آخر تحديث: November 1, 2026

🗺️ خريطة طريق لأربع سنوات · السنة 1, الربيع

عن هذه الدورة

يتضمن هذا المقرر المجاني 9 وحدات و4 مختبرات. هذه الأعداد الحقيقية وليست منهجًا أطول جرى اختصاره.

التحقق هو رقم أو نتيجة برمجية في صفحة مقرر يعيد سكربت حسابها قبل النشر. يجب أن يطابق الرقم المعروض الرقم المعاد حسابه.

هذا مخطط دورة مجانية للتعلّم الذاتي كتبته Saving Planets. وهو ليس مقررًا جامعيًا، ولا يمنح ساعات معتمدة، ولا تشارك فيه أي مؤسسة تعليمية.

يتبع نطاقه مقررات البرمجة في السنة الأولى مثل MIT 6.100L (سابقًا 6.0001)، وCS50 في Harvard، وCS106A في Stanford، وCS 1371 في Georgia Tech للمهندسين. نضع روابط لمواد مجانية في كل مكان. لا تتبنى أي من هذه المؤسسات هذه الصفحة.

ما نضيفه: كل مثال هو كود بايثون حقيقي يشغّله نص البناء لدينا قبل نشر الصفحة، ويجب أن يطابق ناتجه المطبوع ما تراه هنا تمامًا. تأتي البيانات من مصادر عامة حقيقية: سجل CO_2 في ماونا لوا من NOAA، وتحليل درجات الحرارة GISTEMP من NASA، ومقاييس الأنهار من USGS.

لا تحتاج إلى تثبيت أي شيء: استخدم Google Colab أو JupyterLite، الذي يشغّل بايثون داخل متصفحك باستخدام Pyodide. يُحفظ تقدمك في هذا المتصفح فقط.

كيف تساعد هذه الدورة في إنقاذ الكوكب

علوم المناخ والبيئة تعمل بالكود. منحنى CO_2 وسجلات درجات الحرارة العالمية وتوقعات الأنهار كلها تبنيها برامج تقرأ بيانات عامة، ويمكنك قراءة الملفات نفسها.

تنتهي كل وحدة بقسم قصير موثق المصادر يبيّن كيف تساعد أفكارها الناس والكوكب.

للمعلمين

يمكنك تدريس هذا المقرر، أو أن يتابعه الطالب وحده. إنه مجاني. Saving Planets لا تمنح درجة ولا ساعات معتمدة.

- هذه الصفحة هي المقرر كاملًا: 9 وحدات و4 مختبرات (ENGR 130).
- خطّط لخمسة عشر أسبوعًا، نحو 6 إلى 9 ساعات في الأسبوع، بما في ذلك أسابيع المختبر.
- اتبع جدول الخمسة عشر أسبوعًا. اقرأ الوحدة، حلّ التمارين على الورق، ثم افتح الإجابة. في أسبوع المختبر، نفّذ المختبر أو نسخته المنزلية.
- التحقق هو رقم أو نتيجة برمجية في الصفحة يعيد سكربت حسابها قبل النشر. يجب أن يطابق الرقم الظاهر الرقم المعاد حسابه.

- استخدم طباعة هذه الصفحة، أو نزّل دليل الدراسة PDF. ورقة القوانين قرب النهاية.
- اسأل: كيف تساعد هذه الوحدة الناس والكوكب؟ في كل وحدة مربع أخضر فيه حقيقة مع مصدرها.
- انسخ نموذج دليل المعلم واكتب اسم صفك والوقت المتاح والأسئلة التي ستطرحها. افتح نموذج دليل المعلم

قبل أن تبدأ

المتطلبات السابقة

- الجبر في المرحلة الثانوية.
- لا تلزم خبرة في البرمجة.
- الفيزياء I والتفاضل والتكامل I يساعدان في وحدة المحاكاة لكنهما غير مطلوبين.

كيف تذاكر

- اكتب كل مثال بنفسك بدل لصقه؛ فأصابعك تتعلم الصياغة.
- توقّع الناتج قبل تشغيل الكود.
- عندما تتعثر، اقرأ رسالة الخطأ من الأسفل إلى الأعلى.
- غيّر شيئًا واحدًا في كل مرة وأعد التشغيل.
- حل كل تمرين قبل فتح الإجابة.
- أعد الاختبارات القصيرة بعد بضعة أيام. التدرّب المتباعد أفضل من جلسة طويلة واحدة.

ماذا تحتاج

- حاسوب أو جهاز لوحي فيه متصفح ويب حديث.
- حساب Google مجاني لـ Colab، أو JupyterLite دون أي حساب.
- اختياري: Python 3 مثبت محليًا مع NumPy وMatplotlib.

جدول مقترح من 15 أسبوعًا

نحو 6 إلى 9 ساعات أسبوعيًا: قراءة وتمارين، وفي أسابيع المختبر، المختبر.

الأسبوع	الوحدة	الموضوعات	القراءة
1	البداية: المتغيرات والأنواع والتعابير	الدفاتر، والمتغيرات، والأنواع، والتعابير	درس بايثون التعليمي §3
2	التحكم في التدفق: الشروط والحلقات	الشروط والحلقات	درس بايثون التعليمي §4
3	الدوال	الدوال	MIT 6.100L
4	القوائم والقواميس والسلاسل النصية	القوائم، والقواميس، والسلاسل النصية	درس بايثون التعليمي §5
5	الملفات وبيانات CSV	الملفات وCSV	توثيق الوحدة csv
6	الملفات وبيانات CSV	بيانات حقيقية بها أخطاء؛ المختبر A	NOAA GML

الأسبوع	الوحدة	الموضوعات	القراءة
7	NumPy وMatplotlib	مصفوفات NumPy؛ تمارين منتصف الفصل	أساسيات NumPy
8	NumPy وMatplotlib	Matplotlib؛ المختبر B	Matplotlib pyplot
9	تصحيح الأخطاء والاختبار	تصحيح الأخطاء والاختبار؛ المختبر C	توثيق pytest
10	الخوارزميات والتعقيد	البحث والتعقيد	MIT 6.100L
11	الخوارزميات والتعقيد	التنصيف والطرق العددية	MIT 6.100L
12	المحاكاة العددية	محاكاة المقذوفات	الفيزياء I
13	المحاكاة العددية	المدارات والتجمعات السكانية	التفاضل والتكامل 2
14	المحاكاة العددية	السحب والتقارب؛ المختبر D	مدخل إلى هندسة الطيران والفضاء
15	المراجعة	مراجعة ومجموعة تمارين نهائية	كل الوحدات

تقدّمك

ضع علامة على الوحدات والمختبرات عند إنهائها، وخض الاختبارات. يُحفظ التقدم ونتائج الاختبارات في هذا المتصفح فقط، ولا تُرسل إلينا أبدًا.

الوحدات

البداية: المتغيرات والأنواع والتعابير

شغّل أول كود بايثون لك في دفتر داخل المتصفح، وخزّن القيم في متغيرات، واحسب بالأرقام والنصوص.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- افتح دفتر Colab أو JupyterLite وشغّل الخلايا.
- استخدم المتغيرات والأنواع int وfloat وstr وbool.
- استخدم العوامل الحسابية، بما فيها // و% و**.
- نسّق الناتج باستخدام f-strings.

المفاهيم الأساسية

- البرنامج سلسلة من التعليمات. ويتيح لك الدفتر تشغيلها خلية بعد خلية ورؤية النتائج فورًا.
- المتغير اسم مرتبط بقيمة: mass_kg = 1100. اختر أسماء تقول ما الشيء وبأي وحدات.
- لكل قيمة نوع. int (الأعداد الصحيحة)، وfloat (الأعداد العشرية)، وstr (النص)، وbool (True/False) هي الأساسيات؛ وتخبرك type() بالنوع.
- / يعطي دائمًا float؛ و// قسمة صحيحة؛ و% الباقي؛ و** القوة.
- الأعداد float تقريبات ثنائية: 0.2 + 0.1 لا يساوي 0.3 تمامًا، لذا قارن قيم float بهامش تسامح.

المعادلات الأساسية

$$W = mg$$

W=mg

المثال 1.1 يحسب وزنًا

$$\text{change} = \frac{x_{new} - x_{old}}{x_{old}}$$

change=_new-x_oldx_old

التغير النسبي (المثال 1.2)

أمثلة محلولة خطوة بخطوة

مثال 1.1 احسب وزن طائرة كتلتها 1100 kg وتحقق من الأنواع.

```
mass_kg = 1100
g = 9.81
weight = mass_kg * g
```

```
print(weight)
print(type(mass_kg), type(weight))
```

1. mass_kg من النوع int؛ وضربه في g من النوع float يعطي float.

