الائتمان عند الاشتباه في حدوث احتيال.

د. **خوارزميات تعلُّم الآلة** (Machine Learning): تُعَدُّ خوارزميات تعلُّم الآلة أداة فاعلة للتحليل التوجيهي؛ فهي تساعد على معالجة كَمٍّ كبير من البيانات لتحديد أفضل مسارات العمل. تُستخدَم هذه الخوارزميات في تقديم توصيات دقيقة عن طريق تدريب النماذج باستخدام معادلات رياضية وقواعد شرطية.

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أنواع جديدة ومُتقدِّمة لتحليل البيانات، ثمَّ أكتب تقريرًا عن ذلك، وأُضمِّنه بيانًا لأهمية كل نوع من أنواع تحليل البيانات في دعم القرارات وتطوير العمليات، وذكرًا للتقنيات والأدوات المُستخدَمة في تنفيذ كلٍّ من هذه الأنواع. بعد ذلك أقرأ التقرير أمام الزملاء/ الزميلات في الصف، ثمَّ أُناقِشهم فيه.

نشاط

أدرس الحالة الآتية، ثمَّ أُبيِّن – بالتعاون مع أفراد مجموعتي – كيف يُمكِن استخدام أنواع تحليل البيانات في حَلِّ المشكلة:

"لاحظت إدارة شركة لبيع الملابس تغيُّرًا في حجم مبيعاتها خلال الأشهر الثلاثة الأخيرة، وهي تسعى إلى فهم هذا التغيُّر، واتِّخاذ قرارات ناجعة لتحسين مُعدَّل المبيعات".

بعد ذلك أُقارِن النتائج التي توصَّلنا إليها في المجموعة بما توصَّل إليه أفراد المجموعات الأُخرى من نتائج، ثمَّ نتبادل معًا التغذية الراجعة بإشراف المُعلِّم/ المُعلِّمة.

المواطنة الرقمية:

- **الخصوصية** (Privacy): أحترم خصوصية الأفراد أثناء عملية جمع البيانات وتحليلها، وأتأكَّد أنَّ البيانات الشخصية تُجمَع وتُستخدَم وَفقًا للقوانين والسياسات المعمول بها.
- **الشفافية** (Transparency): أحرص على توفير معلومات واضحة عن الهدف من تحليل البيانات وكيفية استخدامها.
- **المساءلة** (Accountability): أتحمَّل المسؤولية عن القرارات والإجراءات التي تُتَّخَذ بناءً على تحليل البيانات.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُعرِّف المصطلحات الآتية:

1- البيانات (Data).

2- تحليل البيانات (Data analysis).

3- التحليل التنبُّؤي (Predictive Analysis).

السؤال الثاني: أُعدِّد أنواع البيانات، وأذكر مثالًا واحدًا على كل نوع منها.

السؤال الثالث: أرسم مُخطَّطًا يُبيِّن مراحل عملية تحليل البيانات.

السؤال الرابع: أذكر تقنية واحدة مُستخدَمة في كل نوع من أنواع تحليل البيانات الواردة في الجدول الآتي:

نوع تحليل البيانات	التقنية المُستخدَمة
التحليل التشخيصي	
التحليل الوصفي	
التحليل التنبُّؤي	
التحليل التوجيهي	

السؤال الخامس: أختار رمز الإجابة الصحيحة في كلٍّ ممّا يأتي:

1. **نوع البيانات الذي يُمكِن إجراء عمليات حسابية عليه هو:**
   - أ. البيانات العددية.
   - ب. البيانات النصية.
   - ج. البيانات المنطقية.
   - د. البيانات الترتيبية.

2. **أحد أنواع البيانات الآتية من البيانات النصية:**
   - أ. 123
   - ب. "Hello".
   - ج. True.
   - د. 45.6

3. **نوع البيانات الذي يُمثِّل مجموعة مُرتَّبة من العناصر التي لا يُمكِن تعديلها بعد تعريفها هو:**
   أ. القوائم.
   ب. الصفوف.
   ج. المجموعات.
   د. القواميس.
4. **يهدف التحليل التشخيصي (Diagnostic Analytics) إلى:**
   أ. تلخيص البيانات.
   ب. التنبُّؤ بالأحداث المستقبلية.
   ج. تحديد الأسباب الجذرية للمشكلات.
   د. تقديم توصيات عملية.
5. **نوع التحليل الذي يجيب عن سؤال:** ماذا يجب أنْ نفعل حيال ذلك؟ هو:
   أ. التحليل الوصفي.
   ب. التحليل التشخيصي.
   ج. التحليل التنبُّؤي.
   د. التحليل التوجيهي.

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أُصنِّف الجمل الآتية إلى نوع تحليل البيانات الذي تُعبِّر عنه كلٌّ منها:

1. "إذا زدنا قيمة الخصومات بما نسبته 10٪، فإنَّ النموذج يقترح أنْ يرتفع مُعدَّل المبيعات بنسبة 15٪".
2. "الرسوم البيانية تُوضِّح أنَّ الأداء في الربع الأخير كان هو الأفضل خلال العام".
3. "تحليل أسباب زيادة الشكاوى يُظهِر أنَّ التأخير في الشحن هو المشكلة الأساسية".
4. "النظام يقترح جدولة الموظفين أثناء أوقات الذروة بناءً على البيانات التنبُّؤية".
5. "البيانات تُظهِر أنَّ العملاء يُفضِّلون التسوُّق يوم الجمعة أكثر من الأيام الأُخرى".

السؤال الثاني: جاء في تقرير لإحدى المدارس: "أظهرت نتائج تحليل علامات الطلبة أنَّ الأداء في بعض المواد كان أقلَّ من المُتوقَّع؛ ما أثار قلق الإدارة". بناءً على ذلك، أُبيِّن كيف يُمكِن للإدارة استخدام أنواع التحليل المختلفة في فهم المشكلة واعتماد الحلول المناسبة لتحسين أداء الطلبة.

الدرس الثاني

تقنيات تحليل البيانات (Data Analysis Techniques)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس الأدوات والطرائق الخاصة بجمع البيانات وتقنيات تحليلها واستكشافها، ثمَّ أستخدم هذه الأدوات في تحديد اتِّجاه البيانات ونزعتها، وأُفاضِل بينها بناءً على طبيعة المشكلة التي أحتاج إلى حَلِّها. كذلك سأُوظِّف تقنيات التحليل في اكتشاف الأنماط التي تُمثِّل الأنظمة المُعقَّدة.

المفاهيم والمصطلحات:

أدوات جمع البيانات (Data Collection Tools)، استكشاف البيانات (Data Exploration)، أدوات تحليل البيانات (Data Analysis Tools)، المكتبات (Libraries)، الاستبانات (Questionnaires)، الاستطلاعات (Surveys)، دراسة الحالة (Case Study)، مجموعات التركيز (Focus Groups)، الملاحظة (Observation)، عرض البيانات (Data Visualization).

نتاجات التعلُّم (Learning Outcomes):

- أُوضِّح طرائق جمع البيانات واستخداماتها.
- أُبيِّن مراحل استكشاف البيانات.
- أُميِّز بين أدوات جمع البيانات وتحليلها واستكشافها.
- أستخدم الأدوات والتقنيات الخاصة بتحليل البيانات في تحديد اتِّجاه نزعة المركزية.
- أُفاضِل بين الأدوات والتقنيات المُتوافِرة التي تُستخدَم في تحليل البيانات اعتمادًا على المشكلة التي يُراد حَلُّها.

تؤدّي الأدوات الخاصة بجمع البيانات وتحليلها دورًا أساسيًّا في استكشاف الأنماط والاتِّجاهات الكامنة داخل البيانات وفهمها. وتُعَدُّ عملية استكشاف الأنماط والاتِّجاهات واحدة من العمليات المُهِمَّة التي تهدف إلى استخراج معلومات قيِّمة من كَمٍّ كبير من البيانات، لا سيَّما في ظلِّ التدفُّق الهائل للبيانات الذي يشهده العالَم اليوم.

نشاط تمهيدي

بناءً على تجارب سابقة، أُبيِّن طريقة واحدة يُمكِن استخدامها في كلٍّ ممّا يأتي:

- جمع بيانات شخصية عن طلبة الصف من طرف المُعلِّم/ المُعلِّمة.
- تعرُّف المشكلات التعليمية التي يعانيها أحد الطلبة.
- تعرُّف حالة الطقس في المدينة التي أسكن فيها على مدار أسبوع كامل.

هل توجد طرائق أُخرى لجمع البيانات غير تلك التي دوَّنْتُها.

طرائق جمع البيانات:

توجد طرائق عِدَّة لجمع البيانات، أبرزها:

1. **المقابلات (Interviews):** تُعَدُّ المقابلات واحدة من أكثر طرائق جمع البيانات شيوعًا وفاعليةً، وهي تمتاز بتنوُّع أساليبها التي تُناسِب مختلف الأهداف. كذلك تُعَدُّ المقابلات أداة مَرِنة وفاعلة لجمع البيانات النوعية.

تختلف أشكال المقابلات وأنواعها تبعًا لاعتبارات عديدة، أبرزها:

أ. **عدد المشاركين:**

- **المقابلة الفردية:** تُركِّز المقابلة الفردية على شخص واحد للحصول على بيانات عميقة ومُفصَّلة عنه.
- **المقابلة الجماعية:** تشمل المقابلة الجماعية الالتقاء بمجموعة من الأشخاص؛ ما يتيح جمع آراء مُتعدِّدة في آنٍ معًا.

ب. **طبيعة المقابلة:**

- **المقابلة الرسمية:** يُخطَّط للمقابلة الرسمية مُسبَقًا بأسئلة مُجهَّزة وواضحة.
- **المقابلة غير الرسمية:** تُعقَد المقابلة غير الرسمية من دون تخطيط دقيق ومُحْكم، وفيها تكون الأسئلة مفتوحة لتتيح نقاشًا حُرًّا.

ج. **وسيلة التنفيذ:**

- **وجهًا لوجه:** يمتاز هذا النوع من المقابلات بالتفاعل المباشر بين الطرفين.
- **عن بُعْد:** يمتاز هذا النوع من المقابلات باستخدام وسائل الاتصال، مثل: الهاتف، والتطبيقات المرئية.

2. **الاستبانات (Questionnaires):** تُعَدُّ الاستبانة أداة لجمع البيانات، وهي تتضمَّن مجموعة من الأسئلة (ورقية أو إلكترونية) يجيب عنها أفراد من مجموعة مُستهدَفة. تُعَدُّ الاستبانة أداة مثالية إذا كان الهدف منها هو جمع بيانات شاملة ومُوحَّدة تُسهِّل عملية التحليل واتِّخاذ القرارات، وهي تمتاز بتوافرها لشريحة واسعة من الأفراد، وسهولة جمعها للبيانات وتحليلها.

نشاط عملي

أُنشِئ استبانة إلكترونية باستخدام نماذج جوجل (Google Forms)، وأُضمِّنها مجموعة من الأسئلة بهدف جمع بيانات عن الطلبة في صفي، مثل: الاسم، والعمر، وتاريخ الميلاد. أُشارِك الاستبانة مع زملائي/ زميلاتي في الصف، ثمَّ أطلب إليهم تعبئتها والإجابة عن الأسئلة الواردة فيها. بعد ذلك، أطَّلِع على نتائج الاستبانة، وأُلاحِظ السرعة والسهولة في الحصول على البيانات وتحليلها.

روابط مساعدة:

لإنشاء استبانة إلكترونية: https://www.google.com/forms/about

للحصول على إرشادات خاصة بإنشاء استبانة إلكترونية:

https://support.google.com/a/users/answer/9303071?hl=ar

3. **المسوحات أو الاستطلاعات (Surveys):** تُشبِه هذه الطريقة الاستبانات من حيث التصميم، لكنَّها تختلف عنها في جمعها بيانات من عدد أكبر من الأفراد. وفيها تُكتَب مجموعة من الأسئلة المُخصَّصة، ثمَّ تُوزَّع على عيِّنة مختارة بعناية من أفراد المجتمع المُستهدَف. تُعَدُّ المسوحات وسيلة فاعلة لجمع كَمٍّ كبير من البيانات بهدف استكشاف مواقف الأفراد، وتحديد الفروق في السلوكات، ومراقبة التغيُّرات المُحتمَلة بمرور الوقت، وصولًا إلى فهم الاتِّجاهات العامة داخل المجتمعات.

4. **الملاحظة (Observation):** تُعَدُّ الملاحظة إحدى أدوات جمع البيانات التي تعتمد على مراقبة الباحث لعناصر مُعيَّنة أو عمليات وأحداث مُحدَّدة، إضافةً إلى تسجيل ملاحظاته بدِقَّة. تُستخدَم هذه الطريقة كثيرًا في مراقبة السلوكات والظواهر بصورة مباشرة ودقيقة؛ ما يتيح جمع بيانات موضوعية عن الواقع بعيدًا عن التحيُّز أو التعصُّب. من الأمثلة على الملاحظة: مراقبة سلوك الطلبة الصغار داخل الصفوف؛ لفهم كيفية تفاعلهم مع المُعلِّمين/ المُعلِّمات والزملاء/ الزميلات، ودراسة تفاعل الأفراد في بيئة العمل؛ لفهم ديناميكيات الفريق.

5. **مجموعات التركيز (Focus Groups):** تُعَدُّ هذه الطريقة واحدة من الطرائق النوعية لجمع البيانات، وهي تعتمد على مناقشات جماعية مُنظَّمة تُركِّز على موضوع مُعيَّن. تضمُّ المجموعة عادةً (6- 12) فردًا لديهم اهتمام (أو خبرة) مُشترَك بالموضوع الذي يُراد دراسته. تهدف هذه الطريقة إلى فهم تصوُّرات المشاركين وآرائهم ومواقفهم عن طريق طرح مجموعة من الأسئلة التوجيهية التي تتعلَّق بالموضوع. من الأمثلة على مجموعات التركيز: دراسة آراء المُستهلِكين بخصوص مُنتَج جديد، واستكشاف مواقف المُعلِّمين/ المُعلِّمات تجاه تطبيق استراتيجيات تعليمية مُبتكَرة.

6. **دراسة الحالة (Case Study):** تُعَدُّ دراسة الحالة أسلوبًا بحثيًّا يُركِّز على التحليل المُعمَّق لوحدة مُحدَّدة، مثل: الفرد، والأُسْرة، والمجتمع، والمدرسة، والموقع المُعيَّن. تهدف هذه الطريقة إلى جمع بيانات شاملة عن هذه الوحدة؛ بُغْيَةَ فهم الظاهرة المدروسة بوضوح ودِقَّة، وهي تعتمد على ملاحظة الظاهرة في سياقها الطبيعي، مع توثيق التفاصيل المحيطة بها. يُذكَر أنَّ دراسة الحالة تُستخدَم على نطاق واسع في المجالات الاجتماعية والتعليمية والطبية.

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن طرائق أُخرى لجمع البيانات، ثمَّ أُشارِك الزملاء/ الزميلات في ما أتوصَّل إليه من نتائج.

نشاط فردي

أقرأ الجمل الآتية، ثمَّ أُبيِّن طريقة جمع البيانات المناسبة لكلٍّ منها:

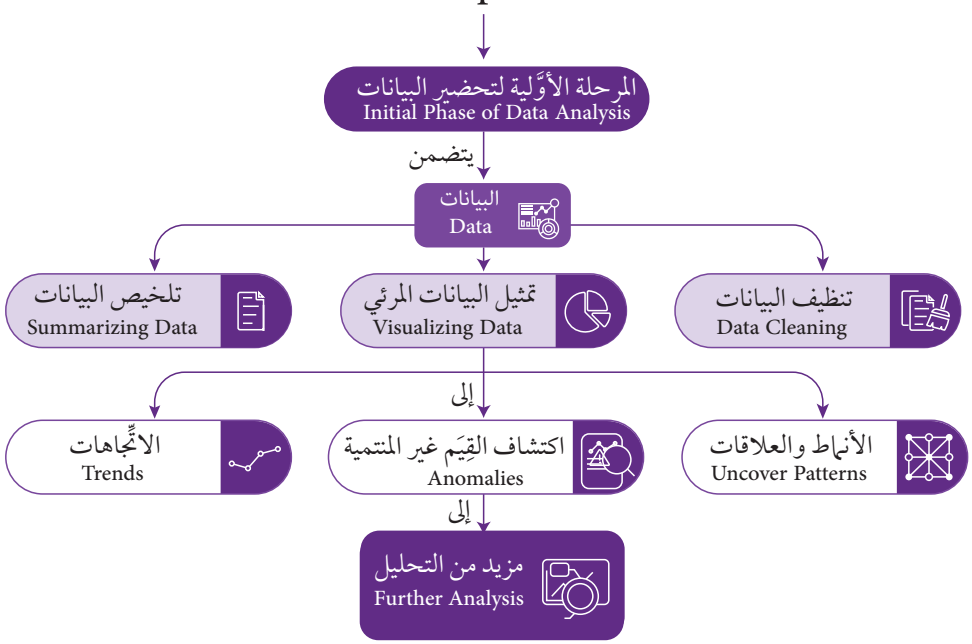
- "شركة تريد تقييم رضا العملاء عن مُنتَج جديد بعد إطلاقه".
- "مَدْرسة تسعى إلى جمع معلومات عن حضور الطلبة اليومي على مدار العام".
- "باحث يرغب في فهم أسباب التغيُّر في مُعدَّلات البطالة في منطقة مُعيَّنة".
- "متجر إلكتروني يريد معرفة أكثر المُنتَجات شيوعًا بين العملاء".
- "مدينة تسعى إلى جمع معلومات عن مستوى الرضا عن وسائل النقل العام".

أُقارِن إجابتي بإجابات الزملاء/ الزميلات في الصف، ثمَّ نُناقِش معًا كيف يُمكِن تحديد أنسب طريقة لجمع البيانات في كل حالة مع التبرير.

استكشاف البيانات (Data Exploration):

يُمثِّل استكشاف البيانات المرحلة الأولى من مراحل تحليل البيانات، وهي مرحلة مُهِمَّة لفهم هيكل البيانات، واكتشاف الأنماط والعلاقات بينها . أنظر الشكل (2-1).

مراحل استكشاف البيانات

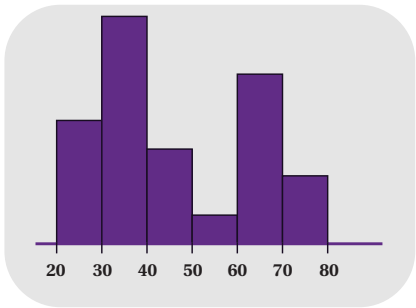
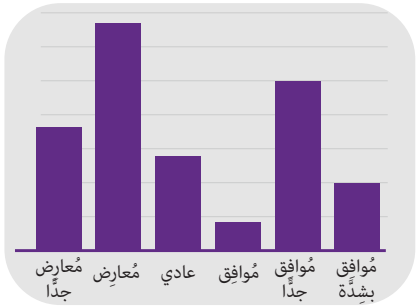


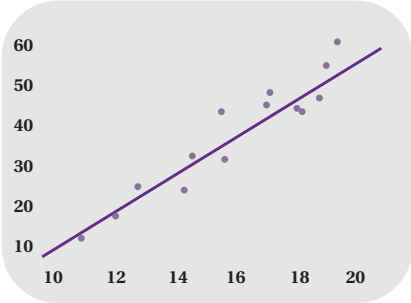
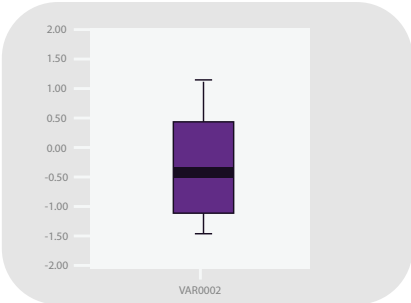
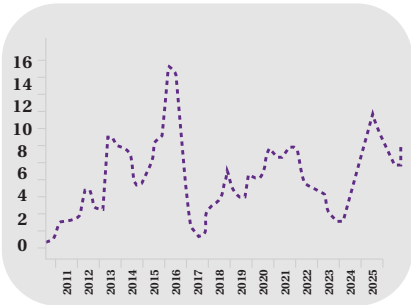
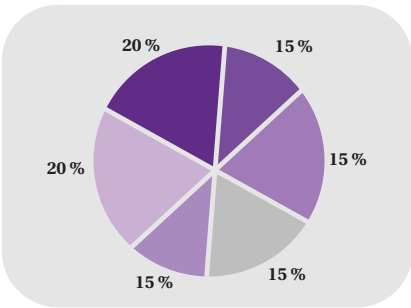
الشكل (2-1): مراحل استكشاف البيانات وعلاقتها بالتحليل العميق.

يُبيِّن الشكل (2-1) مراحل استكشاف البيانات (Data Exploration)، وهي:

1. **المرحلة الأوَّلية لتحضير البيانات:** تبدأ هذه المرحلة بتحديد المشكلة أو السؤال الذي نرغب في الإجابة عنه باستكشاف البيانات، ويُعَدُّ وجود هدف واضح خطوة أساسية لتوجيه عملية الاستكشاف بصورة فاعلة. بعد ذلك تُجمَع البيانات اللازمة للتحليل من مصادر مُتنوِّعة، مثل: قواعد البيانات، والواجهات البرمجية (APIs)، وجداول البيانات، والملفات النصية. يُذكَر أنَّ هذه المرحلة تُعَدُّ حجر الأساس في استكشاف البيانات وتحضيرها لتحليل أعمق.
2. **تنظيف البيانات (Data Cleaning):** تتضمَّن هذه المرحلة إزالة الأخطاء والقِيَم المفقودة والقِيَم غير المنتمية لضمان جودة البيانات وتحضيرها للتحليل.
3. **تمثيل البيانات المرئي (Data Visualizing):** يُقصَد بذلك إنشاء تمثيلات بصرية للبيانات بهدف الكشف عن الأنماط والعلاقات بينها (Uncover Patterns) والاتِّجاهات (Trends)، واكتشاف القِيَم غير المنتمية (Anomalies) بسهولة.

يعتمد اختيار طرائق عرض البيانات على نوع البيانات المُستخدَمة. ومن التقنيات الشائعة لتمثيل البيانات:

المُخطَّطات (Histograms): تُستخدَم المُخطَّطات في عرض التوزيعات التكرارية، وهي مناسبة للبيانات العددية المُتَّصِلة.	
المُخطَّطات العمودية (Bar Charts): تُستخدَم هذه المُخطَّطات في عرض التوزيعات التكرارية، وهي مناسبة للبيانات التصنيفية.	