الناتج

```
10791.0
<class 'int'> <class 'float'>
```

الإجابة: 10791 N، من النوع float.

مثال 1.2 كم ارتفع CO_2 في ماونا لوا من المتوسط السنوي لعام 1959 إلى متوسط 2025 (NOAA)؟

```
ppm_1959 = 315.98
ppm_2025 = 427.35
rise = ppm_2025 - ppm_1959
print(f"CO2 rose {rise:.2f} ppm, or {rise / ppm_1959:.1%}")
```

1. اطرح، ثم اقسم على القيمة الابتدائية.
2. تنسّق f-string القيمة بـ .2f: (منزلتان عشريتان، 2) و.1%: (نسبة مئوية).

الناتج

```
CO2 rose 111.37 ppm, or 35.2%
```

الإجابة: 111.37 ppm، أي ارتفاع بنسبة 35.2%.

مثال 1.3 استكشف العوامل ومفاجآت float.

```
print(7 / 2, 7 // 2, 7 % 2, 2 ** 10)
print(0.1 + 0.2 == 0.3, abs((0.1 + 0.2) - 0.3) < 1e-9)
print(int("42") + 1, str(42) + "1")
```

1. 7/2 = 3.5, 7//2 = 3, 7%2 = 1.
2. 0.1 + 0.2 == 0.3 يعطي False؛ واختبار بهامش تسامح يعطي True.
3. int("42") يحول النص إلى عدد؛ و"1" + str(42); يضم النصوص.

الناتج

```
3.5 3 1 1024
False True
```

```
43 421
```

الإجابة: استخدم هوامش تسامح مع float وحوّل الأنواع عن قصد.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. ماذا يعطي 5 % 17؟

   2.

2. ما نوع 2.0 * 3؟

   float (6.0).

3. اكتب f-string تطبع pi بـ 3 منازل عشرية.

   3.142 ← f"{math.pi:.3f}" (بعد import math).

4. لماذا mass_kg اسم أفضل من m؟

   لأنه يذكر الكمية ووحدتها، فيمنع أخطاء الوحدات.

مفاهيم خاطئة شائعة

خرافة: = تعني «يساوي».
في الحقيقة: في بايثون = تُسند؛ و== تقارن.

خرافة: 0.1 + 0.2 == 0.3 يعطي True.
في الحقيقة: يعطي False بسبب التقريب الثنائي.

خرافة: 5 + "5" يعطي 10.
في الحقيقة: إنه TypeError؛ حوّل باستخدام int("5").

بطاقات تعليمية

المتغير	int / float
اسم مرتبط بقيمة.	عدد صحيح / عدد عشري.
str	**bool**
نص.	True أو False.

f-string	**//, %**
التنسيق f"{x:.2f}".	القسمة الصحيحة، والباقي.

اختبار ذاتي

1. 7 // 2 =

3.5

3

4

1

3. القسمة الصحيحة.

2. type(2.0) هو...

int

float

str

bool

float. فيه فاصلة عشرية.

3. 2 ** 10 =

20

100

1024

12

1024. القوة.

4. الإسناد في بايثون يستخدم...

==

=

=only :

<-

=. = تربط اسمًا.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

الأرقام في المثال 1.2 مأخوذة مباشرة من مختبر الرصد العالمي (GML) التابع لـ NOAA، الذي ينشر سجل CO_2 في ماونا لوا كملف CSV مجاني. بضعة أسطر من بايثون تحوله إلى الحقائق الرئيسية في علم المناخ.

على Saving Planets: التفاضل والتكامل 1

المصادر: NOAA GML: Trends in atmospheric CO_2, data (https://gml.noaa.gov/ccgg/trends/data.html) · Google Colab (دفاتر مجانية) (https://colab.research.google.com/)

مصادر تعلّم مجانية

- درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/): القسمان 3 و4.
- Google Colab (دفاتر مجانية) (https://colab.research.google.com/): ابدأ دفترًا.
- JupyterLite: بايثون في متصفحك (Pyodide) (https://jupyter.org/try-jupyter/lab/): خيار دون حساب.

أنهيت هذه الوحدة

التحكم في التدفق: الشروط والحلقات

اجعل البرامج تقرر باستخدام if/elif/else وتكرر باستخدام حلقات for وwhile.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- اكتب if/elif/else مع عوامل المقارنة والعوامل المنطقية.
- كرّر على المجالات والتسلسلات باستخدام for.
- استخدم حلقات while بشرط توقف واضح.
- اجمع المجاميع والأعداد.

المفاهيم الأساسية

- المسافة البادئة تحدد الكتل في بايثون. والمعيار أربع مسافات لكل مستوى.
- المقارنات (<، >=، ==، =!) تنتج قيمًا منطقية؛ واجمعها بـ and وor وnot.
- for x in sequence تتكرر مرة لكل عنصر؛ وrange(a, b) تعطي a، a+1، ...، b−1.
- while تتكرر حتى يصبح شرطها False. تأكد أن شيئًا داخل الحلقة يغيّره، وإلا فلن تنتهي أبدًا.
- نمط المُراكِم: ابدأ المجموع من 0 وأضف داخل الحلقة.

المعادلات الأساسية

$$\sum_{n=1}^{100} n = \frac{100 \cdot 101}{2} = 5050$$

_n=1^100n=(100·101)/(2)=5050

مجموع غاوس يتحقق من المثال 2.2

$$x_{k+1} = x_k + \Delta$$

x_k+1=x_k+Δ

حلقة تزيد قيمة خطوة بخطوة

أمثلة محلولة خطوة بخطوة

مثال 2.1 صنّف شذوذ درجة الحرارة العالمية لعام 2024 من NASA.

```
anomaly = 1.28   # NASA GISTEMP, 2024, deg C vs 1951-1980
if anomaly >= 1.0:
    print("at least 1.0 C above the 1951-1980 average")
elif anomaly > 0:
```

```
    print("warmer than average")
else:
    print("at or below average")
```

1. 1.0 ≥ 1.28، لذا يُنفَّذ الفرع الأول ويُتخطى الباقي.

الناتج

```
at least 1.0 C above the 1951-1980 average
```

الإجابة: تُختبر الفروع بالترتيب؛ ولا يُنفَّذ إلا أول فرع صحيح.

مثال 2.2 اجمع الأعداد من 1 إلى 100 بحلقة.

```
total = 0
for n in range(1, 101):
    total += n
print(total)
```

1. range(1, 101) تتوقف قبل 101.
2. total += n تضيف كل عدد.

الناتج

```
5050
```

الإجابة: 5050، مطابقًا لصيغة غاوس.

مثال 2.3 لو استمر CO_2 في الارتفاع 2.5 ppm سنويًا من متوسط 2025، فمتى سيتجاوز 450 ppm؟

```
ppm = 427.35   # 2025 annual mean, Mauna Loa
year = 2025
while ppm < 450:
    ppm += 2.5
    year += 1
print(year, round(ppm, 2))
```

1. تضيف الحلقة 2.5 ppm وسنة واحدة حتى ppm ≥ 450.

الناتج

```
2035 452.35
```

الإجابة: في 2035، عند 452.35 ppm في هذا السيناريو البسيط ذي المعدل الثابت (وليس توقعًا).

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. كم مرة تُنفَّذ for i in range(3, 10, 2)؟

   4 مرات (3، 5، 7، 9).

2. اكتب حلقة تعدّ الأعداد الفردية الأصغر من 20.

   count = sum(1 for n in range(20) if n % 2) :10.

3. ما الخطأ في: while x < 10: print(x)

   x لا تتغير أبدًا، فتدور الحلقة إلى الأبد.

مفاهيم خاطئة شائعة

خرافة: range(1, 10) تشمل 10.
في الحقيقة: تتوقف عند 9.

خرافة: else بعد if إلزامية.
في الحقيقة: إنها اختيارية.

خرافة: المسافة البادئة مجرد أسلوب.
في الحقيقة: في بايثون تحدد بنية البرنامج.

بطاقات تعليمية

if / elif / else

اختر فرعًا واحدًا.

for

كرّر لكل عنصر.

while

كرّر ما دام الشرط صحيحًا.

range(a, b, s)

من a إلى b−1، بخطوة s.

المُراكِم

break / continue

x =+ total داخل حلقة.	اخرج من الحلقة / انتقل إلى الدورة التالية.

اختبار ذاتي

1. list(range(2, 6)) يساوي...

[2, 3, 4, 5, 6]

[2, 3, 4, 5]

[3, 4, 5, 6]

[2, 6]

[2, 3, 4, 5]. تتوقف قبل 6.