مُخطَّطات التشتُّت (Scatter Plots): تُظهِر هذه المُخطَّطات العلاقة بيـــن مُتغيِّرين عدديين، وهي تُســـهِم في الكشـــف عن الأنماط أو الاتِّجاهات.	
المُخطَّطات الصندوقية (Box Plots): تُستخدَم هذه المُخطَّطات فـــي توضيح نطاق البيانات وتوزيعها، مع التركيز على القِيَم غير المنتمية.	
مُخطَّطات السلاســـل الزمنية (Time Series Plots): تُســـتخدَم هـــذه المُخطَّطـــات في استكشاف البيانات على مدى زمني؛ ما يساعد على فهم الاتِّجاهات أو التغيُّرات الموسمية.	
القطاعات الدائرية (Pie Chart): تُستخدَم هذه المُخطَّطات في عرض النســـب المئوية بين الفئات، وهي مناسبة للفئات المحدودة العدد (3–5 فئات).	

نشاط جماعي

أدرس التمثيلات البيانية السابقة، ثمَّ أُحلِّلها - بالتعاون مع أفراد مجموعتي- لفهم البيانات بصورة أوَّلية، ثمَّ أُحدِّد مقاييس النزعة المركزية، مثل: المُتوسِّط. كذلك أُحدِّد الاتِّجاه العام للبيانات. بعد ذلك أُحلِّل اتِّجاه النزعة؛ سواء كانت زيادةً، أو نقصانًا، أو ثباتًا، إضافةً إلى رصد أيَّة قِيَم غير منتمية أو بعيدة عن بقيَّة القِيَم تَظهر في التمثيل البياني.

أُناقِش أفراد مجموعتي في ملاحظاتي الأوَّلية، ثمَّ نُشارِكها مع أفراد المجموعات الأُخرى.

4. **تلخيص البيانات** (Summarizing): تُستخدَم الإحصائيات الوصفية في تقديم رؤى واضحة وموجزة عن البيانات.

يتمثَّل الهدف الرئيس لاستكشاف البيانات في توجيه التحليل العميق (Further Analysis) عن طريق اكتشاف الأنماط والعلاقات وتحديد المشكلات (مثل: القِيَم غير المنتمية، والتوزيعات غير الطبيعية) وصولًا إلى تحسين جودة النتائج، علمًا بأنَّ نتائج الاستكشاف تُقدِّم جملة من الإرشادات لاختيار المعالجة الإحصائية المناسبة.

تنتهي هذه العملية باستخلاص نتائج واضحة ترتبط بأهداف التحليل، إضافةً إلى توثيق الخطوات والنتائج. ولمّا كانت هذه العملية تكرارية، فقد يتطلَّب استكشاف البيانات عمل مراجعات عديدة وإجراء تنقيح مستمر لتحقيق أفضل النتائج.

إضاءة

توصَف القيمة (Outlier) في الإحصاء بأنَّها قيمة بعيدة عن معظم القِيَم في مجموعة البيانات.

مثال:

يُبيِّن الجدول المجاور كيف يُمكِن تحديد القيمة المُتطرِّفة في مجموعة بيانات تُمثِّل نتائج امتحان تحصيلي لمجموعة من الطلبة البالغ عددهم (10) طلبة. أُلاحِظ أنَّ جميع علامات الطلبة تقع ضمن المدى (85-95) باستثناء الطالب الرابع؛ إذ انحرفت علامته عن النمط الرئيس لمجموعة البيانات. وتُعَدُّ هذه القيمة قيمة غير منتمية (Outlier).

الطالب	العلامة (100)
الطالب الأوَّل	85
الطالب الثاني	85
الطالب الثالث	90
الطالب الرابع	22
الطالب الخامس	92
الطالب السادس	95
الطالب السابع	91
الطالب الثامن	93
الطالب التاسع	87
الطالب العاشر	89

أدوات تحليل البيانات (Data Analysis Tools):

تشير أدوات تحليل البيانات إلى البرامج والتطبيقات المُستخدَمة في فهم البيانات وتحليلها على نحوٍ يُبرِز الأنماط، وتوفير المعلومات التي يُمكِن استخدامها في صنع القرارات. ولا شكَّ في أنَّ اختيار الأداة المناسبة يُفْضي إلى تحقيق أهداف التحليل كما خُطِّط لها.

تختلف أدوات تحليل البيانات تبعًا لاختلاف نوع البيانات، والغرض من التحليل. وهذه أبرز الأدوات:

1- لغة البرمجة بايثون (Python):

تمتاز لغة البرمجة بايثون بأنَّها مُتعدِّدة الاستخدامات، وسهلة التعلُّم، وهي تُستخدَم على نطاق واسع في تحليل البيانات؛ لأنَّها مفتوحة المصدر (Open Source)، ومجّانية. كذلك تمتاز لغة البرمجة بايثون بوجود مجتمع عالمي ضخم، وأكثر من (137000) مكتبة مُتخصِّصة في مختلف المجالات، وهي تتيح للمُستخدِمين إجراء معالجة مُتقدِّمة للبيانات، وعمل تحليل عددي دقيق.

من أبرز مكتبات برمجية بايثون لتحليل البيانات:

أ. **مكتبة (Pandas):** تُستخدَم هذه المكتبة في معالجة البيانات وعرضها، والتعامل مع الجداول بسهولة.

ب. **مكتبة (PySpark):** يُمكِن لهذه المكتبة التعامل مع البيانات الضخمة (Big Data) ومعالجتها.

ج. **مكتبة (NumPy):** الاسم الكامل لهذه المكتبة هو (Numerical Python)، وهي تتيح تنفيذ العمليات الحسابية والعمليات المنطقية بكفاءة عالية، ويُمكِنها التعامل مع المصفوفات والعمليات الرياضية.

د. **مكتبة (Scikit-learn):** تُعَدُّ هذه المكتبة واحدة من المكتبات الشاملة للتعلُّم الآلي والنمذجة الإحصائية، وهي تُستخدَم في التصنيف (Classification)، والانحدار (Regression)، والتجميع (Clustering)، وتقليل الأبعاد (Dimensionality Reduction).

ه. **مكتبة (SciPy):** الاسم الكامل لهذه المكتبة هو (Scientific Python)، وهي مكتبة شاملة تُوفِّر وظائف رياضية مُتقدِّمة مثل الجبر الخَطِّي، وتُناسِب بوجه خاص المهام التي تتطلَّب تحليلًا علميًّا وهندسيًّا مُتقدِّمًا، وتستلزم أساليب رياضية وإحصائية مُتخصِّصة.

و. **مكتبة (Matplotlib):** تتيح هذه المكتبة إنشاء الرسوم البيانية والمُخطَّطات، مثل: المُخطَّطات الخَطِّية، ومُخطَّطات التشتُّت.

ز. **مكتبة (Natural Language Toolkit: NLTK):** تُعْنى هذه المكتبة بمعالجة اللغة الطبيعية (NLP)، ويستفاد منها في تمكين جهاز الحاسوب من فهم النصوص البشرية (لغة الإنسان)، والتفاعل معها، وتحليلها، ومعالجة النصوص.

إضاءة

تُعرَّف المكتبة (Library) بأنَّها مجموعة من الوحدات (Modules)، تحتوي على تعليمات برمجية يُمكِن إعادة استخدامها بسهولة. تعمل المكتبات على تسهيل عملية البرمجة؛ إذ تُغْني عن كتابة التعليمات البرمجية نفسها مِرارًا وتَكرارًا في برامج مختلفة. وتُعَدُّ مكتبات برمجية بايثون رائدة في العديد من المجالات، مثل: التعلُّم الآلي، وعلوم البيانات، وتصوُّر البيانات، وما إلى ذلك.

نشاط فردي

أتذكَّر وظيفة كل وحدة ممّا يأتي من وحدات برمجية بايثون، سبق أنِ استخدمْتُها في الصف الحادي عشر:

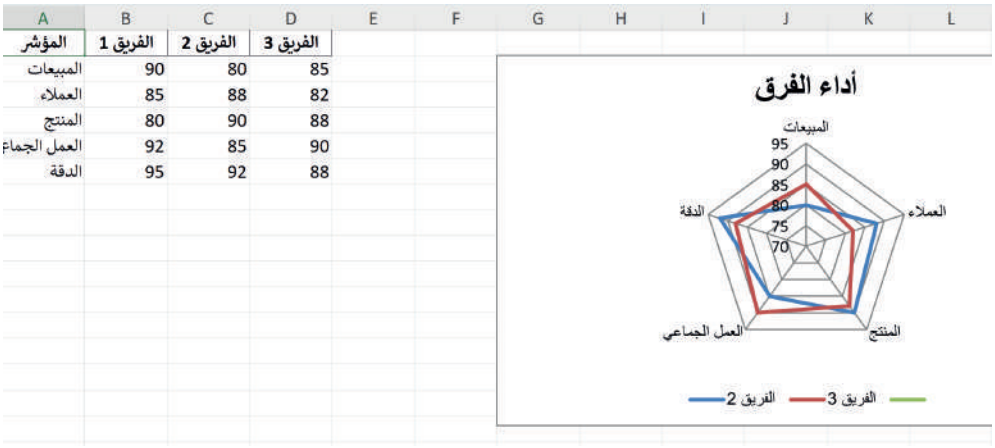
(Random Module)، و(Statistics Module)، ثمَّ أكتب مثالًا على كلٍّ منهما، ثمَّ أُناقِش إجابتي مع الزملاء/ الزميلات في الصف.

2- برمجية إكسل (Excel):

تُعَدُّ برمجية إكسل أداة أساسية وشائعة في تحليل البيانات في مختلف المجالات؛ إذ تُستخدَم في تلخيص البيانات المُمثَّلة في جداول، وإجراء العمليات الحسابية على الخلايا، وتمثيل البيانات باستخدام الرسوم البيانية، وفرز كَمٍّ كبير من البيانات وتنظيمها.

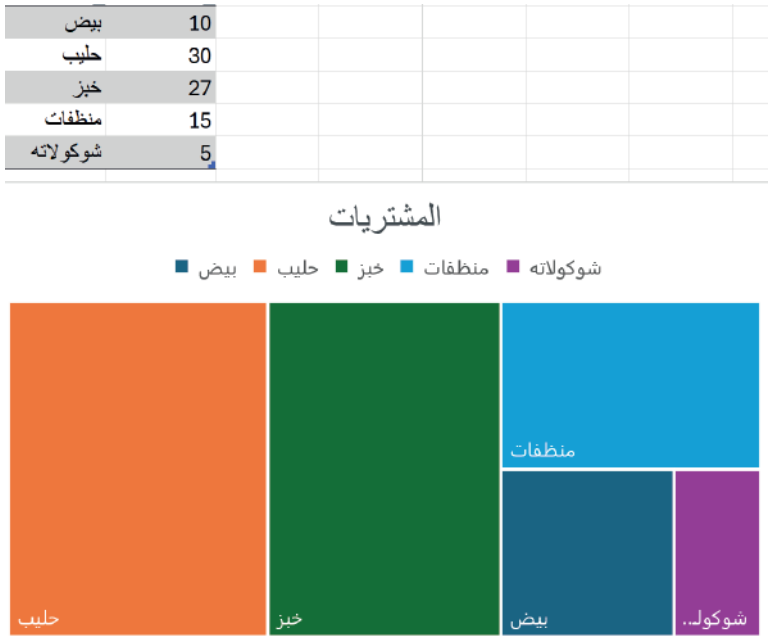
مزايا برمجية إكسل (Excel) في تحليل البيانات:

1. **إنشاء المُخطَّطات:** تُوفِّر برمجية إكسل ما لا يقلُّ عن (20) نوعًا مختلفًا من المُخطَّطات، بَدْءًا بالخيارات القياسية (مثل: المُخطَّطات الشريطية، ومُخطَّطات التشتُّت)، وانتهاءً بأكثر المُخطَّطات تعقيدًا، مثل: مُخطَّطات الرادار (Radar Charts) التي تُستخدَم في تمثيل البيانات التي تحتوي على ثلاثة مُتغيِّرات فأكثر، وتبدأ جميعها من النقطة نفسها. يُعَدُّ مُخطَّط الرادار أنسب المُخطَّطات للبيانات المُتعدِّدة المُتغيِّرات، مثل: تحليل تغيُّر درجات الحرارة في مواقع مختلفة خلال السنة، ومقارنة أداء المُنتَجات في مناطق مُتعدِّدة. أنظر الشكل (2-2) الذي يُبيِّن مثالًا على خرائط الرادار.



الشكل (2-2): مثال على تمثيل البيانات بمُخطَّطات خرائط الرادار باستخدام برمجية إكسل (Excel).

2. **توفير خرائط الأشجار (Treemap):** تُقدِّم هذه الخرائط عرضًا هرميًّا للبيانات باستخدام مستطيلات، وفيها تُمثَّل الفئات بالألوان. وهي تمتاز بسهولة عرض كثير من البيانات التي قد يكون من الصعب عرضها في أنواع أُخرى من المُخطَّطات، وكذلك اكتشاف الأنماط بسهولة، وإبراز أكثر العناصر أهمية، مثل تحديد أكثر المُنتَجات مبيعًا في متجر ما. أنظر الشكل (2-3) الذي يُبيِّن عنوانًا للجدول المجاور، وأُلاحِظ أنَّ الحليب هو أكثر المُنتَجات مبيعًافي إحدى المؤسسات.



الشكل (2-3): مثال على تمثيل البيانات بخرائط الأشجار باستخدام برمجية إكسل (Excel).

أُناقِش

بناءً على ما تعلَّمْتُه في برمجية إكسل مُسبَقًا، أُناقِش أفراد مجموعتي في مُحدِّدات برمجية إكسل الخاصة بتحليل البيانات، ثمَّ نُشارِك أفراد المجموعات الأُخرى في ما نتوصَّل إليه من نتائج.

إثراء

توجد أدوات أُخرى لتحليل البيانات، يختلف استخدامها تبعًا لاختلاف حجم البيانات، وتعقيدها، ونوع التحليل المطلوب. في ما يأتي استعراض لهذه الأدوات ومزاياها:

- **(R):** لغة برمجة مُصمَّمة خصيصًا للتحليل الإحصائي وتمثيل البيانات، وهي تمتاز بقدرتها على إجراء النمذجة الإحصائية المُتقدِّمة واختبار الفرضيات، إضافةً إلى توفيرها مكتبات شاملة لمعالجة البيانات وتصويرها. ومن ثَمَّ، فهي تُعَدُّ خيارًا مثاليًّا للبحوث الأكاديمية والحوسبة الإحصائية.
- **(SQL):** لغة استعلام مُنظَّمة تُستخدَم في الاستعلام عن قواعد البيانات العلائقية وإدارتها. تمتاز هذه الأداة بقدرتها على استخراج البيانات ومعالجتها بكفاءة، وهي لغة قياسية تدعم أنظمة قواعد البيانات المُتعدِّدة، مثل (MySQL)، وتُعَدُّ أداة أساسية للتعامل مع البيانات الهيكلية.
- **(Power BI):** أداة ذكاء الأعمال من شركة مايكروسوفت، وهي تُستخدَم في تمثيل البيانات، وإنشاء تقارير تفاعلية. تمتاز هذه الأداة بسهولة التكامل مع مُنتَجات مايكروسوفت، مثل: (Excel)، و(Azure)، وتُعَدُّ مثالية للتعاون عبر السحابة.
- **(Google BigQuery):** مستودع بيانات مُدار كاملًا عن طريق السحابة من جوجل (Google)، وهو مُصمَّم لتحليل البيانات الضخمة بكفاءة، ويمتاز بقدرته على التوسُّع لمعالجة مجموعات عديدة من البيانات الهائلة، وهو يستخدم استعلامات شبيهة بالأداة (SQL) لتسهيل تحليل البيانات، ويدعم التكامل مع خدمات (Google Cloud).

أبحث

أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن أدوات أُخرى لتحليل البيانات لم يَرِدْ ذكرها في الدرس، ثمَّ أكتب تقريرًا عنها، أعرض فيه لمزاياها الرئيسة، وأُضمِّنه الفَرْق بين كل أداة من هذه الأدوات وبرمجية إكسل (Excel) وبرمجية بايثون (Python) من حيث الاستخدام والإمكانيات، ثمَّ أقرأه أمام الزملاء/ الزميلات في الصف.

المفاضلة بين أدوات تحليل البيانات:

ليس من السهل اختيار الأداة المناسبة لتحليل البيانات بسبب تنوُّع الأدوات، وعدم وجود أداة تُناسِب جميع الاحتياجات والمُتطلَّبات. لذلك يجب أوَّلًا فهم طبيعة البيانات التي يُراد تحليلها، ثمَّ استخدام مجموعة من المعايير للمفاضلة بين الأدوات.

في ما يأتي بيان لبعض العوامل الرئيسة التي يجب مراعاتها وأخذها بالاعتبار عند اختيار الأداة:

1. **سعر الأداة:** تتباين أدوات تحليل البيانات في ما بينها من حيث تكلفة الاستخدام؛ فبعض هذه الأدوات يتطلَّب استخدامها اشتراكًا لقاء مبلغ من المال، وبعضها الآخر متاح للاستخدام بصورة مجّانية. غير أنَّ بعض الإصدارات المجّانية تفتقر إلى مزايا لا تتوافر إلّا في النسخ المدفوعة الثمن. ولهذا يجب أوَّلًا تقييم المزايا المُتوافِرة في الأدوات، وتحليل مدى وفائها بالاحتياجات المطلوبة.
2. **سهولة التعلُّم:** تتفرَّد كل أداة بقواعد وتعليمات خاصة بها. ولهذا تشهد الأدوات التي تمتاز بسهولة تعلُّمها إقبالًا أكثرَ من المُستخدِمين، وبخاصة المُبتدِئون منهم.
3. **الفاعلية في إدارة البيانات:** تتباين أدوات تحليل البيانات في ما بينها من حيث القدرة على إدارة البيانات بكفاءة واقتدار، بما في ذلك مُدَّة الاحتفاظ بالبيانات، وكيفية تخزينها، ومشاركتها، وأرشفتها، واسترجاعها، وكذا التخلُّص منها عند الحاجة، بما يَتَّسِق مع مُتطلَّبات التحليل والاستخدام.
4. **القدرات الرسومية:** تتيح الرسوم البيانية عرض البيانات بشكل واضح وبسيط؛ ما يُسهِّل فهم الأنماط والعلاقات. ولهذا تمتاز بعض الأدوات بقدرات رسومية مُتقدِّمة ومُتنوِّعة؛ ما يدعم التحليل البصري على نحوٍ أكثر شمولًا وفاعليةً.

المواطنة الرقمية:

- **الحقوق الرقميـــة** (Digital Rights): أُراعي حقوق الأفراد الرقمية عند جمع البيانات وتحليلها، مثل حَقِّهم في معرفة البيانات التي جُمِعت عنهم، وحَقِّهم في طلب حذفها. كذلك ألتزم بجمع البيانات بما يَتَّفِق مع القوانين الدولية والمحلية.
- **الأخلاقيات الرقميـــة** (Digital Ethics): أحرص على أخذ موافقة واضحة وصريحة من الأفراد قبل جمع بياناتهم، وأتأكَّد أنَّهم يفهمون الغرض من جمع البيانات واستخدامها.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُعرِّف المصطلحات الآتية:

1. القِيَم غير المنتمية (Outlier).
2. خرائط الأشجار (Treemap).
3. الإحصاءات الوصفية (Descriptive Statistics).
4. استكشاف البيانات (Data Exploration).
5. تصوُّر البيانات (Data visualization).

السؤال الثاني: أختار رمز الإجابة الصحيحة في كلٍّ ممّا يأتي:

1. **أحد الخيارات الآتية لا يُعَدُّ من طرائق جمع البيانات:**
   أ. المقابلات.
   ب. دراسة الحالة.
   ج. الرسم البياني.
   د. الملاحظة.

2. **تُستخدَم دراسة الحالة بصورة رئيسة في:**
   أ. جمع بيانات عن مجموعة كبيرة من الأفراد.
   ب. التحليل المُعمَّق لوحدة مُحدَّدة.
   ج. استكشاف الأنماط الزمنية.
   د. مراقبة الظواهر الطبيعية.

3. **أكثر أداة تُستخدَم في جمع بيانات من مجموعة كبيرة في وقت قصير هي:**
   أ. المقابلات.
   ب. الملاحظة.
   ج. الاستبانات.
   د. مجموعات التركيز.

4. **الأداة النوعية التي تُركِّز على مناقشة مجموعة صغيرة من الأفراد لاستكشاف آرائهم هي:**
   أ. المسوحات.
   ب. مجموعات التركيز.
   ج. دراسة الحالة.
   د. المقابلات الهاتفية.

5. **من مكتبات برمجية بايثون (Python) التي تُستخدَم في إنشاء رسوم بيانية ومُخطَّطات مُتقدِّمة:**
   أ. مكتبة (Pandas).
   ب. مكتبة (NumPy).
   ج. مكتبة (Matplotlib).
   د. مكتبة (Scikit-learn).

السؤال الثالث: أُميِّز العبارات الصحيحة من العبارات غير الصحيحة في ما يأتي:

1. تُعَدُّ الملاحظة واحدة من أدوات جمع البيانات التي تعتمد على تدخُّل الباحث مباشرة لتوجيه السلوك.
2. تُعَدُّ الاستبانات أداة رئيسة لجمع بيانات كَمِّية مُنظَّمة وقابلة للتحليل الإحصائي.
3. تُعَدُّ المقابلات الرسمية أكثر مرونة مقارنةً بالمقابلات غير الرسمية.
4. تُستخدَم مكتبة (Pandas) في معالجة البيانات والتعامل مع الجداول في برمجية بايثون (Python).
5. تُستخدَم المُخطَّطات العمودية في عرض مقارنات بين الفئات.
6. تُعَدُّ لغة البرمجة (Python) الخيار الأفضل لتحليل البيانات غير المُنظَّمة، في حين تُعَدُّ لغة (SQL) الخيار الأفضل لتحليل البيانات المُنظَّمة.