2. أي كلمة مفتاحية تضيف اختبارًا آخر بعد if؟

else if

elif

then

case

.elif. elif

3. تتوقف حلقة while عندما...

يصبح شرطها False

بعد 10 مرات

أبدًا

عند print

يصبح شرطها False. يُفحص الشرط في كل دورة.

4. not (True and False) يساوي...

True

False

None

خطأ

True. True and False يساوي False.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

عتبات مثل 1.0 °C أو 450 ppm هي بالضبط ما تختبره برامج الرصد بعبارات if، فتُعلِّم السجلات والتنبيهات تلقائيًا عند وصول بيانات جديدة.

على Saving Planets: الفيزياء I

المصادر: NASA GISS: GISTEMP v4 surface temperature analysis (/https://data.giss.nasa.gov/gistemp) · NOAA GML: Trends in atmospheric CO_2, data (https://gml.noaa.gov/ccgg/trends/data.html)

مصادر تعلّم مجانية

- درس بايثون التعليمي الرسمي (/https://docs.python.org/3/tutorial): القسم 4: التحكم في التدفق.
- MIT OpenCourseWare 6.100L: Introduction to CS and Programming Using Python (https://ocw.mit.edu/cour/ses/6-100l-introduction-to-cs-and-programming-using-python-fall-2022/): محاضرات عن التفرع والتكرار.

أنهيت هذه الوحدة

الدوال

اجمع المنطق في دوال قابلة لإعادة الاستخدام والاختبار، مع معاملات وقيم افتراضية وقيم مُعادة.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- عرّف الدوال باستخدام def والمعاملات وreturn.
- استخدم الوسائط الافتراضية والوسائط المسماة.
- اكتب docstrings.
- افهم النطاق المحلي.

المفاهيم الأساسية

- الدالة تسمّي جزءًا من المنطق. اكتبها مرة، واختبرها مرة، واستخدمها في كل مكان.
- تستقبل المعاملات القيم؛ وreturn تعيد النتيجة. والدالة التي لا تحتوي return تعيد None.
- الوسائط الافتراضية (cl_max=1.6) تختصر الحالات الشائعة؛ والوسائط المسماة تجعل الاستدعاءات مقروءة.
- المتغيرات المنشأة داخل دالة محلية وتختفي عند عودتها.
- الـ docstring (السلسلة النصية في السطر الأول) يوثّق ما تفعله الدالة.

المعادلات الأساسية

$$F = \frac{9}{5}C + 32$$

F=95C+32

المثال 3.1

$$V_s = \sqrt{\frac{2W}{\rho S C_{L,\max}}}$$

V_s=2Wρ SC_L,

المثال 3.2، من مدخل إلى هندسة الطيران والفضاء

أمثلة محلولة خطوة بخطوة

مثال 3.1 اكتب دالة تحويل من سلسيوس إلى فهرنهايت واختبرها.

```
def c_to_f(c):
    """Convert Celsius to Fahrenheit."""
    return c * 9 / 5 + 32
```

```
for c in (-40, 0, 1.28, 100):
    print(c, "->", round(c_to_f(c), 2))
```

1. 40− متساوية في المقياسين، وهو فحص مفيد.
2. round(..., 2) ترتّب قيم float.

الناتج

```
-40 -> -40.0
0 -> 32.0
1.28 -> 34.3
100 -> 212.0
```

الإجابة: شذوذ قدره 1.28 °C هو فرق 2.3 °F، لكن 1.28 °C كدرجة حرارة تساوي 34.3 °F؛ فالوحدات مهمة.

مثال 3.2 حوّل صيغة سرعة الانهيار إلى دالة واستخدمها عند كثافتين.

```
def stall_speed(weight_n, rho, area, cl_max=1.6):
    return (2 * weight_n / (rho * area * cl_max)) ** 0.5

print(round(stall_speed(10791, 1.227, 16.2), 1))
print(round(stall_speed(10791, 0.738, 16.2), 1))
```

1. لـ cl_max قيمة افتراضية، لذا يمكن للمستدعي حذفها.
2. 0.5 ** جذر تربيعي.

الناتج

```
26.0
33.6
```

الإجابة: 26.0 m/s عند مستوى سطح البحر و33.6 m/s عند 5000 m، مطابقًا لمقرر الطيران والفضاء.

مثال 3.3 اكتب دالة تعيد التغيرات من سنة إلى أخرى لقائمة (المتوسطات السنوية في ماونا لوا 2022–2025).

```
def growth(values):
    """Year-to-year changes of a list."""
    return [b - a for a, b in zip(values, values[1:])]

co2 = [418.53, 421.08, 424.61, 427.35]   # 2022-2025
print([round(d, 2) for d in growth(co2)])
```

1. zip تقرن كل قيمة بالتي تليها.
2. list comprehension تبني النتيجة.

الناتج

```
[2.55, 3.53, 2.74]
```

الإجابة: زيادات قدرها 2.55 و3.53 و2.74 ppm.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. اكتب area_circle(r) واختبرها عند r = 2.

   return math.pi * r**2 → 12.566....

2. ماذا تعيد الدالة التي لا تحتوي return؟

   None.

3. لماذا نختبر c_to_f عند 40−؟

   يتفق المقياسان هناك، فالإجابة المتوقعة معروفة.

مفاهيم خاطئة شائعة

خرافة: print وreturn متماثلتان.
في الحقيقة: print تعرض قيمة؛ وreturn تعيدها لاستخدامها لاحقًا.

خرافة: تغيير المعامل يغيّر متغير المستدعي.
في الحقيقة: إعادة ربط المعامل محلية (والكائنات القابلة للتغيير استثناء).

خرافة: يجب أن تكون الدوال طويلة وتفعل كل شيء.
في الحقيقة: الدوال الصغيرة ذات الغرض الواحد أسهل في الاختبار.

بطاقات تعليمية

def	**return**
تعرّف دالة.	تعيد قيمة.
المعامل	**الوسيط الافتراضي**

اسم لمُدخل.

القيمة المستخدمة عند حذفه.

Docstring

سلسلة التوثيق.

النطاق

حيث يكون الاسم مرئيًا.

اختبار ذاتي

1. تُعرَّف الدالة باستخدام...

func

def

function

lambda فقط

:def. def name(params)

2. الدالة التي لا تحتوي عبارة return تعيد...

0

None

False

خطأ

None. None ضمنيًا.

3. في def f(x, k=2)، يكون k...

إلزامي

وسيط افتراضي

عام

ثابت

وسيط افتراضي. له قيمة افتراضية.

4. المتغيرات المنشأة داخل دالة تكون...

عام

محلية

مشتركة

دائمة

محلية. النطاق المحلي.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

تُبنى أكواد المناخ والهيدرولوجيا من دوال صغيرة مختبرة كهذه، لتحويل الوحدات ومعدلات التغير والصيغ الفيزيائية، وبهذا تبقى النماذج الكبيرة جديرة بالثقة.

على Saving Planets: مقدمة في هندسة الطيران والفضاء

المصادر: NOAA GML: Trends in atmospheric CO_2, data (https://gml.noaa.gov/ccgg/trends/data.html) · درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/)

مصادر تعلّم مجانية

- درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/): القسم 4.8: تعريف الدوال.
- Harvard CS50's Introduction to Programming with Python (https://cs50.harvard.edu/python/): الدوال والمتغيرات.

أنهيت هذه الوحدة

القوائم والقواميس والسلاسل النصية

خزّن مجموعات البيانات: القوائم للتسلسلات، والقواميس للبحث، والسلاسل النصية للنص.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- استخدم الفهرسة والتقطيع وعدّل القوائم.
- استخدم len وsum وmin وmax وsorted.
- ابنِ القواميس واستعلم منها.
- قسّم السلاسل النصية وعُدّ الكلمات.

المفاهيم الأساسية

- القائمة تحمل تسلسلًا مرتبًا: flows[0] هو العنصر الأول، وflows[-1] الأخير، وflows[2:5] شريحة.
- القوائم قابلة للتغيير: append وinsert وremove وsort تغيّرها في مكانها.
- القاموس يربط المفاتيح بالقيم: gistemp[2024] → 1.28. والبحث سريع مهما كان حجمه.
- list comprehensions تبني القوائم باختصار: [x * 2 for x in xs if x > 0].
- السلاسل النصية تسلسلات من الأحرف لها توابع مثل split وstrip وlower وstartswith.