السؤال الرابع: أملأ الفراغ بما هو مناسب في كلٍّ من الجمل الآتية:

1. تُعَدُّ _____ وسيلة فاعلة لجمع بيانات نوعية تفصيلية تتطلَّب فهمًا عميقًا لموضوع مُعيَّن.
2. تُجمَع البيانات باستخدام _____ لتحليل الظواهر بصورة مباشرة من دون تدخُّل الباحث.
3. يُمكِن جمع بيانات شاملة ومُنظَّمة باستخدام _____؛ سواء أكان ذلك ورقيًّا أم إلكترونيًّا.
4. تُستخدَم _____ عند جمع آراء مُتعدِّدة - في الوقت نفسه- عن موضوع مُحدَّد.
5. مكتبة _____ في برمجية بايثون (Python) تُستخدَم في إنشاء الرسوم البيانية والمُخطَّطات.

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أبحث في المواقع الإلكترونية الموثوقة في شبكة الإنترنت عن مَزِيَّة كل مكتبة من المكتبات التي تُوفِّرها برمجية بايثون الواردة في الجدول الآتي.

اسم المكتبة (Library Name)	المَزِيَّة
Pandas	
NumPy	
Scikit-learn	
SciPy	
Matplotlib	
Natural Language Toolkit (NLTK)	

السؤال الثاني: أُقارِن بين لغة البرمجة بايثون (Python) وبرمجية إكسل (Excel) من حيث الفاعلية في تحليل البيانات الكبيرة والرسوم البيانية المُتقدِّمة.

السؤال الثالث: أدرس الحالة الآتية، ثمَّ أُبيِّن كيف يُمكِن للشركة الاستفادة من مراحل استكشاف البيانات في فهم أسباب التباين في أداء الموظفين، والعمل على تحسين أدائهم بصورة ملموسة:
"تعمل إحدى الشركات على استكشاف البيانات المُتعلِّقة بأداء موظفيها لتحسين الإنتاجية. وقد تبيَّن لها أثناء عملية استكشاف البيانات وجود تفاوت كبير في أداء الموظفين العاملين في الفروع المختلفة للشركة".

الدرس الثالث

جمع البيانات وتحليلها باستخدام برمجية بايثون
(Python Data Collection & Analyses Methods)

الفكرة الرئيسة:

سأتعلَّم في هذا الدرس كيف أستخدم مكتبات برمجية بايثون (Python) في جمع البيانات ضمن مجال مُعيَّن، واستكشافها وتحليلها باستخدام الأدوات المناسبة. كذلك سأتعلَّم كيفية تحديد الأنماط والاتِّجاهات ووصفها عن طريق التمثيلات البيانية التي تُوفِّرها برمجية بايثون (Python)، ثمَّ توظيف هذه الأنماط في استخلاص استنتاجات دقيقة قائمة على البيانات.

المفاهيم والمصطلحات:

الأمر (Pip)، الإحصاءات الوصفية (Descriptive Statistics).

نتاجات التعلُّم (Learning Outcomes):

- أُثبِّت المكتبات الخارجية الخاصة بتحليل البيانات في برمجية بايثون، مثل:(Panadas)، و (Matplotlib).
- أستورد ملفات البيانات (مثل: (CSV)، و(Excel)) إلى برمجية بايثون (Python) لتحليلها.
- أُجري عمليات تحليلية على البيانات باستخدام مكتبة (Pandas) في برمجية بايثون (Python).
- أُفسِّر نتائج التحليل الإحصائي الوصفي من شاشة مُخرَجات برمجية بايثون (Python).
- أعرض البيانات بطرائق مختلفة باستخدام مكتبة (Matplotlib) في برمجية بايثون (Python).
- أُحدِّد الأنماط في الرسوم البيانية في برمجية بايثون (Python) باستخدام مزايا بصرية.
- أُحلِّل الرسوم البيانية لاستكشاف الأنماط واستخلاص استنتاجات تدعم الفهم الأوَّلي للبيانات.

تُعَدُّ عملية جمع البيانات خطوة مُهِمَّة وركيزة أساسية في علم البيانات؛ إذ تُمثِّل المرحلة الأولى من مراحل اتِّخاذ القرارات المُستنِدة إلى البيانات. تتبع هذه المرحلة عملية تنظيم البيانات لتحويلها إلى شكل مُنظَّم ومُهيكَل يتيح تحليلها بصورة مناسبة؛ ما يُسهِم في استنباط النتائج والاستنتاجات الداعمة لاتِّخاذ قرارات دقيقة وفاعلة.

نشاط تمهيدي

أتأمَّل المصطلحات الآتية، ثمَّ أُفكِّر في ما يربط بينها، وما يربطها بموضوع جمع البيانات وتحليلها:

الصفوف والأعمدة، الخريطة المفاهيمية، الاستبانة.

أُناقِش أفراد مجموعتي في ذلك، ثمَّ نُدوِّن معًا ما نتوصَّل إليه من نتائج.

جمع البيانات وتحليلها:

تعرَّفْتُ في الدرس السابق وجود طرائق عِدَّة لجمع البيانات، يُمكِن تفريغها وتنظيمها باستخدام أدوات مُتعدِّدة. تُعَدُّ الجداول إحدى أكثر الأدوات شيوعًا لتفريغ البيانات، ويُستخدَم لذلك برمجيات مختلفة، مثل برمجية إكسل (Excel) التي تعمل على ترتيب البيانات بشكل مُنظَّم يُسهِّل التعامل معها وتحليلها.

أُلاحِظ الشكل (3–1) الذي يتعلَّق ببيانات جُمِعت عن أجهزة أبل آيفون (Apple iPhones) التي تُعَدُّ من أكثر الهواتف الذكية مبيعًا في العالَم، ثمَّ فُرِّغت هذه البيانات في ملف إكسل امتداده (CSV).

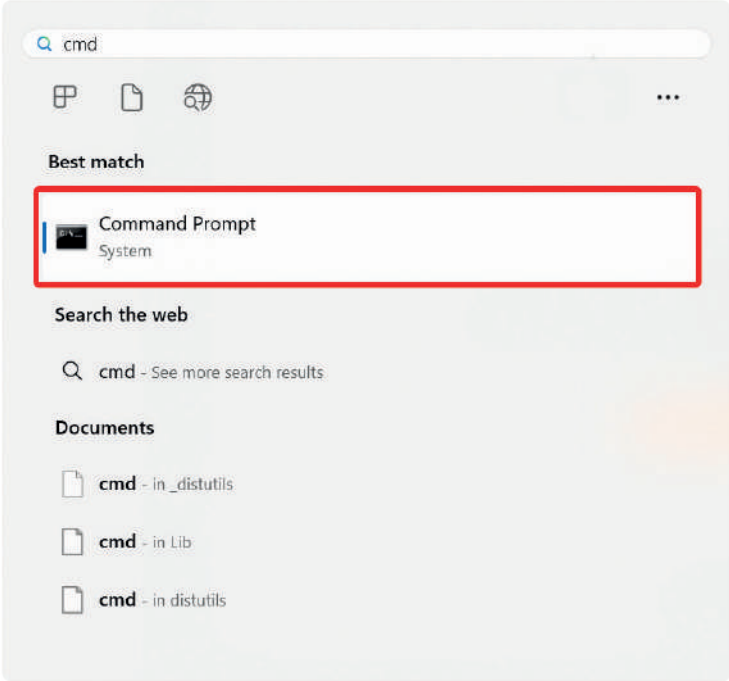
	A	B	C	D	E	F	G	H	I	J	K
1	Product Name	Product URL	Brand	Sale Price	Mrp	Discount Percentage	Number Of Ratings	Number Of Reviews	Upc	Star Rating	Ram
2	APPLE iPho	https://ww	Apple	49900	49900	0	3431	356	MOBEXRGV	4.6	2 GB
3	APPLE iPho	https://ww	Apple	84900	84900	0	3431	356	MOBEXRGV	4.6	2 GB
4	APPLE iPho	https://ww	Apple	84900	84900	0	3431	356	MOBEXRGV	4.6	2 GB
5	APPLE iPho	https://ww	Apple	77000	77000	0	11202	794	MOBEXRGV	4.5	2 GB
6	APPLE iPho	https://ww	Apple	77000	77000	0	11202	794	MOBEXRGV	4.5	2 GB
7	APPLE iPho	https://ww	Apple	49900	49900	0	3431	356	MOBEXRGV	4.6	2 GB
8	APPLE iPho	https://ww	Apple	49900	49900	0	3431	356	MOBEXRGV	4.6	2 GB
9	APPLE iPho	https://ww	Apple	77000	77000	0	11202	794	MOBEXRGV	4.5	2 GB
10	APPLE iPho	https://ww	Apple	89900	89900	0	1454	149	MOBF944E	4.6	4 GB
11	Apple iPhor	https://ww	Apple	41999	52900	20	79512	6796	MOBF9Z7Z	4.6	4 GB
12	Apple iPhor	https://ww	Apple	39999	47900	16	79512	6796	MOBF9Z7Z	4.6	4 GB
13	Apple iPhor	https://ww	Apple	41999	52900	20	79582	6804	MOBF9Z7Z	4.6	4 GB
14	Apple iPhor	https://ww	Apple	41999	52900	20	79512	6796	MOBF9Z7Z	4.6	3 GB
15	Apple iPhor	https://ww	Apple	41999	52900	20	79512	6796	MOBF9Z7Z	4.6	4 GB
16	APPLE iPho	https://ww	Apple	1E+05	131900	0	1078	101	MOBFKCTS	4.7	4 GB
17	APPLE iPho	https://ww	Apple	1E+05	117100	0	1078	101	MOBFKCTS	4.7	4 GB
18	APPLE iPho	https://ww	Apple	1E+05	131900	0	1078	101	MOBFKCTS	4.7	4 GB
19	APPLE iPho	https://ww	Apple	1E+05	117100	0	1078	101	MOBFKCTS	4.7	4 GB

الشكل (3–1): بيانات عن أجهزة أبل آيفون (Apple iPhones).

تُعَدُّ برمجية بايثون واحدة من أبرز الأدوات المُستخدَمة في معالجة البيانات وتحليلها؛ إذ تُوفِّر العديد من المكتبات الجاهزة التي تُستخدَم في تحليل البيانات وعرضها. وبالرغم من ذلك، فإنَّ هذه المكتبات ليست مُثبَّتة بشكل افتراضي مع الحُزْمة الأساسية للغة البرمجة بايثون، وإنَّما تُعَدُّ مكتبات خارجية يتعيَّن تثبيتها بصورة مُنفصِلة. أمّا أهمُّ المكتبات المُستخدَمة في تحليل البيانات وعرضها فهي: مكتبة (Pandas)، ومكتبة (NumPy)، ومكتبة (Matplotlib).

تثبيت المكتبات الخارجية في برمجية بايثون:

يُمكِنني تثبيت المكتبات الخارجية في برمجية بايثون كما تعلَّمْتُ في الوحدة الأولى . أنظر الشكل (3-2) الذي يُبيِّن شاشة الأوامر (Command Prompt) الموجودة في نظام التشغيل.



الشكل (3-2): شاشة الأوامر (Command Prompt).

مثال:

يُمكِن تثبيت وحدة (pandas) في جهاز الحاسوب بفتح شاشة الأوامر (Command Prompt)، ثمَّ كتابة الأمر الآتي:

```
C:\>pip install pandas
```

ثمَّ الضغط على مفتاح الإدخال (Enter)، والانتظار إلى حين الانتهاء من تثبيت هذه المكتبة في جهاز الحاسوب.

نشاط جماعي

- أعمل – بالتعاون مع أفراد مجموعتي– على تثبيـــت مكتبة (Pandas) باتِّباع الخطوات المذكورة آنفًا، ثمَّ نُجرِّب معًا تثبيت مكتبة (Matplotlib)، ومكتبة (NumPy).

استدعاء ملف بيانات في برمجية بايثون:

يُمكِن تحليل ملف بيانات باستخدام برمجية بايثون بعد استدعاء البيانات إلى مُفسِّر لغة البرمجة بايثون. وقبل ذلك يتعيَّن استدعاء المكتبات اللازمة لمعالجة البيانات وتحليلها.

خطوات استدعاء ملف بيانات في برمجية بايثون:

1. **استدعاء مكتبة (Pandas):** تُستخدَم مكتبة (Pandas) في تحليل البيانات بعد استدعائها باستخدام الأمر:

```
import pandas
```

2. **قراءة الملف باستخدام الدالَّة read_csv:** يُمكِن قراءة ملف البيانات باستخدام الدالَّة read_csv () **من مكتبة (Pandas)، وذلك بكتابة الجملة العامة الآتية:**

```
df = pandas.read_csv("C:\\Users\\user\\OneDrive\\Desktop\\apple_products.csv")
```

حيث:

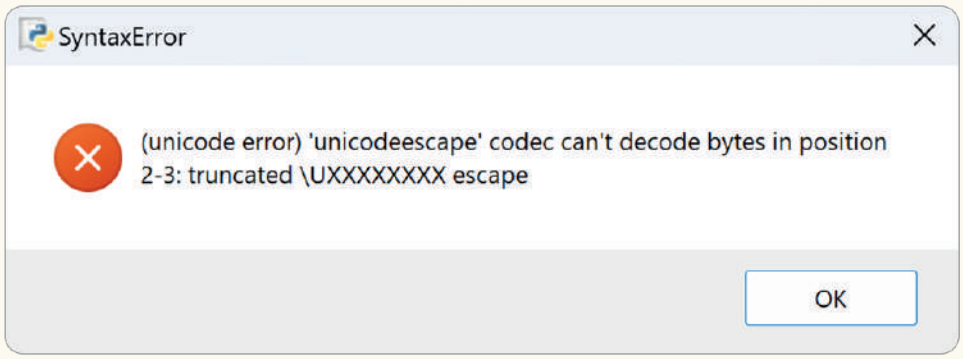
df: اسم المُتغيِّر الذي ستُخزَّن فيه البيانات المستدعاة.

pandas.readv_csv(): دالَّة قراءة الملف الذي امتداده (CSV).

المسار: المسار الكامل للملف الذي يشتمل على مشروعين، ويُكتَب بين قوسين مزدوجين، مع التحقُّق من استخدام رمز (Backslash) مزدوج.

إضاءة

إذا طُبع مسار الملف باستخدام إشارة (backslash) واحدة، فإنَّ رسالةً تُشعِر بوجود خطأ ستظهر عند تنفيذ البرنامج كما في الشكل (3-3).



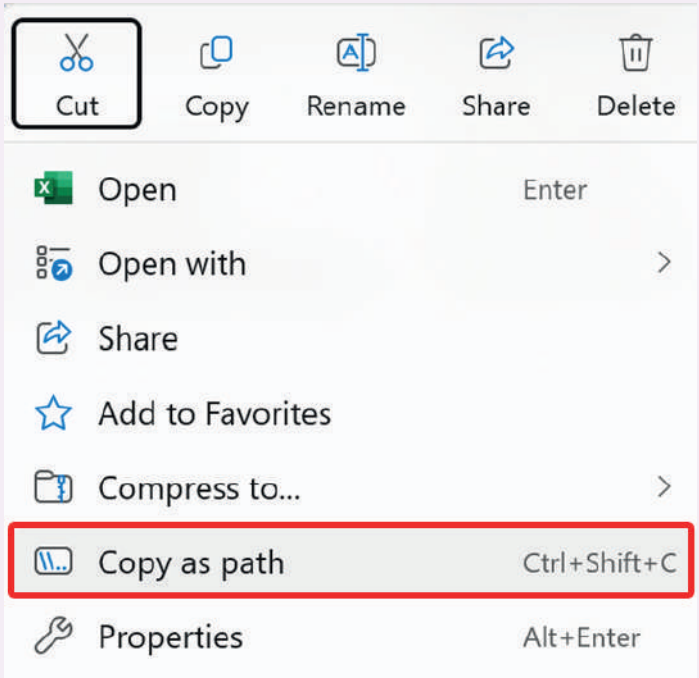
الشكل (3-3): رسالة تفيد بحدوث خطأ في برمجية بايثون.

إثراء

يُمكِن إظهار مسار الملف كاملًا باتِّباع الخطوتين الآتيتين:

أ- التوجُّه إلى موقع تخزين الملف في جهاز الحاسوب.

ب- الضغط بزِرِّ الفأرة الأيمن على الملف، فتظهر القائمة الفرعية كما في الشكل (3-4)، ثمَّ اختيار الخيار (Copy as path) منها، ثمَّ عمل لصق على مُفسِّر لغة البرمجة بايثون، فيظهر مسار الملف كاملًا.



الشكل (3-4): قائمة الخيارات التي تظهر عند الضغط على الزِّرِّ الأيمن للملف.

3. **طباعة البيانات:** تُطبَع جملة (طباعة البيانات) بعد قراءتها من الملف، وظهورها على شاشة المُخرَجات. أنظر الشكل (3-5) الذي يُبيِّن تمثيل المقطع البرمجي كاملًا.

```
import pandas
df = pandas.read_csv("C:\\Users\\user\\OneDrive\\Desktop\\apple_products.csv")
print(df)
```

الشكل (3-5): مقطع برمجي لاستدعاء ملف بيانات وطباعته.

بعد تنفيذ المقطع البرمجي سيكون ناتج التنفيذ كما هو مُبيَّن في الشكل (3-6).

```
                                Product Name  ...  Ram
0        APPLE iPhone 8 Plus (Gold, 64 GB)  ...  2 GB
1   APPLE iPhone 8 Plus (Space Grey, 256 GB)  ...  2 GB
2      APPLE iPhone 8 Plus (Silver, 256 GB)  ...  2 GB
3         APPLE iPhone 8 (Silver, 256 GB)  ...  2 GB
4           APPLE iPhone 8 (Gold, 256 GB)  ...  2 GB
..                                    ...  ...  ...
57        APPLE iPhone SE (Black, 64 GB)  ...  4 GB
58      APPLE iPhone 11 (Purple, 64 GB)  ...  4 GB
59       APPLE iPhone 11 (White, 64 GB)  ...  4 GB
60       APPLE iPhone 11 (Black, 64 GB)  ...  4 GB
61         APPLE iPhone 11 (Red, 64 GB)  ...  4 GB

[62 rows x 11 columns]
```

الشكل (3-6): ناتج تنفيذ المقطع البرمجي لاستدعاء ملف وطباعته.

عند استخدام مكتبة (Pandas) في قراءة ملف البيانات (CSV)، وطباعة المحتوى باستخدام الأمر print(df)، يُسارع مُفسِّر لغة البرمجة بايثون -افتراضيًّا- إلى عرض أوَّل خمسة أسطر وآخر خمسة أسطر فقط من الملف، وطباعة أوَّل عمودين وآخر عمودين إذا كان عدد الأعمدة كثيرًا. وهذا السلوك الافتراضي يُسهِّل معاينة البيانات من دون عرض الجدول كاملًا، وبخاصة إذا كان يحتوي على عدد كبير من الصفوف أو الأعمدة. وفي نهاية شاشة المُخرَجات، يُطبَع إجمالي عدد الصفوف والأعمدة في الجدول الأصلي.

نشاط جماعي

أدرس المقطع البرمجي الآتي، ثمَّ أُجيب - بالتعاون مع أفراد مجموعتي- عن الأسئلة التي تليه:

```
import pandas
data = pandas.read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')
print(data)
```

- ما اسم الملف الذي تمَّ استدعاؤه؟
- ما الدالَّة المسؤولة عن استدعاء الملف؟
- ما اسم المكتبة التي تمَّ استيرادها؟

نشاط فردي

- أفتح مُفسِّر لغة البرمجة بايثون، ثمَّ أعمل على استيراد مكتبة (pandas)، وأتأكَّد أنَّني ثبَّتها في جهاز الحاسوب بصورة صحيحة.
- أستخدم الدالَّة read_csv من مكتبة بايثون لقراءة الملف.
- أطبع البيانات على شاشة المُخرَجات، ثمَّ أُلاحِظ النتيجة لقراءة الملف.

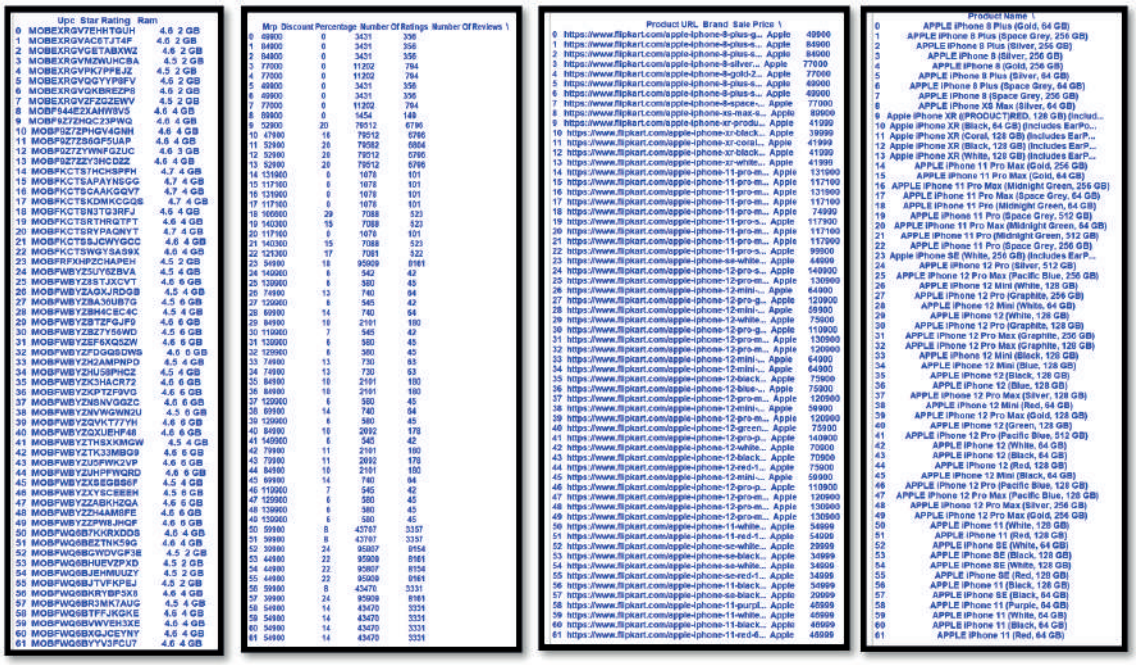
عرض ملف البيانات كاملًا:

يُمكِن عرض جميع الأسطر والأعمدة للبيانات التي قُرِئت باستخدام مكتبة (pandas)، لكنَّ ذلك يتطلَّب أوَّلًا تغيير الإعدادات الافتراضية باستخدام الدالَّة set_option كما هو مُوضَّح في المقطع البرمجي الذي يُمثِّله الشكل (3-7).

```
import pandas
pandas.set_option('display.max_rows', None)
pandas.set_option('display.max_columns', None)
df=pandas.read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')
print(df)
```

الشكل (3-7): مقطع برمجي لتغيير الإعدادات الافتراضية لمكتبة (pandas).

ما إنْ يتمُّ تشغيل المقطع البرمجي، حتّى تظهر جميع الأسطر والأعمدة من ملف testdoc.csv على شاشة النتائج؛ ما يُسهِّل قراءة جميع البيانات وتحليلها. أنظر الشكل (3-8).



	Upc	Star Rating	Ram
0	MOBEXRGV7EHHTGUH	4.6	2 GB
1	MOBEXRGVAC6TJT4F	4.6	2 GB
2	MOBEXRGVGETABXWZ	4.6	2 GB
3	MOBEXRGVMZWUHCBA	4.5	2 GB
4	MOBEXRGVPK7PFEJZ	4.5	2 GB
5	MOBEXRGVQQTYPBFV	4.6	2 GB
6	MOBEXRGVGKBREZP8	4.6	2 GB
7	MOBEXRGV2FZGZEWV	4.5	2 GB
8	MOBF544EZEAHWWVS	4.6	4 GB
9	MOBF9ZTZHQC23PWQ	4.6	4 GB
10	MOBF9ZTZPHGV4GNH	4.6	4 GB
11	MOBF9ZTZ8GGF5UAP	4.6	4 GB
12	MOBF9ZTZYWNFGZUC	4.6	3 GB
13	MOBF9ZTZZY3HCDZZ	4.6	4 GB
14	MOBFKCTS7HCHSPPH	4.7	4 GB
15	MOBFKCTSAPAYNEGG	4.7	4 GB
16	MOBFKCTSCAAKGQV7	4.7	4 GB
17	MOBFKCTSKDMKCCQS	4.7	4 GB
18	MOBFKCTSN3TG3RFJ	4.6	4 GB
19	MOBFKCTSRTHRQTFT	4.6	4 GB
20	MOBFKCTSRYPAQNYT	4.7	4 GB
21	MOBFKCTSSJCWYGCC	4.6	4 GB
22	MOBFKCTSWGY5S9XC	4.6	4 GB
23	MOBFRFXHPZCHAPEH	4.5	2 GB
24	MOBFWBYZ5UY6ZBVA	4.5	4 GB
25	MOBFWBYZ8STJXCVT	4.6	6 GB
26	MOBFWBYZAGXJRDGB	4.5	4 GB
27	MOBFWBYZBA36UB7G	4.5	6 GB
28	MOBFWBYZBMHCEC4C	4.5	4 GB
29	MOBFWBYZBTZFGJP9	4.6	6 GB
30	MOBFWBYZBZ7Y56WD	4.5	6 GB
31	MOBFWBYZEF6XQSZW	4.6	6 GB
32	MOBFWBYZFDGQ6DWS	4.6	6 GB
33	MOBFWBYZKAMPNPD	4.5	4 GB
34	MOBFWBYZHU58PHCZ	4.5	4 GB
35	MOBFWBYZK5HACR72	4.6	6 GB
36	MOBFWBYZKPTZFWVG	4.6	6 GB
37	MOBFWBYZN8NVGGZC	4.6	6 GB
38	MOBFWBYZNVWGWN2U	4.5	6 GB
39	MOBFWBYZQVKT7YYH	4.6	6 GB
40	MOBFWBYZQXUEHP48	4.6	6 GB
41	MOBFWBYZTHSXKMGW	4.5	4 GB
42	MOBFWBYZTK33MBG9	4.6	6 GB
43	MOBFWBYZUSFWKZVP	4.6	6 GB
44	MOBFWBYZUHPFYGRD	4.6	6 GB
45	MOBFWBYZX6EGBS6F	4.5	4 GB
46	MOBFWBYZXYSCEEEH	4.5	6 GB
47	MOBFWBYZZABKHZQA	4.6	6 GB
48	MOBFWBYZZH4AM8FE	4.6	6 GB
49	MOBFWBYZZPWBJHQF	4.6	6 GB
50	MOBFWQ6B7KXRXDDB	4.6	4 GB
51	MOBFWQ6BEZTNK59G	4.6	4 GB
52	MOBFWQ6BGWDVGF3B	4.5	2 GB
53	MOBFWQ6BHUEVZPXD	4.5	2 GB
54	MOBFWQ6BJEHWUUZY	4.5	2 GB
55	MOBFWQ6BJTVFKPEJ	4.5	2 GB
56	MOBFWQ6BKRYBPSX8	4.6	4 GB
57	MOBFWQ6BR3NK7AUG	4.5	4 GB
58	MOBFWQ6BTFFJKGKE	4.6	4 GB
59	MOBFWQ6BVWVEH3XE	4.6	4 GB
60	MOBFWQ6BXGJCEYNY	4.6	4 GB
61	MOBFWQ6BYYVJFCU7	4.6	4 GB

	Mrp	Discount	Percentage	Number Of Ratings	Number Of Reviews \
0	49900	0	3431	356	
1	84900	0	3431	356	
2	84900	0	3431	356	
3	77000	0	11202	794	
4	77000	0	11202	794	
5	49900	0	3431	356	
6	49900	0	3431	356	
7	77000	0	11202	794	
8	59900	0	1454	149	
9	52900	20	79512	6796	
10	47900	16	79512	6796	
11	52900	20	79582	6804	
12	52900	20	79512	6796	
13	52900	20	79512	6796	
14	131900	0	1678	101	
15	117100	0	1678	101	
16	131900	0	1678	101	
17	117100	0	1678	101	
18	90600	29	7088	523	
19	140300	15	7088	523	
20	117100	0	1678	101	
21	140300	15	7088	523	
22	121300	17	7081	522	
23	94900	18	95909	8161	
24	149000	6	542	42	
25	129900	6	580	45	
26	74900	13	740	64	
27	129900	6	545	42	
28	99900	14	740	64	
29	84900	10	2101	180	
30	119900	7	545	42	
31	129900	6	580	45	
32	129900	6	580	45	
33	74900	13	730	63	
34	74900	13	730	63	
35	84900	10	2101	180	
36	84900	10	2101	180	
37	129900	6	580	45	
38	89900	14	740	64	
39	129900	6	580	45	
40	84900	10	2092	178	
41	149900	6	545	42	
42	79900	11	2101	180	
43	79900	11	2092	178	
44	84900	10	2101	180	
45	99900	14	740	64	
46	110900	7	545	42	
47	129900	6	580	45	
48	139900	6	580	45	
49	139900	6	580	45	
50	89900	8	43767	3357	
51	89900	8	43767	3357	
52	99900	24	95907	8154	
53	44900	22	95909	8161	
54	44900	22	95907	8154	
55	44900	22	95909	8161	
56	89900	8	45476	3331	
57	99900	24	95909	8161	
58	54900	14	43470	3331	
59	54900	14	43470	3331	
60	54900	14	43470	3331	
61	54900	14	43470	3331	

	Product URL	Brand	Sale Price \
0	https://www.flipkart.com/apple-iphone-8-plus-g...	Apple	49900
1	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	84900
2	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	84900
3	https://www.flipkart.com/apple-iphone-8-silver...	Apple	77000
4	https://www.flipkart.com/apple-iphone-8-gold-2...	Apple	77000
5	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	49900
6	https://www.flipkart.com/apple-iphone-8-plus-s...	Apple	49900
7	https://www.flipkart.com/apple-iphone-8-space-...	Apple	77000
8	https://www.flipkart.com/apple-iphone-xs-max-s...	Apple	89900
9	https://www.flipkart.com/apple-iphone-xr-produ...	Apple	41999
10	https://www.flipkart.com/apple-iphone-xr-black...	Apple	39999
11	https://www.flipkart.com/apple-iphone-xr-coral...	Apple	41999
12	https://www.flipkart.com/apple-iphone-xr-black...	Apple	41999
13	https://www.flipkart.com/apple-iphone-xr-white...	Apple	41999
14	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	131900
15	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	117100
16	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	131900
17	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	117100
18	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	74999
19	https://www.flipkart.com/apple-iphone-11-pro-s...	Apple	117900
20	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	117100
21	https://www.flipkart.com/apple-iphone-11-pro-m...	Apple	117900
22	https://www.flipkart.com/apple-iphone-11-pro-s...	Apple	99900
23	https://www.flipkart.com/apple-iphone-se-white...	Apple	44999
24	https://www.flipkart.com/apple-iphone-12-pro-s...	Apple	149900
25	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	130900
26	https://www.flipkart.com/apple-iphone-12-mini-...	Apple	64900
27	https://www.flipkart.com/apple-iphone-12-pro-g...	Apple	129900
28	https://www.flipkart.com/apple-iphone-12-mini-...	Apple	59900
29	https://www.flipkart.com/apple-iphone-12-white...	Apple	75900
30	https://www.flipkart.com/apple-iphone-12-pro-g...	Apple	119900
31	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	130900
32	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	129900
33	https://www.flipkart.com/apple-iphone-12-mini-...	Apple	64900
34	https://www.flipkart.com/apple-iphone-12-mini-...	Apple	64900
35	https://www.flipkart.com/apple-iphone-12-black...	Apple	75900
36	https://www.flipkart.com/apple-iphone-12-blue-...	Apple	75900
37	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	120900
38	https://www.flipkart.com/apple-iphone-12-mini-...	Apple	59900
39	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	120900
40	https://www.flipkart.com/apple-iphone-12-green...	Apple	75900
41	https://www.flipkart.com/apple-iphone-12-pro-p...	Apple	149900
42	https://www.flipkart.com/apple-iphone-12-white...	Apple	70900
43	https://www.flipkart.com/apple-iphone-12-black...	Apple	70900
44	https://www.flipkart.com/apple-iphone-12-red-1...	Apple	75900
45	https://www.flipkart.com/apple-iphone-12-mini-...	Apple	59900
46	https://www.flipkart.com/apple-iphone-12-pro-p...	Apple	119900
47	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	120900
48	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	130900
49	https://www.flipkart.com/apple-iphone-12-pro-m...	Apple	129900
50	https://www.flipkart.com/apple-iphone-11-white...	Apple	54999
51	https://www.flipkart.com/apple-iphone-11-red-1...	Apple	54999
52	https://www.flipkart.com/apple-se-white-...	Apple	39999
53	https://www.flipkart.com/apple-se-black...	Apple	34999
54	https://www.flipkart.com/apple-se-white...	Apple	34999
55	https://www.flipkart.com/apple-se-red-1...	Apple	34999
56	https://www.flipkart.com/apple-11-black...	Apple	54999
57	https://www.flipkart.com/apple-se-black...	Apple	29999
58	https://www.flipkart.com/apple-11-purpl...	Apple	49999
59	https://www.flipkart.com/apple-11-white...	Apple	49999
60	https://www.flipkart.com/apple-11-black...	Apple	49999
61	https://www.flipkart.com/apple-11-red-6...	Apple	49999

	Product Name \
0	APPLE iPhone 8 Plus (Gold, 64 GB)
1	APPLE iPhone 8 Plus (Space Grey, 256 GB)
2	APPLE iPhone 8 Plus (Silver, 256 GB)
3	APPLE iPhone 8 (Silver, 256 GB)
4	APPLE iPhone 8 (Gold, 256 GB)
5	APPLE iPhone 8 Plus (Silver, 64 GB)
6	APPLE iPhone 8 Plus (Space Grey, 64 GB)
7	APPLE iPhone 8 (Space Grey, 256 GB)
8	APPLE iPhone XS Max (Silver, 64 GB)
9	Apple iPhone XR ((PRODUCT)RED, 128 GB) (Includ...
10	Apple iPhone XR (Black, 64 GB) (Includes EarPo...
11	Apple iPhone XR (Coral, 128 GB) (Includes EarP...
12	Apple iPhone XR (Black, 128 GB) (Includes EarP...
13	Apple iPhone XR (White, 128 GB) (Includes EarP...
14	APPLE iPhone 11 Pro Max (Gold, 256 GB)
15	APPLE iPhone 11 Pro Max (Gold, 64 GB)
16	APPLE iPhone 11 Pro Max (Midnight Green, 256 GB)
17	APPLE iPhone 11 Pro Max (Space Grey, 64 GB)
18	APPLE iPhone 11 Pro (Midnight Green, 64 GB)
19	APPLE iPhone 11 Pro (Space Grey, 512 GB)
20	APPLE iPhone 11 Pro Max (Midnight Green, 64 GB)
21	APPLE iPhone 11 Pro (Midnight Green, 512 GB)
22	APPLE iPhone 11 Pro (Space Grey, 256 GB)
23	Apple iPhone SE (White, 256 GB) (Includes EarP...
24	APPLE iPhone 12 Pro (Silver, 512 GB)
25	APPLE iPhone 12 Pro Max (Pacific Blue, 256 GB)
26	APPLE iPhone 12 Mini (White, 128 GB)
27	APPLE iPhone 12 Pro (Graphite, 256 GB)
28	APPLE iPhone 12 Mini (White, 64 GB)
29	APPLE iPhone 12 (White, 128 GB)
30	APPLE iPhone 12 Pro (Graphite, 128 GB)
31	APPLE iPhone 12 Pro Max (Graphite, 256 GB)
32	APPLE iPhone 12 Pro Max (Graphite, 128 GB)
33	APPLE iPhone 12 Mini (Black, 128 GB)
34	APPLE iPhone 12 Mini (Blue, 128 GB)
35	APPLE iPhone 12 (Black, 128 GB)
36	APPLE iPhone 12 (Blue, 128 GB)
37	APPLE iPhone 12 Pro Max (Silver, 128 GB)
38	APPLE iPhone 12 Mini (Red, 64 GB)
39	APPLE iPhone 12 Pro Max (Gold, 128 GB)
40	APPLE iPhone 12 (Green, 128 GB)
41	APPLE iPhone 12 Pro (Pacific Blue, 512 GB)
42	APPLE iPhone 12 (White, 64 GB)
43	APPLE iPhone 12 (Black, 64 GB)
44	APPLE iPhone 12 (Red, 128 GB)
45	APPLE iPhone 12 Mini (Black, 64 GB)
46	APPLE iPhone 12 Pro (Pacific Blue, 128 GB)
47	APPLE iPhone 12 Pro Max (Pacific Blue, 128 GB)
48	APPLE iPhone 12 Pro Max (Silver, 256 GB)
49	APPLE iPhone 12 Pro Max (Gold, 256 GB)
50	APPLE iPhone 11 (White, 128 GB)
51	APPLE iPhone 11 (Red, 128 GB)
52	APPLE iPhone SE (White, 64 GB)
53	APPLE iPhone SE (Black, 128 GB)
54	APPLE iPhone SE (White, 128 GB)
55	APPLE iPhone SE (Red, 128 GB)
56	APPLE iPhone 11 (Black, 128 GB)
57	APPLE iPhone SE (Black, 64 GB)
58	APPLE iPhone 11 (Purple, 64 GB)
59	APPLE iPhone 11 (White, 64 GB)
60	APPLE iPhone 11 (Black, 64 GB)
61	APPLE iPhone 11 (Red, 64 GB)

الشكل (3-8): ناتج تنفيذ المقطع البرمجي بعد تغيير الإعدادات.

إضاءة

تُعَدُّ الدالَّة **()info** في مكتبة (Pandas) أداة أساسية للحصول على مُلخَّص شامل عن البيانات، وهي تساعد على تحليل البيانات تحليلًا دقيقًا عن طريق عرض أسماء الأعمدة الموجودة في الجدول، وأنواع البيانات لكل عمود، وعدد القِيَم غير المفقودة في كل عمود؛ ما يُسهِّل تعرُّف الأعمدة التي تحتوي على بيانات مفقودة أو أنواع مختلفة من البيانات.

تحليل الإحصاءات الوصفية (Descriptive Statistics) في برمجية بايثون:

تُستخدَم الإحصاءات الوصفية في برمجية بايثون لتلخيص خصائص مجموعة البيانات ووصفها؛ ما يساعد على فهم البيانات وتحليلها بفاعلية. تُصنَّف الإحصاءات الوصفية في برمجية بايثون إلى ثلاث فئات رئيسة، هي:

1. **مقاييس النزعة المركزية:** تُعبِّر هذه الفئة عن النقطة التي تتركَّز حولها القِيَم، وهي تضمُّ الوسط الحسابي، والوسيط، والمنوال.
2. **مقاييس التشتُّت:** تُعبِّر هذه الفئة عن مدى انتشار القِيَم حول المُتوسِّط، وهي تضمُّ الانحراف المعياري، والتباين.

3. **التكرارات:** يستفاد من التكرارات في قياس عدد مَرّات تكرار كل قيمة في مجموعة البيانات؛ ما يساعد على تحديد أكثر القِيَم شيوعًا.

يُمكِن تعريض البيانات لتحليل الإحصاءات الوصفية باستخدام الدالَّة describe() في جملة الطباعة على النحو الآتي:

```
print (df.describe ( ) )
```

حيث:

df.: اسم المُتغيِّر الذي خُزِّنت فيه البيانات.

`describe ( )`: الدالَّة التي تعمل على حساب الإحصاءات الوصفية للبيانات.

فيكون المقطع البرمجي الناتج كما في الشكل (3-9).

```
import pandas
df = pandas.read_csv('C:\\Users\\user\\OneDrive\\Desktop\\apple_products.csv')
print(df.describe())
```

الشكل (3-9): المقطع البرمجي للدالَّة describe.

بعد الانتهاء من التنفيذ، تظهر الإحصاءات الوصفية الآتية:

```
       Sale Price        Mrp  Discount Percentage  Number Of Ratings  \
count    62.000000    62.000000            62.000000          62.000000
mean  80073.887097  88058.064516             9.951613       22420.403226
std   34310.446132  34728.825597             7.608079       33768.589550
min   29999.000000  39900.000000             0.000000         542.000000
25%   49900.000000  54900.000000             6.000000         740.000000
50%   75900.000000  79900.000000            10.000000        2101.000000
75%  117100.000000 120950.000000            14.000000       43470.000000
max  140900.000000 149900.000000            29.000000       95909.000000

       Number Of Reviews  Star Rating
count          62.000000    62.000000
mean         1861.677419     4.575806
std          2855.883830     0.059190
min            42.000000     4.500000
25%            64.000000     4.500000
50%           180.000000     4.600000
75%          3331.000000     4.600000
max          8161.000000     4.700000
```

وعند تنفيذ الكود الخاص بالدالَّة `describe ( )`، تظهر النتائج الآتية:

- اقتصار الإحصاءات الوصفية فقط على الأعمدة الرقمية، وهي تُمثِّل في المثال السابق كُلًّا من: (Discount Percentage)، و(Mrp)، و(Number Of Ratings)، و(Number Of Review)، و(Star Rating)، و(Sale Price).
- الصف الأوَّل يُظهِر التكرار (Count) الذي يُمثِّل عدد القِيَم غير المفقودة في كل عمود. فمثلًا، جميع الأعمدة تحتوي على (62) قيمة، في ما يُمثِّل عدد الأسطر نفسها.
- الصف الثاني يُظهِر الوسط الحسابي (Mean) لكل عمود. فمثلًا، الوسط الحسابي لعمود (Sale Price) هو: (80073.887097).

- الصف الثالث يُظهِر الانحراف المعياري (Standard Deviation: std) الذي يقيس مدى تشتُّت القِيَم حول الوسط الحسابي. فمثلًا، الانحراف المعياري لعمود (Number Of Reviews) هو: (2855.883830).
- الصف الرابع يُظهِر القيمة الصغرى (min)، وهي أقلُّ قيمة لكل عمود. فمثلًا، أقلُّ قيمة لعمود (Star Rating) هي (4.500000).
- الصف الخامس يُظهِر المئين 25 (25٪)، ويُمثِّل الربيع الأوَّل، وهو القيمة التي تفصل أوَّل 25٪ من البيانات عن البقيَّة.
- الصف السادس يُظهِر المئين 50 (50٪)، ويُعرَف بالربيع الثاني أو الوسيط، وهو القيمة الوسطى للبيانات.
- الصف السابع يُظهِر المئين 75 (75٪)، ويُمثِّل الربيع الثالث، وهو القيمة التي تفصل 75٪ من البيانات عن أعلى 25٪.
- الصف الثامن يُظهِر القيمة العظمى (max)، وهو أكبر قيمة لكل عمود.

نشاط جماعي

أتأمَّل الشكل (3-10) الذي يُبيِّن ناتج تحليل ملف بيانات للطلبة باستخدام برمجية بايثون للإحصاءات الوصفية، ثمَّ أُجيب - بالتعاون مع أفراد مجموعتي- عن الأسئلة التي تليه:

	grade1	grade2	grade3	average
count	20.000000	20.000000	20.000000	20.000000
mean	79.200000	81.000000	79.600000	79.933333
std	21.199801	18.037972	18.556883	18.265417
min	20.000000	35.000000	38.000000	31.000000
25%	66.000000	74.500000	68.250000	68.250000
50%	89.500000	88.000000	88.000000	89.166667
75%	90.500000	92.000000	93.250000	92.000000
max	100.000000	100.000000	100.000000	100.000000

الشكل (3-10): ناتج تحليل بيانات الطلبة باستخدام الإحصاءات الوصفية.

- **أستخرج من الجدول ما يأتي:**
  - عدد الطلبة في الملف.
  - أقلُّ مُعدَّل وأعلى مُعدَّل في الصف.
  - الوسط الحسابي للعمود garde2.
  - الانحراف المعياري للعمود grade3.
  - الوسيط للعمود garde2.
  - الربيع الثالث للعمود grade1.

نشاط فردي

- أفتح مُفسِّر لغة البرمجة بايثون، ثمَّ أعمل على استيراد مكتبة(pandas).
- أستخدم الدالَّة read_csv لقراءة الملف من مكتبة برمجية بايثون.
- أُنفِّذ الإحصاءات الوصفية لملف بياناتي الذي استدعيته، ثمَّ أطبعها على شاشة المُخرَجات.
- أُفسِّر النتائج التي ظهرت على الشاشة.

إضاءة

اسم ملفات البيانات (CSV) هو اختصار للاسم الكامل (Comma Separated Values) الذي يعني القِيَم المفصولة بفواصل. تُستخدَم هذه الصيغة في تخزين البيانات على هيئة جدول، حيث تُفصَل القِيَم داخل الصف الواحد باستخدام الفواصل (,). وللتحقُّق من ذلك، يجب أنْ يحتوي اسم الملف على الامتداد (.csv). أمّا إذا كان الملف محفوظًا بصيغة أُخرى (مثل Excel.xlsx)، فيجب تحويله إلى صيغة (CSV) قبل استدعائه في برمجية بايثون. وفي هذا السياق، يُمكِن تحويل ملف إكسل (Excel) إلى صيغة (CSV) عن طريق فتح الملف في برمجية إكسل (Excel)، ثمَّ الضغط على خيار حفظ باسم (Save As)، ثمَّ اختيار صيغة (CSV).