المعادلات الأساسية

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

x=1n_i=1^n x_i

متوسط قائمة (المثال 4.1)

أمثلة محلولة خطوة بخطوة

مثال 4.1 لخّص عشرة أيام من متوسط التدفق اليومي من USGS لنهر بوتوماك عند Little Falls (10–1 يونيو 2025، بالقدم المكعبة في الثانية).

```
flows = [44700, 40200, 29100, 22700, 17300, 14100, 12600, 12200, 14300, 22600]
print(len(flows), max(flows), min(flows))
print(sum(flows) / len(flows))
print(sorted(flows)[:3])
```

1. len وmax وmin تعمل على أي قائمة.
2. sum/len هو المتوسط؛ وsorted تعيد قائمة جديدة.

الناتج

```
10 44700 12200
22980.0
[12200, 12600, 14100]
```

الإجابة: المتوسط 22980 cfs؛ والمدى 12200–44700 cfs.

مثال 4.2 أوجد أدفأ سنة في قاموس لشذوذات NASA GISTEMP.

```
gistemp = {2016: 1.01, 2020: 1.01, 2023: 1.17, 2024: 1.28, 2025: 1.18}
warmest = max(gistemp, key=gistemp.get)
print(warmest, gistemp[warmest])
print([y for y, a in gistemp.items() if a > 1.1])
```

1. max مع key=gistemp.get تقارن القيم.
2. الـ comprehension تصفّي العناصر.

الناتج

```
2024 1.28
[2023, 2024, 2025]
```

الإجابة: 2024، عند 1.28 °C؛ والسنوات 2023–2025 كلها فوق 1.1 °C.

مثال 4.3 عُدّ الكلمات باستخدام قاموس.

```
words = "reduce reuse recycle reduce repair reduce".split()
counts = {}
for w in words:
    counts[w] = counts.get(w, 0) + 1
print(counts)
```

1. get(w, 0) تعيد 0 لكلمة جديدة.

الناتج

```
{'reduce': 3, 'reuse': 1, 'recycle': 1, 'repair': 1}
```

الإجابة: «reduce» تظهر 3 مرات.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. ما قيمة [10, 20, 30, 40][1:3]؟

   [20, 30].

2. كيف تضيف 2026: 1.2 إلى gistemp؟

   gistemp[2026] = 1.2.

3. ربّع كل عدد في xs باستخدام comprehension.

   [x**2 for x in xs].

مفاهيم خاطئة شائعة

خرافة: القوائم تبدأ من الفهرس 1.
في الحقيقة: تبدأ من 0.

خرافة: sorted(xs) تغيّر xs.
في الحقيقة: sorted تعيد قائمة جديدة؛ و()xs.sort تغيّرها.

خرافة: مفاتيح القاموس يمكن أن تتكرر.
في الحقيقة: كل مفتاح فريد؛ والإسناد مرة أخرى يستبدل القيمة.

بطاقات تعليمية

القائمة

تسلسل مرتب قابل للتغيير.

الفهرس / الشريحة

xs[i], xs[a:b].

القاموس

ربط مفتاح ← قيمة.

Comprehension

[f(x) for x in xs if cond].

الصف (Tuple)

تسلسل غير قابل للتغيير.

str.split

نص ← قائمة أجزاء.

اختبار ذاتي

1. xs[-1] هو...

العنصر الأول

العنصر الأخير

خطأ

الطول

العنصر الأخير. الفهارس السالبة تعدّ من النهاية.

2. len({"a": 1, "b": 2}) =

1

2

4

3

2. مفتاحان.

3. "a,b,c".split(",") تعطي...

"abc"

["a", "b", "c"]

("a","b","c")

3

["a", "b", "c"]. قائمة من السلاسل النصية.

4. القوائم...

غير قابلة للتغيير

قابلة للتغيير

قواميس

مرتبة دائمًا

قابلة للتغيير. يمكن أن تتغير في مكانها.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

تنشر USGS التدفق اليومي لآلاف مقاييس الأنهار. الأرقام العشرة في المثال 4.1 حقيقية: نبضة تدفق مرتفع في نهر بوتوماك تنحسر على مدى أسبوع، وهو نوع السجلات المستخدم في التخطيط لإمدادات المياه والفيضانات.

على Saving Planets: صفحة الأنهار.

المصادر: USGS: Potomac River near Washington, DC (Little Falls), site 01646500 (https://waterdata.usgs.gov/monitoring-location/USGS-01646500/) · NASA GISS: GISTEMP v4 surface temperature analysis (https://data.giss.nasa.gov/gistemp/)

مصادر تعلّم مجانية

- درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/): القسم 5: هياكل البيانات.
- USGS Water Services (واجهة برمجة البيانات) (https://waterservices.usgs.gov/): كيفية طلب بيانات USGS.

أنهيت هذه الوحدة

الملفات وبيانات CSV

اقرأ الملفات واكتبها، وحلّل بيانات CSV، وتعامل مع الصفوف الفوضوية دون أن يتعطل البرنامج.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- افتح الملفات النصية واقرأها واكتبها باستخدام with.
- حلّل CSV باستخدام الوحدة csv.
- حوّل السلاسل النصية إلى أعداد وتعامل مع الأخطاء باستخدام try/except.
- نزّل مجموعة بيانات حقيقية وحلّلها.

المفاهيم الأساسية

- with open(name) as f: تفتح ملفًا وتغلقه تلقائيًا.
- CSV (القيم المفصولة بفواصل) هو الصيغة الأكثر شيوعًا للبيانات العامة. وcsv.DictReader تحوّل كل صف إلى قاموس مفاتيحه من سطر العناوين.
- كل ما يُقرأ من ملف هو نص: حوّل باستخدام int() وfloat().
- البيانات الحقيقية فيها تعليقات وفراغات وقيم خاطئة. وtry/except ValueError تتيح لك تخطيها أو تسجيلها.
- يبدأ ملف المتوسطات السنوية في ماونا لوا من NOAA بأسطر تعليق تبدأ بـ #، ثم year,mean,unc.

المعادلات الأساسية

$$\text{rate} = \frac{x_{last} - x_{first}}{t_{last} - t_{first}}$$

```
rate=_last-x_firstt_last-t_first
```

متوسط معدل التغير (المثال 5.1)

$$1 \text{ cfs} = 0.0283168 \text{ m}^3/\text{s}$$

```
1\ cfs=0.0283168\ m^3/s
```

تحويل الوحدات (المثال 5.2)

أمثلة محلولة خطوة بخطوة

مثال 5.1 حلّل الصفوف الخمسة الأخيرة من ملف CSV للمتوسطات السنوية في ماونا لوا من NOAA وأوجد متوسط معدل الارتفاع.

```
import csv, io
text = """year,mean,unc
2021,416.41,0.12
```

```
2022,418.53,0.12
2023,421.08,0.12
2024,424.61,0.12
2025,427.35,0.12
"""
rows = list(csv.DictReader(io.StringIO(text)))
years = [int(r["year"]) for r in rows]
ppm = [float(r["mean"]) for r in rows]
print(years[0], years[-1], ppm[-1])
print(round((ppm[-1] - ppm[0]) / (years[-1] - years[0]), 2), "ppm per year")
```

1. io.StringIO تتيح لـ csv قراءة سلسلة نصية كما لو كانت ملفًا.
2. حوّل الحقول النصية إلى int وfloat.

الناتج

```
2021 2025 427.35
2.73 ppm per year
```

الإجابة: 2.73 ppm سنويًا من 2021 إلى 2025.

مثال 5.2 اكتب ملف CSV صغيرًا لتدفقات USGS، ثم اقرأه وحوّل إلى m^3/s.

```
with open("flows.csv", "w") as f:
    f.write("date,cfs\n2025-06-01,44700\n2025-06-02,40200\n2025-06-03,29100\n")
with open("flows.csv") as f:
    next(f)                      # skip the header
    for line in f:
        date, cfs = line.strip().split(",")
        print(date, round(int(cfs) * 0.0283168, 1), "m3/s")
```

1. next(f) تتخطى سطر العناوين.
2. strip تزيل محرف السطر الجديد؛ وsplit تفصل الحقول.

الناتج

```
2025-06-01 1265.8 m3/s
2025-06-02 1138.3 m3/s
2025-06-03 824.0 m3/s
```

الإجابة: 44700 cfs ≈ 1265.8 m^3/s.

مثال 5.3 اقرأ أسطرًا فيها تعليقات وقيم خاطئة دون أن يتعطل البرنامج.

```
lines = ["# comment", "year,mean", "2024,424.61", "2025,bad", "2026,"]
good = []
for line in lines:
    if line.startswith("#") or line.startswith("year"):
        continue
    year, value = line.split(",")
    try:
        good.append((int(year), float(value)))
    except ValueError:
        print("skipping", repr(line))
print(good)
```

1. continue تتخطى التعليقات وسطر العناوين.
2. float("bad") وfloat("") تطلقان ValueError، فيُلتقط.