عرض البيانات باستخدام برمجية بايثون:

يوجد العديد من المكتبات الخارجية التي تُستخدَم في برمجية بايثون لعرض البيانات، وتُعَدُّ مكتبة (Matplotlib) من أكثرها شيوعًا؛ إذ تتيح هذه المكتبة إنشاء مُخطَّطات بيانية مُتنوِّعة (مثل: المُخطَّطات الخَطِّية، ومُخطَّطات التشتُّت)؛ ما يساعد على فهم البيانات وعرضها بطرائق واضحة. كذلك تحتوي هذه المكتبة على العديد من الدوال التي تُسهِّل على المُبرمِجين تنسيق المُخطَّطات البيانية المختلفة. أنظر الجدول (3-1) الذي يُبيِّن أهمَّ الدوال الشائعة في مكتبة (Matplotlib)، التي تُستخدَم بشكل مُتكرِّر في تنسيق المُخطَّطات البيانية وإنشائها.

الجدول (3-1): أمثلة على الدوال الشائعة في مكتبة (Matplotlib).

الدالَّة	الوظيفة
Plot()	إنشاء رسم بياني خَطِّي.
Bar()	رسم مُخطَّط بياني على شكل أعمدة.
show()	إظهار الرسم البياني على شاشة المُخرَجات.
title()	إظهار عنوان للرسم البياني.
xlabel()	إظهار عنوان المحور السيني على المُخطَّط البياني.
ylabel()	إظهار عنوان المحور الصادي على المُخطَّط البياني.
Legend()	إظهار مفتاح الرسم البياني.
Pie	رسم المُخطَّط البياني على شكل دائري.

عرض البيانات برسم بياني خَطِّي (Line Chart):

أدرس المقطع البرمجي الذي يُبيِّنه الشكل (3-11)، والذي يهدف إلى رسم مُخطَّط بياني يعرض اسم الطالب ومُعدَّله في صورة رسم بياني.

```
import pandas
import matplotlib.pyplot as plt

data = pandas.read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')

plt.title('average of class')
plt.xlabel('Names')
plt.ylabel('Average')
plt.plot(data['name'], data['average'])
plt.show()
```

الشكل (3-11): مقطع برمجي لرسم مُخطّط بياني.

في ما يأتي الخطوات الواجب اتِّباعها لإنشاء رسم بياني خَطِّي:

1. **استدعاء المكتبات الخارجية:** تُستدعى المكتبات الخارجية باستخدام الأمر (import)؛ إذ يتمُّ به استدعاء مكتبة (Pandas) المسؤولة عن تحليل البيانات، ومكتبة (Matplotlib) المسؤولة عن عرض البيانات. وبعد استدعاء مكتبة (Matplotlib)، يبدأ استخدام وحدة (pyplot)، وهي وحدة فرعية من المكتبة تساعد على رسم المُخطَّطات البيانية في برمجية بايثون.

```
import pandas
import matplotlib.pyplot as plt
```

إضاءة

لتجنُّب كتابة اسم المكتبة الطويل في التعليمات البرمجية، فقد اختصرنا اسم مكتبة (matplotlib) ليصبح (plt)، وذلك باستخدام كلمة (as). وهذا يعني أنَّنا سنستخدم الاختصار (plt) بدلًا من كتابة اسم المكتبة كاملًا في كل أمر؛ ما يُوفِّر الوقت، ويُبسِّط كتابة الكود.

2. **إضافة عنوان إلى المُخطَّط البياني:** يُمكِن إظهار عنوان للمُخطَّط على شاشة المُخرَجات باستخدام الدالَّة **title** من مكتبة (Matplotlib) إذ تعمل هذه الدالَّة على تحديد العنوان الذي سيظهر أعلى المُخطَّط البياني. ومن ثَمَّ يُكتَب الأمر البرمجي لإضافة العنوان على النحو الآتي:

```
plt.title('average of class')
```

3. **إضافة عناوين إلى المحاور في المُخطَّط البياني:** تُستخدَم الدالَّة **xlabel** في إظهار عنوان للمحور السيني (X-axis) على المُخطَّط البياني، في حين تُستخدَم الدالَّة **ylabel** في إظهار عنوان للمحور الصادي (Y-axis) على المُخطَّط نفسه. ومن ثَمَّ يُكتَب الأمر البرمجي لإضافة العناوين مع ملاحظة أنَّه يجب وضع عناوين المحاور بين إشارتي اقتباس على النحو الآتي:

4.

```
plt.xlabel('Names')
plt.ylabel('Average')
```

5. **رسم المُخطَّط البياني الخَطِّي باستخدام الدالَّة plot:** لرسم مُخطَّط بياني خَطِّي، تُستخدَم الدالَّة **plot** من الوحدة (matplotlib.pyplot) التي اختصرناها بـ (plt).

تأخذ الدالَّة plot مُعامِلين أساسيين، هما:

المُعامِل (X-axis): يُمثِّل هذا المُعامِل القِيَم على المحور الأفقي (السيني). وفي هذا المثال، فإنَّه يُعَدُّ المُتغيِّر في عمود (name).

المُعامِل (Y-axis): يُمثِّل هذا المُعامِل القِيَم على المحور العمودي (الصادي). وفي هذا المثال، فإنَّه يُعَدُّ المُتغيِّر في عمود (average).

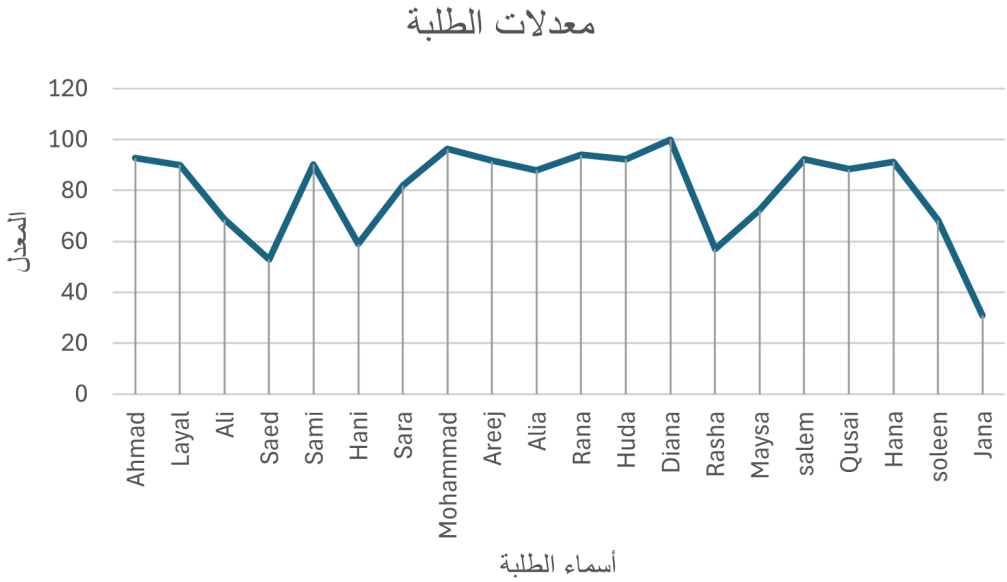
تُكتَب أسماء الأعمدة (المُتغيِّرات) التي سيتمُّ رسمها باستخدام الدالَّة plot، وذلك بوضع اسم العمود بين قوسين مُربَّعين [ ] بعد اسم المُتغيِّر الذي يحتوي على البيانات، وهو في هذا المثال (data). ومن ثَمَّ يُكتَب الأمر البرمجي لرسم المُخطَّط البياني على النحو الآتي:

```
plt.plot(data['name'], data['average'])
```

6. **إظهار الرسم البياني على شاشة المُخرَجات:** يتمُّ إظهار الرسم البياني على شاشة المُخرَجات باستخدام الدالَّة show() من مكتبة (matplotlib.pyplot)؛ إذ تعمل هذه الدالَّة على عرض الرسم البياني الذي أُنشِئ باستخدام التعليمات السابقة.

```
plt.show()
```

عند تنفيذ هذه الدالَّة، سيظهر الرسم البياني الناتج بناءً على التعليمات التي كُتِبت مُسبَّقًا (مثل: إضافة العنوان، وأسماء المحاور، ورسم البيانات). أنظر الشكل (3-12).



الشكل (3-12): مثال على عرض البيانات في هيئة رسم بياني خَطِّي باستخدام مكتبة (Matplotlib) في برمجية بايثون.

بعد أنْ أتأمَّل الرسم البياني وأُحلِّله، أُلاحِظ أنَّ الطالبة Jana حصلت على أدنى مُعدَّل في الصف، وأنَّ الطالبة Diana حصلت على أعلى مُعدَّل في الصف، وأنَّه يُمكِن استخدام الرسم البياني في المقارنة بين مُعدَّلات الطلبة، مثل المقارنة بين مُعدَّلي طالبين/ طالبتين لمعرفة الفَرْق بينهما بسهولة.

كذلك يُمكِن المقارنة بين علامتين لنفس الطالب/ الطالبة لتعرُّف مدى التحسُّن أو التأخُّر في التحصيل الدراسي، وذلك برسم خَطَّين بيانيين على نفس الرسم باستخدام أكثر من أمر (plot). أنظر الشكل (3-13).

```
import pandas
import matplotlib.pyplot as plt
data = pandas read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')
plt.title('average of Class')
plt.xlabel('Names')
plt.ylabel('Average')
plt.plot(data['name'], data['grade1'])
plt.plot(data['name'], data['grade2'])
plt.legend(['grade1','grade2'])
plt.show()
```

الشكل (3-13): مقطع برمجي للمقارنة بين البيانات باستخدام الأمر (plot).

أُلاحِظ في الأمر قبل الأخير استخدام الدالَّة legend التي تُظهِر مفتاحًا للرسم البياني، ويَظهر مفتاح الرسم على هيئة مُربَّع يقع في الزاوية اليمنى من الرسم على شاشة المُخرَجات، ويحتوي على ألوان الخطوط وما يرمز إليه كل خَطٍّ منها.

بعد الانتهاء من تنفيذ المقطع البرمجي السابق في مُفسِّر لغة البرمجة بايثون، فإنَّ ناتج التنفيذ سيكون على النحو الظاهر في الشكل (3-14).

الشكل (3-14): ناتج تنفيذ المقطع البرمجي في الشكل (3-13).

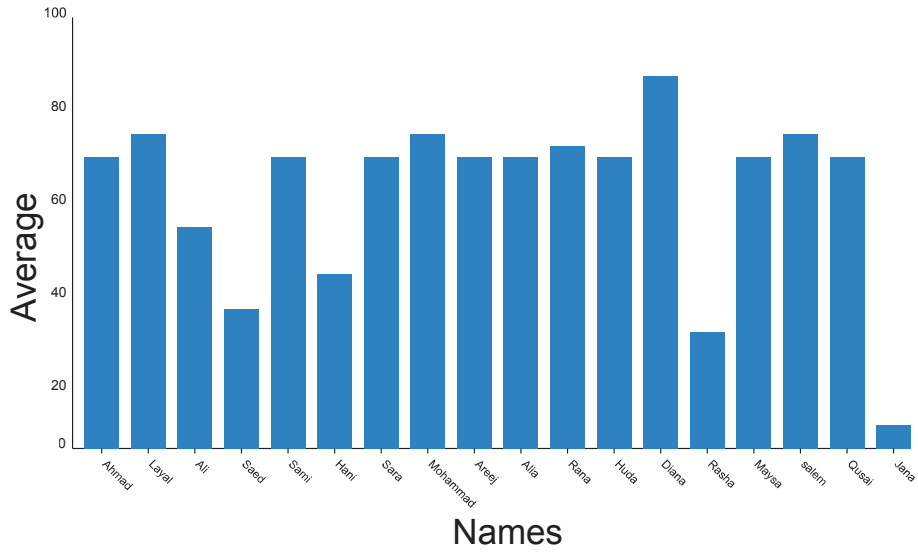
عرض البيانات على هيئة رسم بياني بالأعمدة (Bar Chart):

عند التعامل مع بيانات تتطلَّب المقارنة بين القِيَم من خلال فئات أو مجموعات، فإنَّ الرسم البياني بالأعمدة (Bar Chart) يُعَدُّ الأنسب لهذا الغرض. يُمكِن إنشاء رسم بياني من هذا النوع باستخدام مكتبة (Matplotlib) في برمجية بايثون، وذلك باعتماد الدالَّة **bar** بدلًا من الدالَّة **plot**، فيصبح البرنامج على النحو الظاهر في الشكل (3-15).

```
import pandas
import matplotlib.pyplot as plt
data = pandas read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')
plt.title('average of Class')
plt.xlabel('Names')
plt.ylabel('Average')
plt.bar(data['name'], data['average'])
plt.show()
```

الشكل (3–15): مقطع برمجي لعرض البيانات على هيئة رسم بياني بالأعمدة (bar) باستخدام مكتبة (Matplotlib).

بعد الانتهاء من تنفيذ المقطع البرمجي السابق في مُفسِّر لغة البرمجة بايثون، فإنَّ ناتج التنفيذ سيكون على النحو الظاهر في الشكل (3–16).



الشكل (3–16): ناتج تنفيذ المقطع البرمجي في الشكل (3–15).

كذلك يُمكِن التحكُّم في لون الأعمدة وعرضها باستخدام الوسيط (color) والوسيط (width) في الدالَّة bar، فيكون لون الأعمدة الافتراضي أزرقَ، وعرض الأعمدة الافتراضي (0,8). وإذا أردنا تخصيص اللون والعرض، فإنَّه يُمكِننا كتابة الأوامر على النحو الآتي:

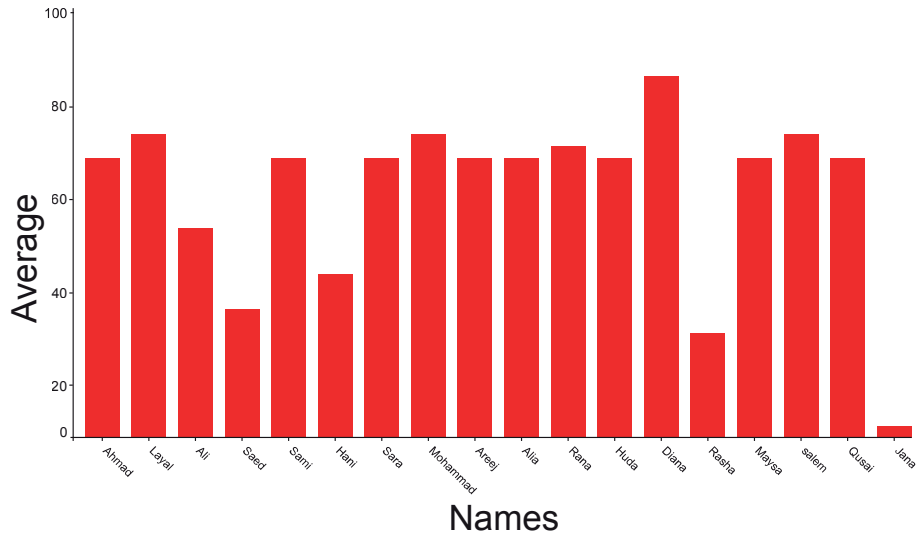
```
plt.bar(data['name'], data['average'], color='red', width=0.5)
```

يعمل الأمر على رسم مُخطَّط بياني في صورة أعمدة، ويُحدَّد عنوان محور السينات باسم (names)، ويُحدَّد عنوان محور الصادات باسم (average)، ثمَّ يُخصَّص لون الأعمدة ليكون بالأحمر ('color='red)، ويُضبَط عرض الأعمدة ليكون بقيمة (width=5 ,0). أنظر الشكل (3-17).

```
import pandas
import matplotlib.pyplot as plt
data = pandas read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')
plt.title('average of Class')
plt.xlabel('Names')
plt.ylabel('Average')
plt.bar(data['name'], data['average'], color='red', width=0.5)
plt.show()
```

الشكل (3-17): مقطع برمجي لتمثيل البيانات باستخدام الرسم البياني بالأعمدة (Bar Chart) من مكتبة (Matplotlib)، والتحكُّم في لون الأعمدة وعرضها.

بعد الانتهاء من تنفيذ المقطع البرمجي السابق في مُفسِّر لغة البرمجة بايثون، فإنَّ ناتج التنفيذ سيكون على النحو الظاهر في الشكل (3-18).



الشكل (3-18): ناتج تنفيذ المقطع البرمجي في الشكل (3-17).

عرض البيانات على شكل قطاع دائري (Pie Chart):

يُعَدُّ مُخطَّط القطاع الدائري أداة فاعلة لعرض البيانات في صورة أجزاء من الكل؛ إذ يُستخدَم في تمثيل النسب المئوية لكل مُتغيِّر بسهولة؛ ما يتيح للمُستخدِم المقارنة بين المُكوِّنات المختلفة ضمن مجموعة البيانات. يمتاز القطاع الدائري بقدرته على إظهار النسبة المئوية لكل عنصر مقارنةً بالإجمالي؛ ما يُسهِّل على المُستخدِم فهم كيف يُسهِم كل عنصر ضمن المجموعة الكاملة.

تُستخدَم الدالَّة **pie** من مكتبة (Matplotlib) في عرض البيانات على هيئة قطاع دائري، وتتطلَّب الدالَّة إدخال قائمة أو سلسلة من القِيَم التي تُمثَّل بوصفها نسبًا مئويةً من إجمالي (100).

خطوات عرض البيانات على هيئة قطاع دائري (Pie chart) في برمجية بايثون:

1. **استدعاء المكتبات:** يتمثَّل ذلك في استدعاء مكتبة (matplotlib.pyplot) للوظائف الخاصة بالرسوم، ومكتبة (numpy) لإنشاء مصفوفة تحتوي على النسب المئوية.

```
import matplotlib.pyplot as plt
import numpy as np
```

2. **تعريف البيانات:** يكون ذلك بإضافة مصفوفة تحتوي على القِيَم المئوية لكل قسم وقائمة تضمُّ أسماء الأقسام.

```
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
```

3. **إنشاء القطاع الدائري:** يُمكِن إنشاء القطاع الدائري باستخدام القِيَم الموجودة في y والأسماء الموجودة في (mylabels).

```
plt.pie(y, labels=mylabels)
```

4. عرض القطاع الدائري وإظهاره على الشاشة في برمجية بايثون.

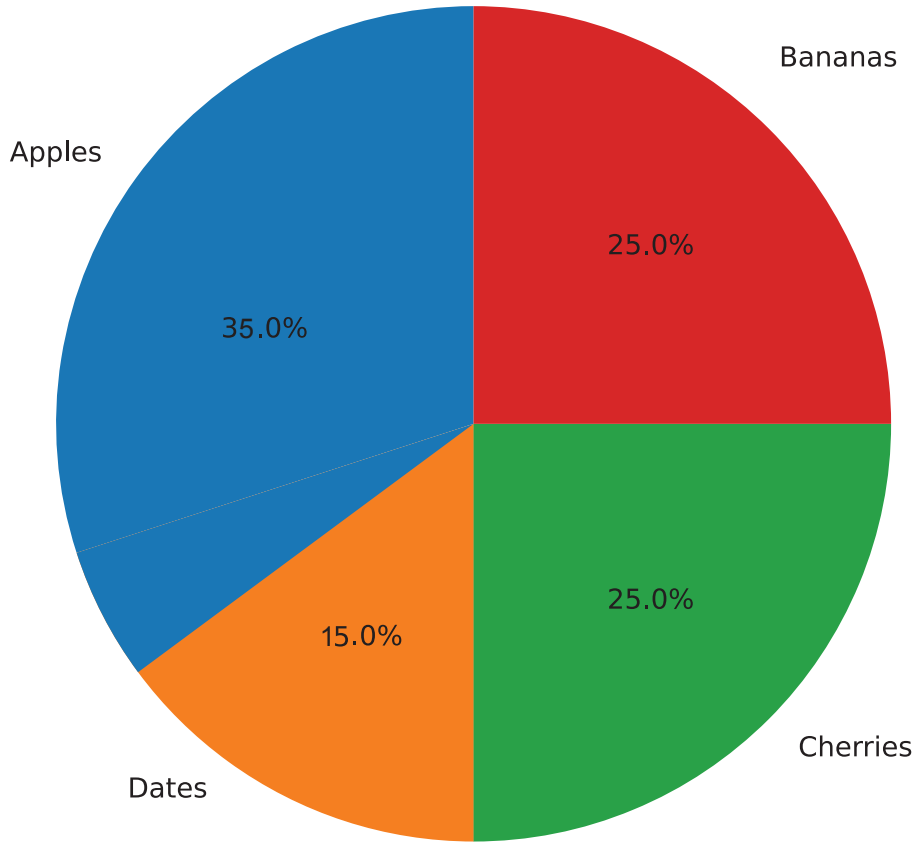
```
plt.show()
```

أنظر الشكل (3-19) الذي يُبيِّن المقطع البرمجي كاملًا، ويُظهِر التمثيل البياني في صورة قطاع دائري (Pie chart) ضمن برمجية بايثون.

```
import matplotlib.pyplot as plt
import numpy as np
y = np.array([35, 25, 25, 15])
mylabels = ["Apples", "Bananas", "Cherries", "Dates"]
plt.pie(y, labels=mylabels)
plt.show()
```

الشكل (3-19): مقطع برمجي لتمثيل البيانات باستخدام الرسم البياني بالقطاع الدائري (Pie Chart) في مكتبة (Matplotlib).

بعد الانتهاء من تنفيذ المقطع البرمجي السابق في مُفسِّر لغة البرمجة بايثون، فإنَّ ناتج التنفيذ سيكون على النحو الظاهر في الشكل (3-20).



الشكل (3-20): ناتج تنفيذ المقطع البرمجي في الشكل (3-19).