الناتج

```
skipping '2025,bad'
skipping '2026,'
[(2024, 424.61)]
```

الإجابة: لا يُحتفظ إلا بالصف الصالح؛ ويُبلَّغ عن الصفوف الخاطئة.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. كيف تقرأ ملف NOAA كاملًا في Colab؟

   استخدم urllib.request.urlopen(url) أو نزّله، ثم تخطَّ الأسطر التي تبدأ بـ #.

2. ماذا تعطي csv.DictReader لكل صف؟

   dict مفاتيحه أسماء الأعمدة، وقيمه سلاسل نصية.

3. لماذا نستخدم with open(...)؟

   يُغلق الملف تلقائيًا، حتى لو حدث خطأ.

مفاهيم خاطئة شائعة

خرافة: الأرقام في CSV أعداد.
في الحقيقة: تُقرأ كسلاسل نصية حتى تُحوَّل.

خرافة: الصفوف الخاطئة تجعل مجموعة البيانات عديمة الفائدة.
في الحقيقة: تخطّها وسجّلها، ثم أبلغ عن عددها.

خرافة: فتح ملف بالوضع "w" يضيف إليه.
في الحقيقة: "w" يكتب فوقه؛ و"a" يضيف إليه.

بطاقات تعليمية

with open()

تفتح ملفًا وتغلقه تلقائيًا.

CSV

قيم مفصولة بفواصل.

csv.DictReader

الصفوف بوصفها قواميس dict.

try / except

التعامل مع الأخطاء.

ValueError

تحويل خاطئ.

الوضع "a" / "w";

الكتابة فوق / الإضافة.

اختبار ذاتي

1. تعطي float("3.5")...

"3.5"

3.5

3

خطأ

3.5. تحوّل النص إلى float.

2. تطلق float("")...

TypeError

ValueError

KeyError

لا شيء

ValueError. السلسلة الفارغة ليست عددًا.

3. وضع الملف "a"...

يكتب فوق المحتوى

يضيف في النهاية

يقرأ

يحذف

يضيف في النهاية. الإضافة.

4. تأتي مفاتيح csv.DictReader من...

أرقام الصفوف

صف العناوين

اسم الملف

عشوائيًا

صف العناوين. السطر الأول.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

تشارك NOAA وNASA وUSGS قياساتها كملفات CSV مجانية. وقدرتك على قراءتها بنفسك تعني أنك تستطيع مقارنة العناوين الإخبارية بالبيانات الأصلية.

على Saving Planets: التفاضل والتكامل 2

المصادر: NOAA GML: Mauna Loa CO_2 annual means (CSV) (https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_annmean_ml.csv) · توثيق بايثون: الوحدة csv (https://docs.python.org/3/library/csv.html)

مصادر تعلّم مجانية

- توثيق بايثون: الوحدة csv (https://docs.python.org/3/library/csv.html): قراءة CSV وكتابتها.
- NOAA GML: Mauna Loa CO_2 annual means (CSV) (https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_annmean_mlo.csv): الملف الحقيقي.
- درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/): القسم 7.2: الملفات.

أنهيت هذه الوحدة

NumPy وMatplotlib

احسب على مصفوفات كاملة دفعة واحدة باستخدام NumPy، وطابق الاتجاهات، وارسم رسومًا واضحة باستخدام Matplotlib.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- أنشئ مصفوفات NumPy واستخدم الحساب المتجه.
- استخدم mean وdiff والأقنعة المنطقية وpolyfit.
- ارسم البيانات بمحاور معنونة وعناوين.
- احفظ الأشكال في ملفات.

المفاهيم الأساسية

- مصفوفة NumPy تحمل أعدادًا من نوع واحد وتجري الحساب عليها كلها دفعة واحدة: cfs * 0.0283168 تحوّل كل قيمة.
- np.diff تعطي الفروق المتتالية؛ والقناع المنطقي مثل cfs > 20000 يختار القيم أو يعدّها.
- np.polyfit(x, y, 1) تطابق خطًا مستقيمًا وتعيد الميل والمقطع.
- pyplot في Matplotlib ترسم الأشكال: fig, ax = plt.subplots(); ax.plot(x, y)؛ عنون المحاور بوحداتها؛ وfig.savefig تحفظ الصورة.
- الرسم الجيد له محاور معنونة بوحداتها وعنوان ومصدر بيانات مذكور.

المعادلات الأساسية

$$y \approx mx + b$$

خط المربعات الصغرى من np.polyfit

$$V = \sum Q_i \, \Delta t$$

```
V= Q_i Δ t
```

الحجم من التدفقات اليومية (المثال 6.2)

أمثلة محلولة خطوة بخطوة

مثال 6.1 استخدم NumPy على المتوسطات السنوية في ماونا لوا 2021–2025: الفروق والمتوسط والاتجاه.

```
import numpy as np
years = np.array([2021, 2022, 2023, 2024, 2025])
ppm = np.array([416.41, 418.53, 421.08, 424.61, 427.35])
print(np.diff(ppm))
```

```
print(round(ppm.mean(), 3), round(np.diff(ppm).mean(), 3))
slope, intercept = np.polyfit(years, ppm, 1)
print(round(slope, 3))
```

1. np.diff تعطي أربع زيادات سنوية.
2. ميل polyfit هو معدل أفضل مطابقة.

الناتج

```
[2.12 2.55 3.53 2.74]
421.596 2.735
2.796
```

الإجابة: الارتفاع بأفضل مطابقة 2.80 ppm ≈ سنويًا خلال 2021–2025.

مثال 6.2 حوّل عشرة أيام من تدفق بوتوماك إلى m³/s، وعُدّ أيام التدفق المرتفع، وأوجد الحجم الكلي.

```
import numpy as np
cfs = np.array([44700, 40200, 29100, 22700, 17300, 14100, 12600, 12200, 14300, 22600])
m3s = cfs * 0.0283168
print(np.round(m3s[:3], 1))
print(int((cfs > 20000).sum()), "days above 20,000 cfs")
print(round(m3s.sum() * 86400 / 1e9, 3), "km3 in 10 days")
```

1. اضرب المصفوفة كلها دفعة واحدة.
2. (cfs > 20000).sum() تعدّ قيم True.
3. مجموع m³/s × 86400 s في اليوم.

الناتج

```
[1265.8 1138.3  824. ]
5 days above 20,000 cfs
0.562 km3 in 10 days
```

الإجابة: خمسة أيام فوق 20000 cfs؛ ومرّ نحو 0.56 km³ من الماء في عشرة أيام.

مثال 6.3 ارسم شذوذات سنوية مختارة من NASA GISTEMP واحفظ الشكل.

```
import matplotlib
matplotlib.use("Agg")
import matplotlib.pyplot as plt
years = [1880, 1951, 1980, 2000, 2016, 2020, 2023, 2024, 2025]
```

```
anom = [-0.17, -0.07, 0.26, 0.39, 1.01, 1.01, 1.17, 1.28, 1.18]
fig, ax = plt.subplots()
ax.plot(years, anom, "o-")
ax.set_xlabel("Year"); ax.set_ylabel("Anomaly (C)")
ax.set_title("NASA GISTEMP v4, selected years")
fig.savefig("gistemp.png")
print(len(ax.lines[0].get_xdata()), "points saved")
```

1. matplotlib.use("Agg") تتيح للكود حفظ الصور دون شاشة (وColab يعرض الرسوم داخل الصفحة على أي حال).
2. عنون المحاور والعنوان؛ واحفظ باستخدام savefig.

الناتج

```
9 points saved
```

الإجابة: رسم خطي محفوظ من 9 نقاط؛ افتح gistemp.png لتراه.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. 2 * np.array([1, 2, 3]) تعطي؟

   array([2, 4, 6]).

2. كيف تختار القيم الأكبر من 1.0 في مصفوفة a؟

   a[a > 1.0].

3. أي استدعاء يضيف عنوانًا للمحور؟

   ax.set_xlabel("Year").

مفاهيم خاطئة شائعة

خرافة: مصفوفات NumPy والقوائم تُجمع بالطريقة نفسها.
في الحقيقة: [3, 4] + [1, 2] يضم القائمتين؛ والمصفوفات تُجمع عنصرًا بعنصر.

خرافة: الخط المطابَق يثبت السببية.
في الحقيقة: إنه يلخص اتجاهًا؛ والأسباب تحتاج إلى فهم فيزيائي.

خرافة: الرسوم دون وحدات مقبولة.
في الحقيقة: اذكر الوحدات دائمًا واستشهد بالمصدر.

بطاقات تعليمية

np.array

مصفوفة عددية سريعة.

متجه

حساب على مصفوفات كاملة.