نشاط فردي

أفتح مُفسِّر لغة البرمجة بايثون، ثمَّ أستدعي من مكتبة (Pandas) ملف بيانات الطلبة الذي أنشأته في الدرس السابق بعد حفظه بصيغة (CSV)، ثمَّ أُنفِّذ ما يأتي:

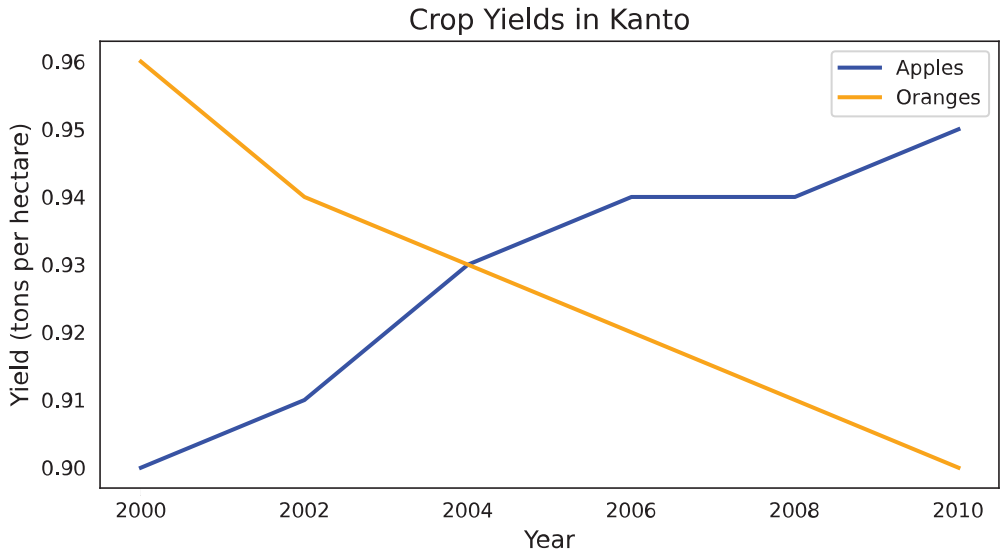
1. أستعين بمكتبة (Matplotlib) لرسم مُخطَّط بياني خَطِّي يُوضِّح العلاقة بين مُتغيِّرين من المُتغيِّرات التي لديَّ، وأتأكَّد أنَّ نوع البيانات مناسب للتمثيل البياني الذي أشتغل به، ثمَّ أُضيف عنوانًا إلى المُخطَّط البياني يدلُّ على محتواه. بعد ذلك أُسمّي محور السينات، ثمَّ محور الصادات.
2. أرسم مُخطَّطًا بيانيًّا آخرَ على هيئة أعمدة، وأختار اللون الأصفر ليكون هو لون الأعمدة، ثمَّ أُحدِّد عرض الأعمدة بـ(0,6).
3. أختار مُتغيّـرًا مناسبًا، ثمَّ أعمل على رسم تمثيل بالقطاع الدائري، وذلك باستخدام مكتبة (Matplotlib)، ثمَّ أُضيف أسماء واضحة لكل جزء في القطاع، وأُظهِر النسب المئوية لكل جزء، وأُضيف كذلك عنوانًا يُوضِّح محتوى القطاع الدائري.

بعد ذلك أتحقَّق من عرض المُخطَّطات الثلاثة بطريقة واضحة ومُنظَّمة، وأتأكَّد أنَّ لكل مُخطَّط مفتاحًا يُوضِّح ما يُمثِّله.

تحديد الأنماط في الرسوم البيانية:

مثال: تحليل الرسم البياني الخَطِّي.

أتأمَّل الشكل (3-21) الذي يُمثِّل الرسم البياني الخَطِّي الذي أُنشِئ باستخدام برمجية بايثون لتمثيل بيانات عن مجموع محصول التفّاح والبرتقال في بلدة (Kanto) خلال الأعوام (2000-2010م). أُلاحِظ من الرسم البياني أنَّ الخَطَّ الأزرق يُمثِّل مجموع إنتاج التفّاح في البلدة على مدار تلك الأعوام، وأنَّ اتِّجاه البيانات موجب؛ ما يعني أنَّ إنتاج التفّاح قد شهد زيادة تدريجية أثناء تلك الأعوام. أمّا الخَطُّ البرتقالي فيُمثِّل مجموع إنتاج البرتقال خلال المُدَّة الزمنية نفسها. كذلك أُلاحِظ أنَّ اتِّجاه البيانات سالب؛ ما يعني أنَّ إنتاج البرتقال انخفض تدريجيًّا بمرور الوقت. ومن ثَمَّ أستنتج أنَّ الرسم البياني يشير إلى حدوث تغيُّرات واضحة في إنتاج المحاصيل ببلدة (Kanto)؛ إذ شهد إنتاج التفّاح زيادة مستمرة، في حين انخفض إنتاج البرتقال بصورة تدريجية. إنَّ هذه الاتِّجاهات قد تقودنا إلى البحث في الأسباب التي أفضت إلى تلك النتائج بهدف اتِّخاذ قرارات مناسبة.



الشكل (3-21): التمثيل البياني الخَطِّي لمجموع محصول التفّاح ومحصول البرتقال في بلدة (Kanto) خلال الأعوام (2000-2010م).

المواطنة الرقمية:

- **جمع البيانات بشكل أخلاقي:** أتجنَّب جمع البيانات من دون أخذ إذن المُستخدِمين. كذلك أستخدم مصادر مفتوحة للبيانات عند تحليل البيانات العامة، وأُوضِّح الأهداف والغايات عند جمع البيانات.
- **الأمان السيبراني وحماية البيانات:** أستخدم التشفير عند تخزين البيانات الحسّاسة ونقلها، وأتجنَّب مشاركة البيانات مع أطراف ثالثة من دون إذن، وأعمل على تأمين الملفات والقواعد باستخدام المكتبات، مثل مكتبة (cryptography).

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أكتب التعليمات البرمجية المسؤولة عن كلٍّ ممّا يأتي في لغة البرمجة بايثون:

- تثبيت المكتبة الخارجية (pandas).
- إجراء الإحصاءات الوصفية لملف بيانات يحمل اسم (data).
- عرض البيانات على هيئة مُخطَّط بياني خَطِّي.

السؤال الثاني: أدرس الجملة البرمجية الآتية، ثمَّ أُجيب عن الأسئلة التي تليها:

```
plt.bar(data['name'], data ['average'], color = 'red', width = 0.5)
```

1. ما المكتبة الخارجية التي يجب استدعاؤها؟ أكتب التعليمة البرمجية المسؤولة عن ذلك.
2. ما هدف هذه التعليمة؟
3. ما الهدف من الدالَّة color؟
4. إذا غُيِّرت قيمة (width) من 0.5 إلى 0.9، فما التغيير المُتوقَّع في الرسم البياني؟
5. ما اسم ملف البيانات؟
6. ما عنوان المحور السيني؟ وما عنوان المحور الصادي؟

السؤال الثالث: أُحدِّد وظيفة كل دالَّة من الدوال الآتية:

اسم الدالَّة (Function Name)	الوظيفة
title	
legend	
show	
import	
set_option	
info	

السؤال الرابع: أختار رمز الإجابة الصحيحة في كلٍّ ممّا يأتي:

1. **المكتبة التي تُستخدَم في تحليل البيانات في برمجية بايثون هي:**
   - أ. مكتبة (NumPy).
   - ب. مكتبة (Matplotlib).
   - ج. مكتبة (Pandas).
   - د. مكتبة (SciPy).

2. **الدالَّة التي تُستخدَم في قراءة ملفات (CSV) في مكتبة (Pandas) هي:**
   - أ. `read_excel`.
   - ب. `read_csv`.
   - ج. `read_data`.
   - د. `read_file`.

3. **يظهر الرسم البياني على شاشة المُخرَجات باستخدام الدالَّة:**
   - أ. `plot()`.
   - ب. `show()`.
   - ج. `display()`.
   - د. `describe()`.

4. **الدالَّة التي تُستخدَم في إضافة عنوان إلى المُخطَّط البياني في مكتبة (Matplotlib) هي:**
   - أ. `title()`.
   - ب. `xlabel()`.
   - ج. `ylabel()`.
   - د. `legend()`.

5. **يظهر مفتاح الرسم البياني في مكتبة (Matplotlib) باستخدام الدالَّة:**
   - أ. `title()`.
   - ب. `xlabel()`.
   - ج. `ylabel()`.
   - د. `legend()`.

السؤال الخامس: أُميِّز العبارات الصحيحة من العبارات غير الصحيحة في ما يأتي:

- تُستخدَم الدالَّة `describe()` في حساب الإحصاءات الوصفية للبيانات في مكتبة (Pandas).
- يُمكِن استخدام مكتبة (Matplotlib) في رسم المُخطَّطات البيانية الخَطِّية فقط.
- الدالَّة `xlabel()` تُستخدَم في إضافة عنوان إلى المحور السيني في المُخطَّط البياني.
- الدالَّة `read_excel()` تُستخدَم في قراءة ملفات إكسل (Excel) في مكتبة (Pandas).
- الدالَّة `bar()` تُستخدَم في رسم المُخطَّط البياني بالأعمدة في مكتبة (Matplotlib).

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أدرس المقطع البرمجي الآتي، ثمَّ أُجيب عن الأسئلة التي تليه:

```
import pandas
import matplotlib.pyplot as plt
data = pandas.read_csv('C:\\Users\\user\\OneDrive\\Desktop\\testdoc.csv')
plt.title('avarage of class')
plt.xlabel ('Names')
plt.ylabel('Average')
plt.plot(data['name'], data ['average'])
plt.show ()
```

ما اسم المكتبات الخارجية التي تمَّ استيرادها في المقطع البرمجي؟ وما هدفها؟

- كيف استُدعِي ملف البيانات؟
- ما هدف الدالَّة `show` في الجملة الأخيرة؟
- إذا أردْتُ تغيير عنوان الرسم البياني إلى 'average of grades'، فما التعديل الذي سأُجريه على المقطع البرمجي؟
- ما شكل الرسم البياني الذي سيظهر على شاشة المُخرَجات؟

السؤال الثاني: لديَّ بيانات موجودة في ملف "students.csv". اعتمادًا على هذه البيانات:

- أكتب الكود البرمجي الصحيح لاستدعاء الملف في برمجية بايثون باستخدام مكتبة (pandas).
- أكتب الكود البرمجي الصحيح لعرض جميع الأسطر والأعمدة الخاصة بالبيانات التي تمَّ قراءتها باستخدام مكتبة (pandas).
- أكتب الكود البرمجي الصحيح لتحليل الإحصاءات الوصفية للبيانات.
- أكتب الكود البرمجي الصحيح لرسم مُخطَّط بياني خَطِّي وإظهاره في برمجية بايثون، ثمَّ أُضيف عنوانًا للتمثيل وعناوينَ للمحاور ومفتاحًا للتمثيل.

331/990
Test Status Controller

الدرس الرابع

علاقة السبب والنتيجة في البيانات
(Data Cause–and–Effect Relation)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس مفهوم علاقة السبب والنتيجة، وأتعلَّم طريقة تفسيرها، إضافةً إلى كيفية استنتاج هذه العلاقة من البيانات المُتوافِرة، والتنبُّؤ بالنتائج بناءً على الأسباب. كذلك سأتعرَّف مفهوم العلاقة الزائفة وكيفية التمييز بينها وبين العلاقات الحقيقية.

المفاهيم والمصطلحات:

علاقة السبب والنتيجة (Cause and Effect Relationship)، الارتباط (Correlation)، الأسبقية الزمنية (Temporal Precedence)، العلاقة الزائفة (Spurious Relationship)، المُتغيِّر الزائف (Spurious Variable)، التحليل السببي (Causal Analysis).

نتاجات التعلُّم (Learning Outcomes):

- أُوضِّح مفهوم علاقة السبب والنتيجة.
- أُفسِّر علاقة السبب والنتيجة.
- أستنتج علاقة السبب والنتيجة من البيانات.
- أتنبَّأ بالنتيجة عن طريق السبب، وأتنبَّأ بالسبب عن طريق النتيجة.

تعرَّفْتُ سابقًا البيانات والمعلومات والمعرفة والعلاقة بينها. كذلك توجد علاقات بين البيانات نفسها في علم البيانات، فما هذه العلاقات؟ وما تأثيرها في المعلومات؟

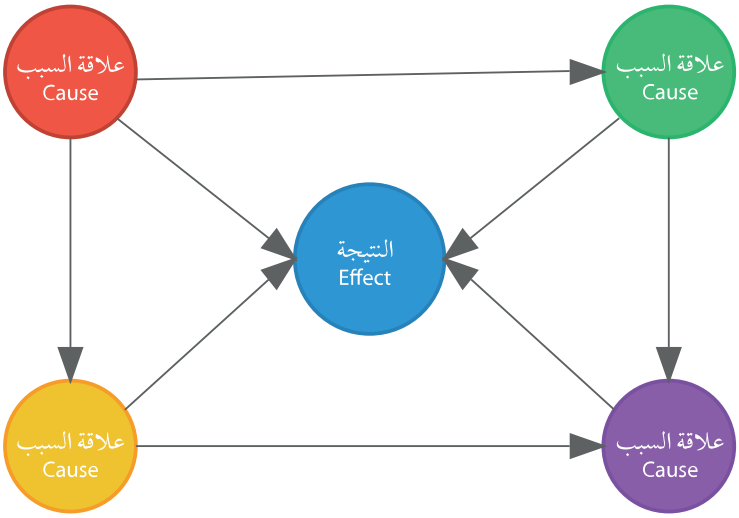
نشاط تمهيدي

أقرأ الحكمة الآتية، ثمَّ أُجيب عن الأسئلة التي تليها:

"مَنْ جَدَّ وجد، ومَنْ زرع حصد".

- أُوضِّح مفهوم الحكمة من وجهة نظري.
- إلامَ تدعو هذه الحكمة؟ أُفسِّر إجابتي.
- ماذا يحدث لمَنْ يزرع؟
- ما نتيجة عدم الجِدِّ؟

علاقة السبب والنتيجة (Cause and Effect):



يُنظَر إلى علاقة السبب والنتيجة بوصفها مفهومًا محوريًّا في علم البيانات، يُعرَف **بالسببية (Causality)**، ويُستخدَم في فهم الروابط وتحليلها، وفهم العلاقات بين الأحداث والنتائج. ومن ثَمَّ، فهو يُقدِّم أساسًا لاختيار الإجراءات التي تُسهِم في تحقيق النتائج المنشودة، ويساعد أيضًا على تحديد الأنماط، والتنبُّؤ بالنتائج المستقبلية، وفهم كيف يُؤثِّر حدث مُعيَّن في تحقيق نتيجة مُعيَّنة.

تهدف علاقة السبب والنتيجة إلى توضيح العلاقات والروابط بين المُتغيِّرات، وذلك بتحليل تأثير تغيُّر أحد المُتغيِّرات (السبب) في الآخر (النتيجة). وفي سياق تحليل البيانات، يُركِّز المُحلِّلون على دراسة هذه العلاقات لفهم كيفية تأثير العوامل المختلفة في النتائج؛ ما يساعد على استخلاص استنتاجات دقيقة تدعم اتِّخاذ قرارات مَبْنية على البيانات.

فهم العلاقة السببية وأهميتها:

من أهم مزايا تعرُّف العلاقات السببية في البيانات إمكانيةُ التنبُّؤ بالنتائج بناءً على الأسباب. وهذه العلاقات تظهر بصورة واضحة وجَلِيَّة في حياتنا اليومية كما يَتَّضِح من المثالين الآتيين.

مثال (1):

إذا لم يتوقَّف سائق سيّارة عند إشارة المرور الحمراء، وأدّى ذلك إلى وقوع حادث بالسيّارة، فإنَّ السبب هو عدم التوقُّف عند الإشارة الحمراء، والنتيجة هي وقوع الحادث.

مثال (2):

أفترض وجود بيانات عن النشاط البدني والأمراض القلبية. عند تحليل هذه البيانات، قد يُظهِر التحليل أنَّ الأشخاص الذين يمارسون الرياضة بانتظام تقلُّ لديهم مُعدَّلات الإصابة بالأمراض القلبية. في هذه الحالة، يُمكِنني استنتاج وجود علاقة سببية بين النشاط البدني والأمراض القلبية، تتمثَّل في أنَّ زيادة النشاط البدني (السبب) تُسهِم في تقليل خطر الإصابة بالأمراض القلبية (النتيجة). أنظر الجدول (4-1) الذي يتضمَّن أمثلة حياتية أُخرى على علاقة السبب والنتيجة.

الجدول (4-1): أمثلة حياتية على علاقة السبب والنتيجة.

السبب (Cause)	النتيجة (Effect)
عدم تنظيف الأسنان أبدًا.	حدوث تسوُّسات كثيرة في الأسنان.
تناول الطعام غير الصِّحِّي بانتظام.	زيادة الوزن، أو الإصابة بمشكلات صِحِّية.
عدم صيانة السيّارة بصورة دورية.	زيادة احتمال حدوث أعطال مُفاجِئة.
المُذاكَرة الجيِّدة قبل الامتحان.	إحراز علامات أعلى في الامتحان.

نشاط جماعي

أقرأ الجمل الآتية، ثمَّ أُحدِّد - بالتعاون مع أفراد مجموعتي- النتيجة التي قد تنتج من السبب في كلٍّ منها:

1- ازدياد عدد الحصص التدريبية في المدرسة.

2- تنويع العروض الترويجية لمُنتَج ما.

3- نزول الأمطار الغزيرة في منطقة ما.

4- انخفاض أسعار الوقود.

أُقارِن الإجابة التي توصَّلنا إليها في المجموعة بإجابات أفراد المجموعات الأُخرى، ثمَّ نُفسِّر معًا سبب الاختلاف بينها (إنْ وُجِد).

ضوابط علاقة السبب والنتيجة:

توجد مجموعة من الضوابط والشروط التي تَحْكم إنشاء علاقة السبب والنتيجة بين مُتغيِّرين، وهي على النحو الآتي:

1. **الارتباط (Correlation):** يتعيَّن وجود ارتباط واضح بين المُتغيِّرين أو الحدثين اللذينِ يُمثِّلان السبب والنتيجة، بحيث يَظهر أنَّ تغيُّر أحد المُتغيِّرين يتزامن مع تغيُّر الآخر. وبالرغم من ذلك، فإنَّ وجود الترابط وحده لا يكفي لإثبات العلاقة السببية بينهما.

مثال (1):

الأفراد الذينَ يحظَون بمستوى أعلى من التعليم يُرجَّح أنْ يحصلوا على دخل أعلى؛ إذ يوجد ارتباط بين التعليم والدخل، وقد تكون بينهما علاقة سببية.

مثال (2):

سقوط قِطَّة من أعلى إحدى الأشجار في الغابة لا يُمكِن أنْ يكون له أيَّة علاقة باندلاع حريق في الغابة؛ إذ لا يوجد ارتباط بين سقوط القِطَّة واندلاع الحريق. ومن ثَمَّ، فلا يُمكِن أنْ توجد علاقة سببية بينهما.

2. **الأسبقية الزمنية (Temporal Precedence):** بعد إثبات وجود ارتباط بين المُتغيِّرين أو الحدثين، يجب الانتباه إلى شرط الأسبقية الزمنية؛ أيْ أنْ يسبق المُتغيِّرُ المستقلُّ (السبب) المُتغيِّـرَ التابعَ (النتيجة) في الزمن؛ ما يعني أنَّ السبب يحدث أوَّلًا، ثمَّ تظهر النتيجة.

مثال:

قد يتسبَّب إلقاء عَقِب سيجارة في الغابة في حدوث حريق فيها؛ إذ لا يُمكِن أنْ يحدث الحريق قبل إلقاء عَقِب السيجارة؛ ما يُؤكِّد شرط الأسبقية الزمنية بأنْ يسبق السببُ النتيجةَ.

3. **التفسير النظري (Theoretical Explanation):** لا بُدَّ من وجود تفسير منطقي أو نظرية داعمة لتفسير كيف يُؤثِّر السبب في النتيجة، وبيان مُسوِّغات ذلك.
4. **غياب العلاقات الزائفة (Non-Spurious Relationship):** يجب التأكُّد أنَّ العلاقة بين المُتغيِّرين أو الحدثين هي علاقة مباشرة، وليست نتيجة وجود مُتغيِّر خارجي ثالث له علاقة بالمُتغيِّرين أو الحدثين (علاقة خَفِيَّة تُؤثِّر في كلٍّ من السبب والنتيجة).

مثال:

كلَّما زاد حجم حذاء الطفل، ازدادت المعرفة لديه.

في هذا المثال، يوجد مُتغيِّر ثالث (العمر) يُؤثِّر في كلٍّ من حجم الحذاء والمعرفة؛ فكلَّما زاد عمر الطفل، زاد حجم حذائه. وكلَّما زاد عمر الطفل، زادت معرفته نتيجة التعلُّم والنضج. ومن ثَمَّ، فإنَّ العلاقة بين حجم الحذاء والمعرفة هي علاقة زائفة، والارتباط بينهما ناتج من المُتغيِّر الثالث (العمر).

عندما يرتبط مُتغيِّران فقط بسبب تأثُّرهما بنفس العامل (السبب المُشترَك)، فإنَّ هذا الارتباط يوصَف بالزائف. وفي حال إزالة السبب المُشترَك (المُتغيِّر الثالث)، فإنَّ العلاقة بين المُتغيِّرين تختفي، ويصبح الارتباط بينهما صفرًا.

ففي المثال السابق، إذا حُذِف مُتغيِّر العمر (السبب المُشترَك) من علاقة حجم الحذاء بالمعرفة لدى الطفل، فإنَّ العلاقة بين حجم الحذاء والمعرفة تصبح صفرًا؛ أيْ إنَّه لا توجد علاقة بينهما أصلًا.

نشاط جماعي

أقرأ الجمل الآتية، ثمَّ أُبيِّن – بالتعاون مع أفراد مجموعتي– الجمل التي تتضمَّن علاقة سببية زائفة:

1- كلَّما زاد عدد الطيور المهاجرة، زادت مبيعات المعاطف الشتوية.

2- ارتفاع أسعار الوقود أدّى إلى تقليل عدد الرحلات بالسيّارات الخاصة.

3- زيادة ممارسة التمارين الرياضية اليومية تُسهِم في تحسين جودة النوم.

4- كلَّما زاد استخدام مُستحضَرات (كريمات) الوقاية من الشمس، زادت حالات سرطان الجلد.

أُناقِش الإجابات مع زملائي/ زميلاتي في المجموعة، ثمَّ أعرضها أمام أفراد المجموعات الأُخرى بهدف التوصُّل إلى استنتاجات مُشترَكة.