القناع المنطقي

a[a > x].

np.polyfit

مطابقة متعددة حدود بالمربعات الصغرى.

plt.subplots

تنشئ شكلًا ومحاور.

savefig

تحفظ رسمًا.

اختبار ذاتي

1. np.diff([1, 4, 9]) =

[5 ,3]

[9 ,4 ,1]

[5 ,3 ,0]

14

[3, 5]. الفروق المتتالية.

2. np.polyfit(x, y, 1) تعيد...

المتوسط

الميل والمقطع

رسمًا

القيمة العظمى

الميل والمقطع. مطابقة من الدرجة 1.

3. (a > 0).sum() تعدّ...

القيم الموجبة

كل القيم

الأصفار

لا شيء

القيم الموجبة. True تُحسب 1.

...4. fig.savefig("x.png")

تطبع

تحفظ الشكل

تحذفه

تعرض الكود

تحفظ الشكل. تكتب ملف صورة.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

يُنشر تحليل GISTEMP من NASA مع بياناته ليتمكن أي شخص من إعادة إنتاج سجل درجات الحرارة العالمية. ورسمه بنفسك، مع الوحدات والمصدر، هو الطريقة الأمينة لمشاركة حقائق المناخ.

على Saving Planets: بيانات المناخ على هذا الموقع: CO2M

المصادر: NASA GISS: GISTEMP v4 surface temperature analysis (/https://data.giss.nasa.gov/gistemp) · NumPy: the absolute basics for beginners (https://numpy.org/doc/stable/user/absolute_beginners.html)

مصادر تعلّم مجانية

- NumPy: the absolute basics for beginners (https://numpy.org/doc/stable/user/absolute_beginners.html): المصفوفات للمبتدئين.
- Matplotlib: pyplot tutorial (https://matplotlib.org/stable/tutorials/pyplot.html): أساسيات الرسم.
- NASA GISTEMP v4 global means (CSV) (https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.csv): البيانات الحقيقية.

أنهيت هذه الوحدة

تصحيح الأخطاء والاختبار

اعثر على الأخطاء البرمجية بشكل منهجي، واقرأ تتبعات الأخطاء (tracebacks)، واكتب اختبارات تلتقط الأخطاء قبل أن تصبح مهمة.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- اقرأ traceback وحدد السطر الذي فشل.
- استخدم print والتأكيدات لتضييق نطاق الأخطاء.
- أطلق أخطاء واضحة للمدخلات السيئة.
- اكتب اختبارات وحدات باستخدام assert وpytest.

المفاهيم الأساسية

- اقرأ رسائل الخطأ من الأسفل: السطر الأخير يسمّي نوع الخطأ، والأسطر فوقه تبيّن أين حدث.
- صحّح الأخطاء بأسلوب علمي: كوّن فرضية، واختبرها بمدخل صغير، وغيّر شيئًا واحدًا.
- الحالات الحدية (القوائم الفارغة، والصفر، والأعداد السالبة، والبيانات المفقودة) تسبب معظم الأخطاء.
- assert تنص على ما يجب أن يكون صحيحًا؛ وفشل assert يوقف البرنامج عند الخطأ.
- الاختبارات دوال صغيرة تتحقق من إجابات معروفة؛ وpytest يشغّل تلقائيًا كل دالة اسمها test_....

المعادلات الأساسية

$$|x_{computed} - x_{expected}| < \varepsilon$$

اختبار بهامش تسامح لقيم float

أمثلة محلولة خطوة بخطوة

مثال 7.1 اعثر على الخطأ في دالة متوسط تُستدعى على قائمة فارغة.

```
def mean(xs):
    return sum(xs) / len(xs)

try:
    print(mean([]))
except ZeroDivisionError as e:
    print("bug found:", e)
```

1. القسمة على 0 = len([]) تطلق ZeroDivisionError.
2. التقاطه يُظهر الرسالة.

الناتج

```
bug found: division by zero
```

الإجابة: يجب أن تتعامل الدالة مع المدخل الفارغ.

مثال 7.2 أصلحها بخطأ واضح وأضف اختبارات تأكيد.

```
def mean(xs):
    if not xs:
        raise ValueError("mean() of empty list")
    return sum(xs) / len(xs)

assert mean([1, 2, 3]) == 2
assert abs(mean([0.1, 0.2]) - 0.15) < 1e-12
try:
    mean([])
except ValueError:
    print("all tests passed")
```

1. أطلق ValueError مع رسالة عند المدخل الفارغ.
2. تُقارن نتائج float بهامش تسامح.

الناتج

```
all tests passed
```

الإجابة: تنجح كل الاختبارات.

مثال 7.3 اكتب اختبارًا بأسلوب pytest للدالة f_to_c باستخدام نقاط معروفة.

```
def f_to_c(f):
    return (f - 32) * 5 / 9

def test_f_to_c():
    assert f_to_c(32) == 0
    assert f_to_c(212) == 100
    assert abs(f_to_c(-40) - (-40)) < 1e-12

test_f_to_c()
print("ok")
```

1. أزواج معروفة: 0 °C = 32 °F، 100 °C = 212 °F، −40 = −40.

2. في ملف، سيكتشف pytest الدالة test_f_to_c تلقائيًا؛ وهنا نستدعيها نحن.

الناتج

```
ok
```

الإجابة: تجتاز الدالة الفحوص الثلاثة كلها.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. ماذا تعني عادةً «IndexError: list index out of range»؟

   استخدمت فهرسًا ≥ len(list)، وغالبًا ما يكون خطأ إزاحة بواحد.

2. اكتب assert تتحقق أن x موجب.

   assert x > 0, "x must be positive";.

3. اذكر ثلاث حالات حدية لدالة تحسب متوسط البيانات.

   قائمة فارغة، وقيمة واحدة، وقيم مفقودة أو غير عددية.

مفاهيم خاطئة شائعة

خرافة: إذا اشتغل فهو صحيح.
في الحقيقة: قد يشتغل الكود ويعطي إجابات خاطئة؛ فاختبره مقابل نتائج معروفة.

خرافة: قارن قيم float باستخدام ==.
في الحقيقة: استخدم هامش تسامح.

خرافة: الاختبارات للمشاريع الكبيرة فقط.
في الحقيقة: الاختبارات الصغيرة توفر الوقت من اليوم الأول.

بطاقات تعليمية

Traceback

تقرير الخطأ؛ يُقرأ من الأسفل.

الحالة الحدية

مدخل غير معتاد عند الحدود.

assert

توقّف إذا كان الشرط خاطئًا.

اختبار الوحدة

يتحقق من دالة واحدة مقابل إجابات معروفة.

pytest

مشغّل الاختبارات.

raise

تطلق خطأً عمدًا.

اختبار ذاتي

1. أكثر أسطر traceback فائدة عادةً هو...

الأول

الأخير

الأوسط

لا شيء

الأخير. إنه يسمّي الخطأ.

2. يشغّل pytest الدوال المسماة...

*run_

*test_

*check_

main

*test_. اصطلاح.

3. فشل assert يطلق...

ValueError

AssertionError

لا شيء

TypeError

.AssertionError. AssertionError

4. خطوة أولى جيدة في تصحيح الأخطاء هي...

إعادة كتابة كل شيء

إعادة إنتاج الخطأ بمدخل صغير

إضافة print عشوائية في كل مكان

تجاهله

إعادة إنتاج الخطأ بمدخل صغير. صغّر المشكلة.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

الكود العلمي الذي يوجّه السياسات يجب التحقق منه. وهذا المقرر يطبّق ما يدعو إليه: كل مثال كود في هذه الصفحة يشغّله نص البناء لدينا، ويجب أن يطابق ناتجه تمامًا قبل نشر الصفحة.

على Saving Planets: الدورات

المصادر: توثيق pytest (https://docs.pytest.org/) · درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/)

مصادر تعلّم مجانية

- توثيق pytest (https://docs.pytest.org/): كتابة الاختبارات.
- درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/): القسم 8: الأخطاء والاستثناءات.

شاهد المحاضرة

Lecture 7 testing debugging exceptions and assertions (https://ocw.mit.edu/courses/6-0001-introduction-to-compute-programming-in-python-fall-2016/resources/lecture-7-testing-debugging-exceptions-and-assertions/) (باللغة الإنجليزية)

MIT 6.0001 Introduction to Computer Science and Programming in Python (Fall 2016), MIT OpenCourseWare.

الترخيص: CC BY-NC-SA 4.0 (https://creativecommons.org/licenses/by-nc-sa/4.0/) · شروط الاستخدام (https://ocw.mit.edu/pages/privacy-and-terms-of-use/). الرابط للدراسة المجانية فقط؛ Saving Planets غير تابع لهذه الجامعة ولا يحظى بتأييدها.

أنهيت هذه الوحدة

الخوارزميات والتعقيد

البحث، وإيجاد الجذور، وكيف يزداد زمن التشغيل مع حجم المدخل: التفكير بترميز O الكبير.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- نفّذ البحث الخطي والبحث الثنائي.
- استخدم التنصيف لحل المعادلات.
- قدّر التعقيد: O(1)، O(log n)، O(n)، O(n²).
- اختر هياكل البيانات من أجل السرعة.

المفاهيم الأساسية

- الخوارزمية طريقة دقيقة خطوة بخطوة. وكلفتها هي كيف يزداد عدد الخطوات مع حجم المدخل n.
- البحث الخطي يفحص العناصر واحدًا تلو الآخر: O(n). والبحث الثنائي ينصّف قائمة مرتبة في كل خطوة: O(log n).
- يجد التنصيف جذرًا للدالة f على [a, b] حيث تغيّر f إشارتها، بتنصيف الفترة؛ وبعد n خطوة تصبح الفترة (b − a)/2ⁿ.
- الحلقات المتداخلة على n عنصر هي O(n²): مقبولة لـ 1000 عنصر، وميؤوس منها لمليون.
- البحث في القواميس والمجموعات O(1) في المتوسط، وفي القوائم O(n).

المعادلات الأساسية

$$\text{steps}_{binary} \approx \lceil \log_2 n \rceil$$

خطوات البحث الثنائي

$$n = \left\lceil \log_2 \frac{b-a}{\text{tol}} \right\rceil$$

`n=_2-atol`

تكرارات التنصيف

أمثلة محلولة خطوة بخطوة

مثال 8.1 ابحث عن العنصر الأخير في قائمة مرتبة من مليون عدد زوجي، بالطريقتين.

```
def linear_search(xs, target):
    for i, x in enumerate(xs):
        if x == target:
            return i
    return -1
```

```
def binary_search(xs, target):
    lo, hi, steps = 0, len(xs) - 1, 0
    while lo <= hi:
        steps += 1
        mid = (lo + hi) // 2
        if xs[mid] == target:
            return mid, steps
        if xs[mid] < target:
            lo = mid + 1
        else:
            hi = mid - 1
    return -1, steps

data = list(range(0, 2_000_000, 2))
print(linear_search(data, 1_999_998))
print(binary_search(data, 1_999_998))
```

1. يفحص البحث الخطي 1000000 عنصر ليصل إلى الفهرس 999999.
2. يحتاج البحث الثنائي إلى 20 خطوة فقط.

الناتج

```
999999
(999999, 20)
```

الإجابة: 20 خطوة مقابل مليون: $\log_2(10^6) \approx 20$.

مثال 8.2 أوجد $\sqrt{2}$ بالتنصيف على $x^2 - 2$ بدءًا من $[0, 2]$ حتى 10^{-6}.

```
def bisect_root(f, a, b, tol=1e-6):
    n = 0
    while b - a > tol:
        m = (a + b) / 2
        if f(a) * f(m) <= 0:
            b = m
        else:
            a = m
        n += 1
    return (a + b) / 2, n

root, n = bisect_root(lambda x: x**2 - 2, 0, 2)
print(round(root, 6), n)
```

1. كل خطوة تنصّف الفترة.
2. $2/2^{21} < 10^{-6}$ بعد 21 خطوة.

الناتج

```
1.414214 21
```

الإجابة: 1.414214 بعد 21 تكرارًا.

مثال 8.3 قارن نمو $\log_2 n$ وn^2.

```
import math
for n in (10, 1_000, 1_000_000):
    print(n, math.ceil(math.log2(n)), n * n)
```

1. math.ceil(math.log2(n)) هو عدد خطوات البحث الثنائي.

الناتج

```
10 4 100
1000 10 1000000
1000000 20 1000000000000
```

الإجابة: عندما $n = 10^6$، يكون $\log_2 n$ مساويًا 20 بينما n^2 يساوي 10^{12}.

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. كم خطوة بحث ثنائي لـ 1 مليار عنصر مرتب؟

   $\lceil \log_2 10^9 \rceil = 30$.

2. ما تعقيد جمع قائمة؟

   $O(n)$.

3. التنصيف على $[0, 1]$ حتى 10^{-3}: كم خطوة؟

   $\lceil \log_2 1000 \rceil = 10$.

مفاهيم خاطئة شائعة

خرافة: الحواسيب الأسرع تجعل التعقيد غير مهم.
في الحقيقة: طريقة $O(n^2)$ على مليون عنصر تعني 10^{12} خطوة؛ ولا حاسوب يصلح ذلك.

خرافة: البحث الثنائي يعمل على أي قائمة.

في الحقيقة: على البيانات المرتبة فقط.

خرافة: ترميز O الكبير يعطي الزمن الدقيق.
في الحقيقة: إنه يصف النمو متجاهلًا الثوابت.

بطاقات تعليمية

الخوارزمية

طريقة دقيقة خطوة بخطوة.

Big-O

معدل نمو الكلفة.

البحث الخطي

O(n).

البحث الثنائي

O(log n)، بيانات مرتبة.

التنصيف

إيجاد الجذور بالتنصيف.

O(n²)

حلقات متداخلة.

اختبار ذاتي

1. البحث الثنائي...

O(n)

O(log n)

O(n²)

O(1)

O(log n). ينصّف في كل خطوة.

2. يتطلب التنصيف...

مشتقة

تغيّر الإشارة على [a, b]

بيانات مرتبة

أعدادًا صحيحة

تغيّر الإشارة على [a, b]. مبرهنة القيمة الوسطى.

3. حلقة داخل حلقة على n عنصر هي…

O(n)

O(n²)

O(log n)

O(1)

O(n²). n × n.

4. متوسط البحث في dict هو…

O(n)

O(1)

O(n²)

O(log n)

O(1). التجزئة (Hashing).

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

الخوارزميات الفعالة تتيح معالجة سيل بيانات رصد الأرض، من الأقمار الصناعية إلى آلاف مقاييس الأنهار. واختيار O(n log n) بدل O(n²) قد يحوّل أسبوعًا من الحوسبة إلى ثوانٍ، والحوسبة الأقل تعني أيضًا طاقة أقل.

على Saving Planets: التفاضل والتكامل 1

المصادر: USGS Water Services (واجهة برمجة البيانات) (https://waterservices.usgs.gov/) · MIT OpenCourseWare 6.100L: Introduction to CS and Programming Using Python (https://ocw.mit.edu/courses/6-100l-introduction-to-cs-and-programming-using-python-fall-2022/)

مصادر تعلّم مجانية

- MIT OpenCourseWare 6.100L: Introduction to CS and Programming Using Python (https://ocw.mit.edu/courses/6-100l-introduction-to-cs-and-programming-using-python-fall-2022/): محاضرات عن البحث والتنصيف والتعقيد.
- Harvard CS50's Introduction to Programming with Python (https://cs50.harvard.edu/python/): تدريب على حل المسائل.

- [ ] **أنهيت هذه الوحدة**

المحاكاة العددية

قدّم النماذج الفيزيائية في الزمن خطوة بخطوة: مقذوف، ومدار، وتجمع سكاني، وهي جوهر كل نموذج مناخي وهندسي.

أهداف التعلّم

في نهاية هذه الوحدة ستستطيع أن:

- استخدم الخطوات الزمنية (أويلر وأويلر-كرومر) لمحاكاة الحركة.
- تحقق من المحاكاة مقابل الإجابات الدقيقة.
- حاكِ مدارًا دائريًا وتجمعًا سكانيًا لوجستيًا.
- استكشف كيف يؤثر حجم الخطوة في الدقة.

المفاهيم الأساسية

- تحدّث المحاكاة الحالة (الموضع، والسرعة، والتجمع السكاني) بخطوات زمنية صغيرة dt باستخدام معدلات التغير: x_new = x + v·dt.
- أويلر-كرومر تحدّث السرعة أولًا، ثم الموضع بالسرعة الجديدة؛ وتحافظ على استقرار المدارات أفضل بكثير من أويلر البسيطة.
- تحقق دائمًا: قارن بحل دقيق في حالة يوجد فيها (دون سحب هوائي، أو مدار دائري، أو الصيغة اللوجستية).
- dt الأصغر أدق لكنه أبطأ؛ نصّف dt وانظر هل تتغير الإجابة.
- نماذج الطقس والمناخ نسخ ضخمة من هذه الحلقات، تدفع معادلات الفيزياء قدمًا عبر شبكة عالمية.