خطوات التحليل السببي:

لاستنتاج علاقة السبب والنتيجة من البيانات، يتعيَّن إجراء التحليل السببي باتِّباع الخطوات الآتية:

1. **تحديد المشكلة بوضوح:** أصوغ المشكلة التي أرغب في تحليلها على شكل علاقة واضحة بين السبب والنتيجة.
2. **تحديد المُتغيِّرات:** أُحدِّد المُتغيِّرات التي أريد دراسة العلاقة السببية بينها، وأُميِّز بين السبب والنتيجة.
3. **جمع البيانات:** أجمع البيانات ذات الصلة بالمُتغيِّرات المُحدَّدة لضمان وجود بيانات كافية ودقيقة للتحليل.
4. **بناء العلاقات:** أُحدِّد كيفية العلاقة بين المُتغيِّرات باستخدام بعض الأدوات، مثل المُخطَّطات البيانية التي تُستخدَم في تحديد الأنماط والارتباطات.
5. **تمييز الارتباط من السببية:** أتأكَّد أنَّ العلاقة بين المُتغيِّرات غير زائفة؛ فالارتباط بين مُتغيِّرين لا يعني بالضرورة أنَّ أحدهما يُسبِّب الآخر.

مثال:

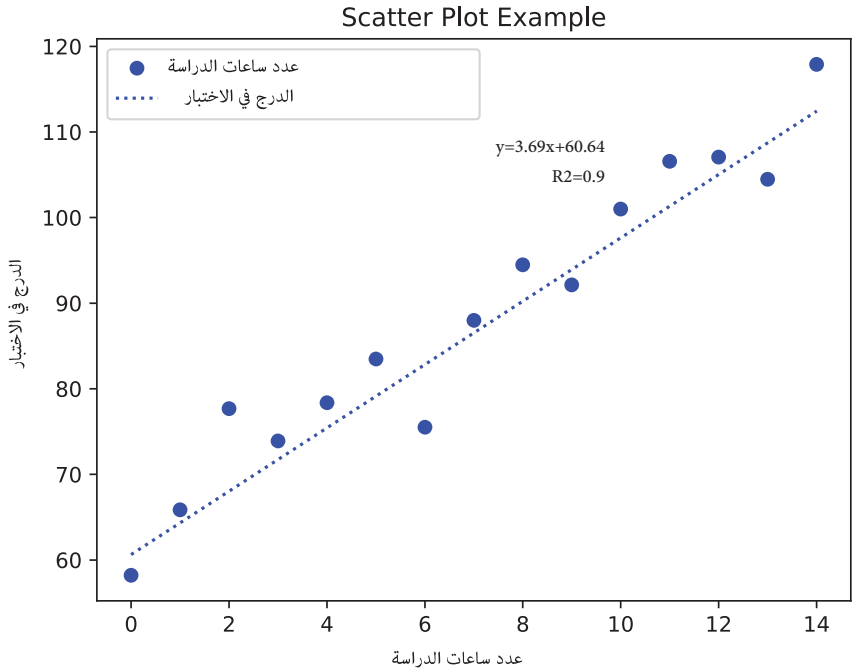
يُبيِّن هذا المثال إجراءات التحليل السببي الخاصة باستكشاف علاقة السبب والنتيجة بين عدد ساعات الدراسة والعلامة المرتفعة في الامتحان:

عدد ساعات الدراسة	العلامة في الامتحان
2	65
3	70
4	75
5	80
6	85
7	90
8	92
9	95
10	98
12	100

1. **تحديد المشكلة:** المشكلة التي يُراد تحليلها هي إيجاد العلاقة بين عدد ساعات الدراسة والعلامة في الامتحان؛ لتعرُّف كيف يُؤثِّر الوقت المُخصَّص للدراسة في الأداء أثناء عقد الامتحان.
2. **تحديد المُتغيِّرات:** المُتغيِّر الأوَّل هو عدد ساعات الدراسة (السبب)، والمُتغيِّر الثاني هو العلامة المرتفعة في الامتحان (النتيجة).
3. **جمع البيانات ذات الصلة بالمُتغيِّرات:** تُجمَع البيانات المُتعلِّقة بالمُتغيِّرات المُحدَّدة لضمان وجود بيانات كافية ودقيقة للتحليل.

4. بناء العلاقات:

يُمكِن تحليل العلاقة بين المُتغيِّرات باستخدام التمثيل البياني في بعض البرامج والبرمجيات، مثل برمجية إكسل (Excel)؛ إذ يتمُّ إنشاء مُخطَّط بياني (Scatter Plot) لبيان العلاقة بين عدد ساعات الدراسة (المحور السيني X) والعلامة في الامتحان (المحور الصادي Y) كما في الشكل (4-1)، وتحليل الأنماط المرصودة في البيانات لتحديد الارتباط.



الشكل (4-1): مُخطَّط تشتُّت (Scatter Plot) يُوضِّح العلاقة بين عدد ساعات الدراسة والعلامات في الامتحان لمجموعة من الطلبة.

أُلاحِظ من المُخطَّط البياني وجود علاقة موجبة بين عدد ساعات الدراسة والعلامة في الامتحان؛ فكلَّما زاد عدد ساعات الدراسة، زادت العلامة في الامتحان؛ ما يعني وجود اتِّجاه موجب في البيانات. كذلك يُمكِن تحليل علاقة السبب والنتيجة باستخدام العديد من الأساليب الإحصائية، مثل أسلوب تحليل الانحدار الخَطِّي البسيط؛ إذ تتيح برمجية إكسل (Excel) – من خلال إعداد التمثيل البياني – إمكانية إضافة معادلة الانحدار الخَطِّي البسيط.

بحسب المعادلة الظاهرة في الشكل السابق (y=3.69 x + 60.64)، فإنَّ:

العلامة في الامتحان= 3.690 × عدد ساعات الدراسة + 60.64

يتبيَّن من المعادلة أنَّ العلامة في الامتحان (النتيجة) تعتمد على عدد ساعات الدراسة، بمُعدَّل (3.69) لكل ساعة دراسية، مضافًا إليها قيمة ثابتة تبلغ (60.64). يُمكِن استخدام هذه المعادلة في التنبُّؤ بعلامة الطالب في الامتحان بناءً على عدد ساعات دراسته. على سبيل المثال، إذا لم يدرس الطالب أبدًا (عدد ساعات الدراسة= 0)، فإنَّ علامته المُتوقَّعة في الامتحان ستكون (60.64).

إضاءة

تحليل الانحدار (Regression Analysis): أسلوب إحصائي في تحليل البيانات، يُستخدَم في دراسة العلاقة بين مُتغيِّرين أو أكثر، ويهدف إلى التنبُّؤ بالقِيَم المستقبلية، أو فهم تأثير مُتغيِّر في مُتغيِّر آخر، وتحديد كَمِّ هذا التأثير.

الانحدار الخطي الخَطِّي البسيط: نموذج رياضي يُستخدَم في دراسة العلاقة بين مُتغيِّرين.

المُتغيِّر المستقل (X): مُتغيِّر يُستخدَم في عملية التنبُّؤ.

المُتغيِّر التابع (Y): مُتغيِّر يُحاوَل التنبُّؤ به بناءً على X.

المعادلة العامة: $Y=a+bX$

حيث:

Y = القيمة المُتوقَّعة للمُتغيِّر التابع.

a = القيمة الابتدائية (المقطع أو الجزء المقطوع من المحور Y).

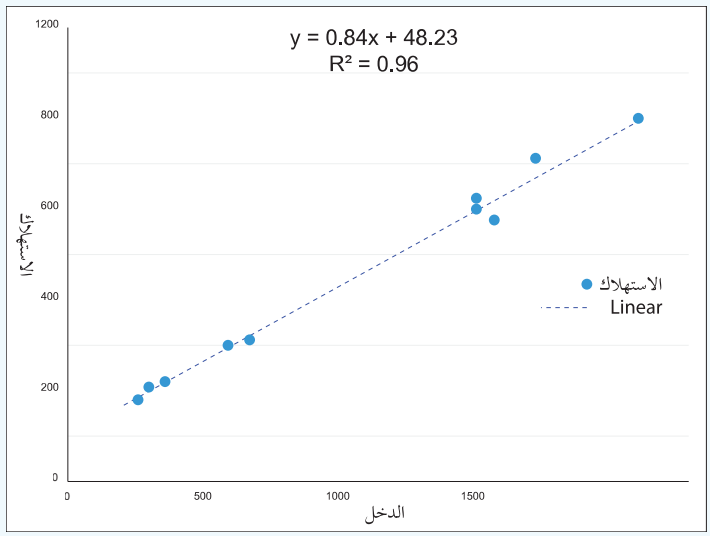
b = ميل الخَطُّ المستقيم، وهو مقدار التغيير في (Y) نتيجة لتغير (X) بوحدة واحدة.

نشاط جماعي

أُجرِي تحليل بيانات لكشف العلاقة بين مُتغيِّري الدخل والاستهلاك (بالدينار الأردني) للفرد في مدينة ما. وقد طُبِّق التحليل على عيِّنة مُكوَّنة من (10) أُسَر، فجاءت النتائج على النحو الآتي:

الدخل:	300	350	500	600	900	1000	900	1200	1050	250
الاستهلاك:	280	340	500	550	800	750	850	1050	1000	250

بعد تمثيل البيانات باستخدام برمجية إكسل (Excel)، نتج مُخطَّط التشتُّت المُبيَّن في الشكل (2-4).



الشكل (2-4): مُخطَّط تشتُّت (Scatter Plot) يُوضِّح العلاقة بين الدخل والاستهلاك لمجموعة من الأفراد في مدينة ما.

كانت نتائج التحليل حسب المعادلة ($y=0.84x + 48.23$) كالآتي:

الاستهلاك= 0.84 × الدخل + 48.23

أُجيب عن الأسئلة الآتية:

1. أُفسِّر المعادلة.
2. إذا كان دخل الفرد (470) دينارًا، فهل يُمكِن التنبُّؤ باستهلاك الأُسْرة؟ أُوضِّح ذلك حسابيًّا باستخدام معادلة الانحدار.
3. إذا كان دخل الفرد صفرًا (أيْ إنَّ الفرد لا يعمل)، فكم يبلغ استهلاكه؟

إثراء

في ما يأتي مثال توضيحي يُبيِّن كيف يُمكِن استخدام مكتبات برمجية بايثون في إيجاد معادلة الانحدار الخَطِّي لتحليل العلاقة بين مُتغيِّرين، مثل الدخل والاستهلاك.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# 1. إنشاء بيانات تجريبية (عدد ساعات الدراسة - الدرجات)
hours_studied = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9]).reshape(-1, 1) # السبب
exam_scores = np.array([50, 55, 60, 65, 70, 75, 80, 85, 90]) # النتيجة

# 2. إنشاء نموذج الانحدار الخطي
model = LinearRegression()
model.fit(hours_studied, exam_scores)

# 3. توقع القيم بناءً على النموذج
predicted_scores = model.predict(hours_studied)

# 4. تحليل العلاقة
slope = model.coef_[0] # الميل (يعبر عن تأثير كل ساعة دراسة على الدرجات)
intercept = model.intercept_ # (الدرجة المتوقعة عند ساعات دراسة = 0) التقاطع
mse = mean_squared_error(exam_scores, predicted_scores) # خطأ التربيع المتوسط

print(f"المعادلة: الدرجات = {slope:.2f} * عدد ساعات الدراسة + {intercept:.2f}")
print(f"خطأ التربيع المتوسط: {mse:.2f}")

# 5. رسم العلاقة
plt.scatter(hours_studied, exam_scores, color='blue', label='البيانات الحقيقية')
plt.plot(hours_studied, predicted_scores, color='red', label='الانحدار الخطي')
plt.title('علاقة ساعات الدراسة بالدرجات')
plt.xlabel('عدد ساعات الدراسة')
plt.ylabel('الدرجات')
plt.legend()
plt.show()

```

المواطنة الرقمية:

- **التفكير الناقد (Critical Thinking):** أحرص عند تقييم العلاقات على عدم افتراض علاقة سببية من دون وجود دليل كافٍ. كذلك أتحقَّق من صِحَّة الروابط بين الأسباب والنتائج، وأعي جيِّدًا أنَّ وجود ارتباط بين مُتغيِّرين لا يعني بالضرورة وجود علاقة سببية بينهما.
- **الأمانة الفكرية (Intellectual Honesty):** إذا كنْتُ أُجري تحليلًا أو أكتب تقريرًا يحتوي على علاقات سببية، فيجب أنْ أُقدِّم المعلومات بصورة واضحة مَبْنِية على الحقائق، وأنْ أتجنَّب تقديم علاقات زائفة بزعم أنَّها حقائق علمية، وبخاصة إذا كانت تُؤثِّر في قرارات الآخرين.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُوضِّح المقصود بالمفاهيم الآتية:

1. علاقة السبب والنتيجة.
2. المُتغيِّر الزائف.
3. الارتباط.
4. الأسبقية الزمنية.

السؤال الثاني: أُوضِّح الفَرْق بين الارتباط والسببية.

السؤال الثالث: أدرس الحالات الآتية، ثمَّ أُحدِّد العلاقة (سببية، زائفة) في كلٍّ منها، وأُوضِّح السبب:

أ. زيادة استهلاك المُثلَّجات (الآيس كريم) تقترن بزيادة عدد حوادث الغرق.

ب. زيادة مبيعات النظّارات الشمسية تقترن بزيادة حالات حروق الشمس.

ج. زيادة كَمِّية المياه المُستهلَكة بناءً على زيادة عدد الأفراد في المنزل.

السؤال الرابع: توجد مجموعة من الضوابط والشروط يجب التزامها لإنشاء علاقة سببية (علاقة السبب والنتيجة). أُوضِّح هذه الضوابط والشروط.

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن السؤالين الآتيين:

السؤال الأوَّل:

أ. **يُظهِر الجدول الآتي البيانات الخاصة باستهلاك الطاقة (بالكيلو واط/ ساعة) ودرجة الحرارة (بالوحدة المئوية):**

استهلاك الطاقة:	300	350	400	450	500
درجة الحرارة:	20	25	30	35	40

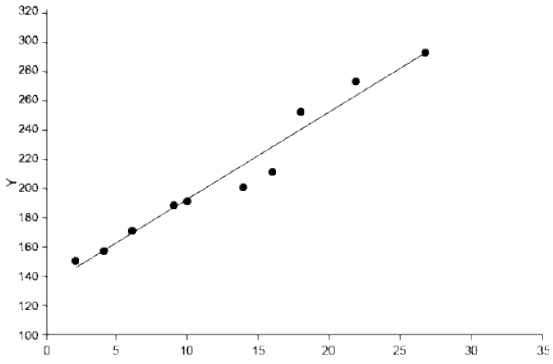
هل يُمكِن استنتاج علاقة سببية بين استهلاك الطاقة ودرجة الحرارة؟ أُوضِّح ذلك.

ب. **يُظهِر الجدول الآتي البيانات الخاصة بالدخل الشهري (بالدينار الأردني) ومقدار الادخار الشهري (بالدينار الأردني) لعيِّنة من الأفراد:**

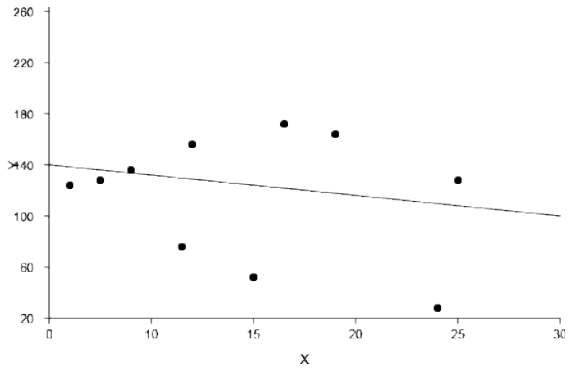
الدخل:	300	350	500	600	900	1000	900	1200	1050	250
الادخار:	20	10	0	50	100	250	50	150	0	0

هل يُمكِن استنتاج علاقة سببية بين الدخل والادخار؟ أُوضِّح ذلك.

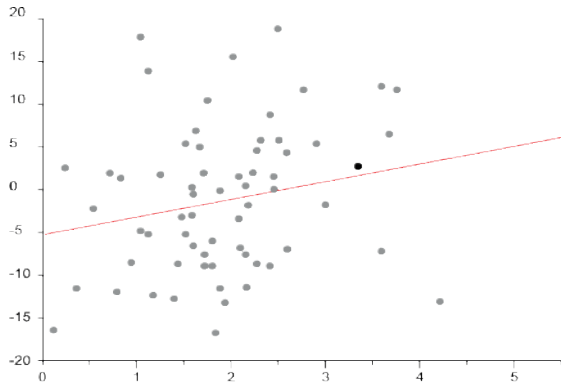
السؤال الثاني: أيُّ خطوط الانحدار الآتية يُمثِّل أفضل خَطٍّ مُلائِم للتنبُّؤ بالقِيَم المستقبلية لمُتغيِّر ما؟ أُبرِّر إجابتي.



(أ)



(ب)



(ج)

الدرس الخامس

الفرضيات في تحليل البيانات

(Hypotheses in Data Analysis)

الفكرة الرئيسة:

سأتعرَّف في هذا الدرس مفهوم الفرضية، وأنواعها، والشروط الواجب التزامها عند صياغة الفرضية، إضافةً إلى تعرُّف خطوات فحصها، وآليَّة التحقُّق من صِحَّتها. كذلك سأتعلَّم كيف أُقيِّم كفاءة النماذج وأساليب المحاكاة في اختبار فرضية مُحدَّدة ودعمها، ثمَّ أتعرَّف بعض مكتبات برمجية بايثون التي تُستخدَم في فحص الفرضيات ودعم التحليل العلمي.

المفاهيم والمصطلحات:

الفرضية (Hypothesis)، الفرضية الصفرية (Null Hypothesis)، الفرضية البديلة (Alternative Hypothesis)، مستوى الدلالة المعنوية (Significance Level).

نتاجات التعلُّم (Learning Outcomes):

- أُوضِّح مفهوم الفرضية، وأُبيِّن أنواعها.
- أُميِّز الفرضية الصفرية من الفرضية البديلة.
- أكتب صياغة كلٍّ من الفرضية الصفرية، والفرضية البديلة.
- أُوضِّح خطوات اختبار الفرضيات وفحصها.
- أُفسِّر دور مستوى الدلالة في الحكم على صِحَّة الفرضية أو عدم صِحَّتها.
- أضع فرضية عن مشكلة من البيئة المحيطة، وأجمع بياناتها، وأتحقَّق من صِحَّتها أو عدم صِحَّتها.

يُعَدُّ الفضول جزءًا أصيلًا من الطبيعة البشرية منذ القِدَم، وهو يُمثِّل إحدى الأدوات المحورية التي ساعدت الإنسان على التطوُّر والاكتشاف. وفي ظلِّ تعاقب الحضارات وتتابع الابتكارات والاختراعات، فإنَّ الفضول بلغ أَوْجَهُ؛ ما دفع الإنسان إلى طرح مزيد من الأسئلة، وحفَّزه إلى البحث المستمر عن تفسير للظواهر المحيطة به، ودعاه إلى وضع تنبُّؤات لم يتمَّ تأكيد صِحَّتها أو خطئها بعدُ. يُطلَق على هذه التنبُّؤات اسم الفرضيات، فما المقصود بالفرضيات؟ وما أنواعها؟ وما الشروط الواجب التزامها عند صياغة الفرضيات؟ وما الطرائق المُتَّبَعة في اختبارها وفحصها؟

نشاط تمهيدي

أفترض أنَّني باحث في علم النباتات، وأنَّني رصدْتُ الملاحظتين الآتيتين:

1- وجود نبات مُعيَّن ينمو بشكل أسرع عندما يُروى في الصباح.

2- رَيُّ النباتات بعد الظهر يجعلها تبدو أقلَّ نضارة.

ما الفرضيات التي يُمكِن أنْ أبنيها على هاتين الملاحظتين؟ كيف يُمكِن إثبات هذه الفرضيات أو نفيها؟

أُدوِّن إجاباتي، ثمَّ أُشارِكها مع الزملاء/ الزميلات في الصف.

الفرضية (Hypothesis):

تُعرَّف الفرضية بأنَّها عبارة مُقترَحة أو ادِّعاء أو تصوُّر أو تخمين مُؤقَّت قابل للاختبار بخصوص طبيعة العلاقات بين المُتغيِّرات التي تخضع للفحص بُغْيَةَ التحقُّق من صِحَّتها أو خطئها. تُستخدَم الفرضيات عادةً في اختبار علاقة مُتغيِّر بمُتغيِّر آخر، أو أثر مُتغيِّر في مُتغيِّر آخر؛ ما يعني أنَّها تقوم على فحص العلاقة بين مُتغيِّرين أو أكثر.

ضوابط كتابة الفرضيات:

يُشترَط عند كتابة الفرضيات وصياغتها الالتزام بجملة من الضوابط، هي:

1. **الوضوح ودِقَّة الصياغة:** يجب أنْ تُكتَب الفرضية بلغة واضحة غير مُبهَمة.
2. **القابلية للاختبار:** يجب أنْ تكون الفرضية قابلة للتحقُّق عن طريق التجربة أو البيانات.
3. **وجود علاقة بين مُتغيِّرين:** يجب أنْ تتضمَّن الفرضية اختبار العلاقة بين مُتغيِّرين أو أكثر.
4. **تحديد المدى:** يجب أنْ تكون الفرضية مُحدَّدة بموقف مُعيَّن أو بظروف مُعيَّنة.

أمثلة على بعض الفرضيات:

1. وجود علاقة بين امتلاك الطلبة أجهزة حاسوب مُتَّصِلة بشبكة الإنترنت وتحصيلهم الدراسي.
2. عدم وجود فَرْق بين مُتوسِّط تحصيل الطلبة في مبحث الرياضيات ومُتوسِّط تحصيلهم في مبحث اللغة العربية.
3. عدم وجود علاقة بين مستوى تعليم الأُمِّ ومستوى تحصيل الطلبة في الصف الثاني الثانوي.
4. وجود أثر لاستخدام استراتيجية العمل الجماعي في مُتوسِّط تحصيل الطلبة في مبحث اللغة العربية.
5. عدم وجود علاقة بين طول الفرد ومُعدَّل ذكائه.
6. عدم وجود علاقة بين عدد ساعات النوم وأداء الموظفين في العمل.

نشاط فردي

أذكر أمثلة على بعض الفرضيات، وأُراعي استيفاءها للضوابط المذكورة آنفًا في الفرضية، ثمَّ أُشارِكها مع الزملاء/ الزميلات في الصف.

أنواع الفرضيات:

1. **الفرضية الصفرية أو فرضية العدم** (Null Hypothesis): يُرمَز إلى هذه الفرضية بالرمز H0، وهي تُكتَب عادةً بصيغة النفي، وتشير إلى عدم وجود فَرْق أو علاقة أو تأثير بين المُتغيِّرات. في البداية، يُفترَض صِحَّة الفرضية الصفرية إلى أنْ يَثبت العكس عن طريق البيانات التجريبية. وفي حال تبيَّن وجود علاقة بين المُتغيِّرات، فإنَّ ذلك يُعَدُّ نتيجة المصادفة فقط، لا بسبب وجود علاقة سببية حقيقية بينها.

من الأمثلة على الفرضية الصفرية:

- عدم وجود علاقة بين عدد ساعات الدراسة والعلامة في الامتحان.
- عدم وجود علاقة بين العدوانية والإحباط.
- عدم وجود فَرْق بين مُتوسِّط تحصيل الطلبة في مبحث الرياضيات ومُتوسِّط تحصيلهم في مبحث اللغة العربية.
- عدم وجود علاقة بين التدخين والإصابة بالسكتات القلبية.
- عدم وجود أثر لطريقة العلاج بالدواء (س) في القلب.

2. **الفرضية البديلة** (Alternative Hypothesis): يُرمَز إلى هذه الفرضية بالرمز H1، وهي تُكتَب عادةً بصيغة الإثبات خلافًا للفرضية الصفرية.

من الأمثلة على الفرضية البديلة:

- وجود علاقة بين عدد ساعات الدراسة والعلامة في الامتحان.
- وجود فَرْق بين مُتوسِّط عمر الإنسان وانتظامه في ممارسة الرياضة.
- وجود علاقة بين مستوى التحصيل في مبحث الفيزياء ومُعدَّل الذكاء لدى الطلبة.
- وجود أثر لطريقة التدريس الحديثة في مستوى تحصيل الطلبة في مبحث العلوم.

نشاط جماعي

أقرأ الفرضيات الآتية، ثمَّ أعمل - بالتعاون مع أفراد مجموعتي- على تحليل صياغة كل فرضية، وتحديد إذا كانت تشير إلى فرضية صفرية أو فرضية بديلة:

1. لا توجد علاقة بين المستوى المادي للطالب ومستوى تعليم الأبناء.
2. توجد علاقة بين تناول المُثلَّجات (الآيس كريم) والنزلة الصدرية لدى الأطفال.
3. لا توجد علاقة بين عدد أيام غياب الطالب والعلامة في مبحث الرياضيات لطلبة الصف التاسع.
4. لا يوجد فَرْق بين حالة الطالب المادية ومستوى التحصيل لديه.
5. يوجد أثر لطريقة التدريس باستخدام أوراق العمل في مستوى تحصيل الطلبة.

أُناقِش الإجابات مع زملائي/ زميلاتي في المجموعة، ثمَّ أعرضها أمام أفراد المجموعات الأُخرى بهدف التوصُّل إلى استنتاجات مُشترَكة.

اختبار الفرضيات:

يُقصَد باختبار الفرضيات عملية الفحص التي تهدف إلى التحقُّق من صِحَّة الفرضية أو خطئها، والتي تُركِّز عادةً على فحص الفرضية الصفرية؛ ذلك أنَّ إثبات خطأ الفرضية أسهل من إثبات صِحَّتها.

نتائج اختبار الفرضية:

1. **قَبول الفرضية الصفرية:** يُقصَد بذلك أنَّ الفرضية الصفرية صحيحة بناءً على البيانات التي تشير إلى عدم وجود علاقة أو أثر أو فَرْق بين المُتغيِّرات موضوع الدراسة.

مثال:

لا يوجد تأثير لتناول المُثلَّجات (الآيس كريم) في حدوث النزلة الصدرية.

2. **رفض الفرضية الصفرية:** يُقصَد بذلك أنَّ الفرضية الصفرية غير صحيحة بناءً على البيانات التي تشير إلى وجود علاقة أو أثر أو فَرْق بين المُتغيِّرات موضوع الدراسة.

مثال: تجربة توضيحية لإنشاء الفرضية واختبارها تجريبيًا.

نفَّذ أحد الطلبة تجربة نباتية، ثمَّ كتب فرضيته على النحو الآتي:

- **الفرضية الصفرية (H0):** لا يوجد أثر لنوع السماد (س) في طول النبات.
- **الفرضية البديلة (H1):** يوجد أثر لنوع السماد (س) في طول النبات.

إجراءات تنفيذ التجربة:

1. **نوع النبات:** اختار الطالب عيِّنتين من نفس نوع النبات.
2. **طبيعة الزراعة:** زرع الطالب العيِّنتين في نفس التربة، وعرَّضهما لنفس الظروف البيئية؛ من: إضاءة، وماء، وحرارة.
3. **السماد:** وضع الطالب السماد (س) لإحدى العيِّنتين، ولم يضعه للعيِّنة الأُخرى.
4. **المراقبة اليومية:** راقب الطالب نمو النبات في كلتا العيِّنتين يوميًّا على مدار أسبوعين، ثمَّ دوَّن النتائج في جدول.

تحليل النتائج (المقارنة بين عيِّنتي النبات من حيث الطول):

1- إذا كان طول النبات الذي أُعطِي السماد (س) أطول بشكل ملحوظ من النبات الذي لم يُعطَ السماد، فإنَّ الفرضية الصفرية تُرفَض خلافًا للفرضية البديلة، ويُستنتَج أنَّ للسماد (س) تأثيرًا في طول النبات.

2- إذا لم يوجد فَرْق ملحوظ في الطول بين النباتين، يتمُّ قَبول الفرضية الصفرية، ويُستنتَج أنَّ السماد (س) لا يُؤثِّر في طول النبات.

خطوات اختبار الفرضية:

1. **صياغة الفرضية (الصفرية، والبديلة):** يجب قبل اختبار أيَّة فرضية تحديدُ ما يُراد فحصه، وصياغة كلٍّ من الفرضية الصفرية والفرضية البديلة. ففي المثال السابق، يُمكِن صياغة كلتا الفرضيتين (الصفرية، والبديلة) على النحو الآتي:
   أ. **الفرضية الصفرية (H0):** لن ينمو النبات أطول عند منحه نوعًا مُعيَّنًا من السماد (س).
   ب. **الفرضية البديلة (H1):** سينمو النبات، ويزداد طوله عند منحه نوعًا مُعيَّنًا من السماد (س).
2. **تحديد مستوى الأهمية، أو مستوى الدلالة المعنوية (Significant Level):** يتمثَّل مستوى الدلالة المعنوية – الذي اختير قبل اختبار الفرضية– في رقم يتمُّ اختياره معيارًا يُعتمَد عليه في تحديد صِحَّة الفرضية من خطئها. وفي معظم الحالات، يُختار مستوى الدلالة المعنوية ليكون الرقم (0.05)، ويُرمَز إليه بالرمز α، ويُقرَأ: ألفا. ومن ثَمَّ يُقال: مستوى الدلالة المعنوية = 0.05، أو (α= 0.05).

إضاءة

في بعض الحالات، يُختار مستوى الدلالة المعنوية ليكون α= 0.01، فيصبح المعيار عندئذٍ أكثر دِقَّة.

3. **جمع البيانات:** تُجمَع البيانات المُتعلِّقة بالفرضية بشكل مُنظَّم ودقيق عن طريق التجارب أو المسوحات أو الدراسات.
4. **اختبار الفرضية:** تُستخدَم اختبارات عديدة في فحص الفرضيات، مثل:

- **اختبار t:** يتيح هذا الاختبار المقارنة بين مُتوسِّطين، مثل: مُتوسِّط نمو النبات مع السماد، ومن دونه.
- **اختبار Z:** يتيح هذا الاختبار تحليل العيِّنات الكبيرة.
- **اختبار مُربَّع كاي ($\chi 2$):** يتيح هذا الاختبار فحص العلاقة بين المُتغيِّرات الفئوية (اختبار لا معلمي).

5. **تحليل النتيجة:** تكون نتيجة هذه الاختبارات عادةً قيمة مستوى الدلالة المعنوية (α)، ويتمُّ مقارنتها بمستوى الدلالة المعنوية الذي اختير أداة للقياس. وبناءً على نتيجة المقارنة، يُمكِن الحكم على الفرضية الصفرية بأنَّها صحيحة أو غير صحيحة:

أ. إذا كانت القيمة الناتجة أقلَّ من α، فإنَّ الفرضية الصفرية تُرفَض لمصلحة الفرضية البديلة.
مستوى الدلالة الناتج من اختبار فحص الفرضية < 0.05
إذن، الفرضية الصفرية غير صحيحة.

ب. إذا كانت القيمة الناتجة أكبر من أو تساوي α، فإنَّه يتمُّ قَبول الفرضية الصفرية.
مستوى الدلالة الناتج من اختبار فحص الفرضية >= 0.05
إذن، الفرضية الصفرية صحيحة.

مثال:

- **صياغة كلٍّ من الفرضية الصفرية والفرضية البديلة:**
- **الفرضية الصفرية:** لا توجد علاقة بين عدد أيام غياب الطلبة ومستوى التحصيل لديهم.
- **الفرضية البديلة:** توجد علاقة بين عدد أيام غياب الطلبة ومستوى التحصيل لديهم.
- **اختيار مستوى الدلالة المعنوية الذي يُراد استخدامه معيارًا، وهو: (α= 0.05)**
- جمع البيانات والمعلومات عن الطلبة في ما يتعلَّق بعدد أيام غيابهم ومستوى التحصيل لديهم.
- تحليل الفرضية باستخدام أحد اختبارات فحص الفرضيات، وملاحظة قيمة الدلالة المعنوية الناتجة من الاختبار؛ لمقارنة قيمة الدلالة المعنوية الناتجة من الاختبار بمستوى الدلالة المعنوية الذي اختير معيارًا للقياس عليه.
- **تحليل النتيجة، وذلك بالحكم – بناءً على هذه الأرقام– على صِحَّة الفرضية الصفرية أو عدم صِحَّتها كما يأتي:**
  - افتراض أنَّ مستوى الدلالة الناتج من اختبار فحص الفرضية في المثال المذكور آنفًا هو: α= 0.0003
  - مقارنة α= 0.0003 بالمعيار المختار لمستوى الدلالة المعنوية، وهو: α= 0.05
  - ملاحظة أنَّ مستوى الدلالة الناتج من اختبار فحص المعنوية أقلُّ من مستوى الدلالة المختار، وهو (0.05):

$$0.05 > 0.0003$$

إذن، يُحكَم على الفرضية الصفرية – في هذه الحالة– بعدم صِحَّتها؛ ما يعني وجود علاقة بين عدد أيام غياب الطلبة ومستوى التحصيل لديهم.

أُناقِش

في المثال السابق، أفترض أنَّ مستوى الدلالة المعنوية الناتج من اختبار الفرضية هو: $\alpha = 0.6$، هل تُعَدُّ الفرضية الصفرية صحيحة أم غير صحيحة في هذه الحالة؟ أُبرِّر إجابتي.

نشاط فردي

اختبر باحث فرضية صفرية بالقول: "لا توجد علاقة بين مستوى تحصيل الطلبة في مبحث الرياضيات ومُعدَّله العام".

اختبر الباحث هذه الفرضية باستخدام أحد اختبارات الفرضيات بعد جمع البيانات، وكان مستوى الدلالة الناتج من اختبار الفرضية هو (0.00004) على أساس أنَّ مستوى الدلالة المعنوية هو (0.05)، فهل الفرضية الصفرية صحيحة أم غير صحيحة في هذه الحالة؟ أُبرِّر إجابتي، ثمَّ أُناقِش الزملاء/ الزميلات في ما أتوصَّل إليه من نتائج.

المواطنة الرقمية:

- **الشفافية** (Transparency): أُراعي أنْ تكون الفرضيات واضحة ومفهومة للجميع، وأُوضِّح الأساس الذي بُنِيت عليه (مثل: البيانات، والملاحظات). كذلك أُشارِك الآخرين في طريقة صياغة الفرضية واختبارها؛ تعزيزًا للمصداقية.
- **المسؤولية الاجتماعية** (Social Responsibility): أتأكَّد أنَّ الفرضيات التي أختبرها أو أستنتجها لا تُفْضي إلى تأثيرات سلبية على الأفراد أو المجتمع. كذلك أستخدم استنتاجات الفرضيات في تحسين الخدمات أو تعزيز التوعية بدلًا من التضليل.

أُقيِّم تعلُّمي

المعرفة: أستخدم ما تعلَّمْتُه من معارف في هذا الدرس للإجابة عن الأسئلة الآتية:

السؤال الأوَّل: أُوضِّح المقصود بالمفاهيم الآتية:

1. الفرضية (Hypothesis).
2. الفرضية الصفرية (Null Hypothesis).
3. مستوى الدلالة (Significant Level).

السؤال الثاني: أُصنِّف الفرضيات الآتية إلى صفرية وبديلة:

1. توجد علاقة بين عدد ساعات الدراسة ومستوى التحصيل لدى الطلبة.
2. لا يوجد أثر لكَمِّية السماد في كَمِّية الإنتاج الزراعي لمحصول الموز.
3. توجد علاقة بين الإدمان على التدخين وأمراض القلب.
4. لا توجد علاقة بين مستوى الفقر وحدوث ظاهرة الطلاق.

السؤال الثالث: أذكر بعض الاختبارات التي تُستخدَم في فحص الفرضيات.

المهارات: أُوظِّف مهارات التفكير الناقد والبحث الرقمي والتواصل في الإجابة عن السؤالين الآتيين:

السؤال الأوَّل: أدرس الفرضية الآتية، ثمَّ أُجيب عمّا يليها:

"اختبر أحد الأطبّاء فرضيته الصفرية بعد جمع البيانات، قائلًا: لا توجد علاقة بين شرب المشروبات السكرية يوميًّا وزيادة الوزن. وبعد تطبيق الطبيب إحدى طرائق اختبار الفرضيات، توصَّل إلى نتيجة مفادها أنَّ مستوى الدلالة الناتجة من الاختبار هو: $\alpha = 0.00002$".

ما الحكم على ناتج اختبار هذه الفرضية على أساس أنَّ مستوى الدلالة المعنوية هو (0.05)؟

السؤال الثاني: أُصحِّح الخطأ في كلٍّ من الجمل الآتية:

1. يجب فحص الفرضية البديلة واختبارها بوجه عام.
2. تُجمَع البيانات بعد فحص الفرضية واختبارها.
3. تُكتَب الفرضية الصفرية بصيغة النفي.

أسئلة الوحدة

السؤال الأوَّل:

أُعرِّف المصطلحات الآتية:

1. البيانات.
2. الإحصاءات الوصفية.
3. الفرضية.
4. مستوى الدلالة المعنوية.
5. علاقة السبب والنتيجة.

السؤال الثاني:

أُحدِّد أنواع البيانات في كلٍّ ممّا يأتي:

البيانات	أنواعها
100	
“Data Analysis”	
30000.5	
[20,32,45,90]	
(“Ali”, “ Ahmad”, “Sara”)	
{36 : “**name” :** “ John” , “ age”}	

السؤال الثالث:

في ما يأتي مجموعة من الدوال التي توجد في مكتبة (Matplotlib)، والتي تساعد المُبرمِجين على تنسيق المُخطَّطات البيانية المختلفة:

الدالَّة	الوظيفة
`Plot()`	
`Bar ()`	
`show ()`	
`title ()`	
`xlabel ()`	
`ylabel ()`	
`legend ()`	

أكتب بجانب كل دالَّة وظيفتها.

السؤال الرابع:

أُوضِّح الفَرْق بين الفرضية الصفرية والفرضية البديلة، ثمَّ أذكر مثالًا واحدًا على كلٍّ منهما.

السؤال الخامس:

أُوضِّح خطوات اختبار الفرضية وفحصها.

السؤال السادس:

أ. أستعين بالجدول الآتي لإنشاء مُخطَّط بياني باستخدام برمجية بايثون يُبيِّن العلاقة بين عدد ساعات الدراسة وعلامات الامتحان، وأُوضِّح الاتِّجاه العام في الرسم.

ساعات الدراسة	علامات الامتحان
3	60
2	53
5	74
12	96
5	73
9	82
10	90

ب. أستخدم مكتبة (Matplotlib) في إنشاء رسم بياني خَطِّي يُبيِّن العلاقة بين طول الفرد ووزنه بناءً على البيانات الآتية:

الطول (سم)	الوزن (كغ)
155	50
160	76
165	67
168	60
170	65
158	64
163	60

السؤال السابع:

إذا كانت معادلة الانحدار الناتجة من تحليل بيانات هي:

$$Y = 5 * X + 20$$

حيث: y تُمثِّل العلامات، و x تُمثِّل عدد ساعات الدراسة:

أ. فأَجِد العلامة المُتوقَّعة لطالب يَدْرس (4) ساعات يوميًّا.

ب. ماذا تُمثِّل القيمة (20) في المعادلة السابقة؟

السؤال الثامن:

يُظهِر الجدول الآتي البيانـــات الخاصة بعدد الزيارات اليومية لمتجر إلكتروني، والإيرادات اليومية (بالدولار):

عدد الزيارات اليومية لمتجر إلكتروني:	100	160	230	400	350	500
الإيرادات اليومية (بالدولار):	1000	1500	2000	2400	2600	3200

أ. هل توجد علاقة خَطِّية بين عدد الزيارات اليومية والإيرادات؟

ب. أقترح معادلة انحدار بسيطة بناءً على النمط المُلاحَظ.

تقويم ذاتي (Self – Checklist)

بعد دراستي هذه الوحدة، أقرأ الفقرات الواردة في الجدول الآتي، ثمَّ أضع إشارة (✔) في العمود المناسب:

مُؤشِّرات الأداء	نعم	لا	لستُ مُتأكِّدًا
1- أُعرِّف البيانات، وأُبيِّن أنواعها.	○	○	○
2- أُبيِّن أهمية تحليل البيانات.	○	○	○
3- أُوضِّح أهمَّ طرائق تحليل البيانات.	○	○	○
4- أُوضِّح طرائق جمع البيانات واستخداماتها.	○	○	○
5- أُوضِّح مراحل استكشاف البيانات.	○	○	○
6- أُميِّز بين أدوات جمع البيانات وتحليلها واستكشافها.	○	○	○
7- أستخدم الأدوات والتقنيات الخاصة بتحليل البيانات في تحديد اتِّجاه نزعة المركزية.	○	○	○
8- أُفاضِل بين الأدوات والتقنيات المُتوافِرة التي تُستخدَم في تحليل البيانات اعتمادًا على المشكلة التي يُراد حَلُّها.	○	○	○
9- أُثبِّت المكتبات الخارجية الخاصة بتحليل البيانات في برمجية بايثون، مثل؛ (Panadas)، و(Matplotlib).	○	○	○
10- أستورد ملفات البيانات (مثل؛ (CSV)، و(Excel)) إلى برمجية بايثون (Python) لتحليلها.	○	○	○
11- أُجري عمليات تحليلية على البيانات باستخدام مكتبة (Pandas) في برمجية بايثون (Python).	○	○	○
12- أُفسِّر نتائج التحليل الإحصائي الوصفي من شاشة مُخرَجات برمجية بايثون (Python).	○	○	○

مُؤشِّرات الأداء	نعم	لا	لسْتُ مُتأكِّدًا
13- أعرض البيانات بطرائق مختلفة باستخدام مكتبة (Matplotlib) في برمجية بايثون (Python).	○	○	○
14- أُحدِّد الأنماط في الرسوم البيانية في برمجية بايثون (Python) باستخدام مزايا بصرية.	○	○	○
15- أُحلِّل الرسوم البيانية لاستكشاف الأنماط واستخلاص استنتاجات تدعم الفهم الأوَّلي للبيانات.	○	○	○
16- أُوضِّح مفهوم علاقة السبب والنتيجة.	○	○	○
17- أُفسِّر علاقة السبب والنتيجة.	○	○	○
18- أستنتج علاقة السبب والنتيجة من البيانات.	○	○	○
19- أتنبَّأ بالنتيجة عن طريق السبب، وأتنبَّأ بالسبب عن طريق النتيجة.	○	○	○
20- أُوضِّح مفهوم الفرضية، وأُبيِّن أنواعها.	○	○	○
21- أُميِّز الفرضية الصفرية من الفرضية البديلة.	○	○	○
22- أكتب صياغة كلٍّ من الفرضية الصفرية، والفرضية البديلة.	○	○	○
23- أُوضِّح خطوات اختبار الفرضيات وفحصها.	○	○	○
24- أُفسِّر دور مستوى الدلالة في الحكم على صِحَّة الفرضية أو عدم صِحَّتها.	○	○	○
25- أضع فرضية عن مشكلة من البيئة المحيطة، وأجمع بياناتها، وأتحقَّق من صِحَّتها أو عدم صِحَّتها.	○	○	○

تعليمات للمراجعة والتحسين:

إذا اخترْتُ (لا) أو (لسْتُ مُتأكِّدًا) لأيٍّ من الفقرات السابقة، فأَتَّبِع الخطوات الآتية لتجنُّب ذلك:

- أُراجِع المادة الدراسية؛ بأنْ أُعيد قراءة المحتوى المُتعلِّق بالمعيار.
- أطلب المساعدة؛ بأنْ أُناقِش مُعلِّمي/ مُعلِّمتي أو زملائي/ زميلاتي في ما تعذَّر عليَّ فهمه.
- أستخدم مراجع إضافية؛ بأنْ أبحث عن مراجع أُخرى مثل الكتب، أو أستعين بالمواقع الإلكترونية الموثوقة التي تُقدِّم شرحًا وافيًا للموضوعات التي أَجِد صعوبة في فهمها.

تأمُّلات ذاتية

عزيزي الطالب، عزيزتي الطالبة:

التأمُّلات الذاتية هي فرصة لتقييم عملية التعلُّم، وفهم التحدِّيات، وتطوير اســـتراتيجيات لتحسين عملية التعلُّم مستقبلًا.

أملأ الفراغ في ما يأتي بالأفكار والتأمُّلات الشخصية التي يُمكِن بها تحقيق أفضل استفادة من التجربة التعليمية:

تعلَّمْتُ في هذه الوحدة:

يُمكِنني أنْ أُطبِّق ما تعلَّمْتُه في:

الصعوبات التي واجهْتُها أثناء عملية التعلُّم:

ذلَّلْتُ هذه الصعوبات عن طريق:

يُمكِنني مستقبلًا تحسين:

تمَّ بحمد الله تعالى