المعادلات الأساسية

$$v_{k+1} = v_k + a_k\,\Delta t, \quad x_{k+1} = x_k + v_{k+1}\,\Delta t$$

```
v_k+1=v_k+a_k Δ t, x_k+1=x_k+v_k+1 Δ t
```

خطوة أويلر-كرومر

$$R = \frac{v_0^2 \sin 2\theta}{g}$$

```
R=(v_0^22θ)/(g)
```

مدى المقذوف دون سحب

$$\vec{a} = -\frac{GM}{r^3}\vec{r}$$

```
a=-(GM)/(r^3) r
```

الجاذبية لمدار

$$P_{k+1} = P_k + h\,r P_k\left(1 - \frac{P_k}{K}\right)$$

أمثلة محلولة خطوة بخطوة

مثال 9.1 حاكِ مقذوفًا بسرعة 20 m/s بزاوية 45° وقارن بالمدى الدقيق.

```
import math
v0, angle, g, dt = 20.0, math.radians(45), 9.81, 0.001
x, y = 0.0, 0.0
vx, vy = v0 * math.cos(angle), v0 * math.sin(angle)
while y >= 0:
    vy -= g * dt
    x += vx * dt
    y += vy * dt
print(round(x, 2), "m simulated;", round(v0**2 / g, 2), "m exact")
```

1. قدّم السرعة ثم الموضع حتى يهبط.
2. المدى الدقيق $v_0^2 \sin 90°/g$ =.

الناتج

```
40.77 m simulated; 40.77 m exact
```

الإجابة: كلاهما يعطي 40.77 m لمنزلتين عشريتين مع dt = 0.001 s.

مثال 9.2 حاكِ مدارًا على ارتفاع 400 km فوق الأرض بخطوات مدة كل منها 1 ثانية.

```
import math
GM = 3.986e14
r0 = 6.771e6
x, y = r0, 0.0
vx, vy = 0.0, math.sqrt(GM / r0)
dt, t = 1.0, 0
period = 2 * math.pi * math.sqrt(r0**3 / GM)
while t < period:
    r3 = (x * x + y * y) ** 1.5
    vx -= GM * x / r3 * dt          # Euler-Cromer: update velocity first
    vy -= GM * y / r3 * dt
    x += vx * dt
    y += vy * dt
```

```
    t += dt
print(round(period / 60, 1), "min;", round(math.hypot(x, y) / 1000), "km from Earth's center
```

1. ابدأ عند r_0 بالسرعة الدائرية $\sqrt{GM/r_0}$.
2. حدّث السرعة من الجاذبية، ثم الموضع.

الناتج

```
92.4 min; 6771 km from Earth's center
```

الإجابة: بعد دورة واحدة (92.4 دقيقة) تبقى المركبة على بعد 6771 km من مركز الأرض: المدار يبقى دائريًا.

مثال 9.3 قدّم تجمعًا لوجستيًا من الأسماك ($r = 0.5$، $K = 1000$، $P_0 = 100$) لأربع سنوات.

```
r, K, P, h = 0.5, 1000, 100.0, 1.0
for year in range(1, 5):
    P += h * r * P * (1 - P / K)
    print(year, round(P, 1))
```

1. كل سنة تضيف $rP(1 - P/K)$.

الناتج

```
1 145.0
2 207.0
3 289.1
4 391.8
```

الإجابة: 145.0، 207.0، 289.1، 391.8، مطابقًا للمختبر A في التفاضل والتكامل II (هناك 391.9، فرق تقريب).

تمارين

حلّ كلًّا منها على الورق أولًا، ثم تحقق.

1. ماذا يحدث لخطأ المثال 9.1 إذا كان dt يساوي 0.1 s؟

   يُكتشف الهبوط متأخرًا، فيزداد خطأ المدى؛ جرّب وقارن.

2. لماذا تجعل طريقة أويلر البسيطة المدار يلتف حلزونيًا إلى الخارج؟

   تضيف قليلًا من الطاقة في كل خطوة؛ أما أويلر-كرومر فتحفظها تقريبًا.

3. أضف السحب $-k\,v\,v$ إلى المثال 9.1. ماذا يحدث لأفضل زاوية؟

تنخفض دون 45° (المختبر D).

مفاهيم خاطئة شائعة

خرافة: المحاكاة التي تعمل صحيحة.
في الحقيقة: تحقق مقابل حالات دقيقة وتأكد أن تنصيف dt بالكاد يغيّر الإجابة.

خرافة: dt الأصغر أفضل دائمًا.
في الحقيقة: إنه يكلف وقتًا وقد يضيف خطأ تقريب؛ اختر dt باختبارات التقارب.

خرافة: النماذج تتنبأ بالمستقبل بدقة.
في الحقيقة: إنها تقريبات يجب ذكر افتراضاتها.

بطاقات تعليمية

الخطوة الزمنية dt

حجم كل تحديث.

طريقة أويلر

x += v·dt.

أويلر-كرومر

السرعة أولًا، ثم الموضع.

التحقق

مقارنة بإجابات معروفة.

التقارب

تتوقف الإجابة عن التغير مع تصغير dt.

الحالة

المتغيرات التي تصف النظام الآن.

اختبار ذاتي

1. أويلر-كرومر تحدّث...

الموضع أولًا

السرعة أولًا، ثم الموضع بالسرعة الجديدة

السرعة فقط

لا شيء

السرعة أولًا، ثم الموضع بالسرعة الجديدة. مستقرة للمدارات.

2. للتحقق من محاكاة...

ثق بها

قارنها بحل دقيق

اجعل dt ضخمًا

أزل الفيزياء

قارنها بحل دقيق. حالات معروفة.

3. المدى الدقيق عند 20 m/s و45° و9.81 = g:

20.4 m

40.8 m

81.5 m

9.8 m

.40.8 m. 400/9.81

4. تنصيف dt دون رؤية تغير يعني...

خطأً برمجيًا

أن النتيجة تقاربت

أن النموذج خاطئ

لا شيء

أن النتيجة تقاربت. اختبار التقارب.

أعد هذا الاختبار

كيف يساعد هذا في إنقاذ الكوكب

نماذج التجمعات السكانية مثل المثال 9.3 جزء من طريقة تقييم علماء مصايد الأسماك للمخزونات ووضع حدود صيد مستدامة.

فكرة الخطوات الزمنية نفسها، بعد تكبيرها هائلًا، هي طريقة بناء نماذج الطقس والمناخ.

على Saving Planets: التفاضل والتكامل II، المختبر A · دليل المدارات

المصادر: NOAA Fisheries: تقييمات المجموعات السمكية (https://www.fisheries.noaa.gov/topic/population-assessments) · MIT OpenCourseWare 6.100L: Introduction to CS and Programming Using Python (https://ocw.mit.edu/courses/6-100l-introduction-to-cs-and-programming-using-python-fall-2022/)

مصادر تعلّم مجانية

- MIT OpenCourseWare 6.100L: Introduction to CS and Programming Using Python (https://ocw.mit.edu/courses/6-100l-introduction-to-cs-and-programming-using-python-fall-2022/): محاضرات المحاكاة والنمذجة.
- NumPy: the absolute basics for beginners (https://numpy.org/doc/stable/user/absolute_beginners.html): تحويل المحاكاة إلى عمليات متجهية.

أنهيت هذه الوحدة

مجموعات المسائل

هذه التمارين داخل كل وحدة. حلّها على الورق أولًا، ثم افتح الإجابة.

1. البداية: المتغيرات والأنواع والتعابير
   4 مسائل
2. التحكم في التدفق: الشروط والحلقات
   3 مسائل
3. الدوال
   3 مسائل
4. القوائم والقواميس والسلاسل النصية
   3 مسائل
5. الملفات وبيانات CSV
   3 مسائل
6. NumPy وMatplotlib
   3 مسائل
7. تصحيح الأخطاء والاختبار
   3 مسائل
8. الخوارزميات والتعقيد
   3 مسائل
9. المحاكاة العددية
   3 مسائل

مختبرات البرمجة

تستخدم أربعة مختبرات برمجة بيانات عامة حقيقية ومحاكاة: سجل CO_2 في ماونا لوا، وسجل درجة الحرارة العالمية من ناسا، ومحطة قياس نهرية تابعة لهيئة USGS، ومقذوفًا مع سحب هوائي.

تعمل الشيفرة الأولية لكل مختبر كما هي معروضة؛ وقد نتجت النتائج المطبوعة عن تشغيلها. ثم نزّل مجموعة البيانات كاملة ووسّع التحليل.

مرجع السلامة: درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/)

المختبر A: سجل CO_2 في ماونا لوا (NOAA)

الأسبوع 6

الهدف: حمّل ملف CSV للمتوسطات السنوية لـ CO_2 من NOAA، واحسب الارتفاع الكلي ومعدل النمو الأخير، وارسم المنحنى.

$$\text{rate} = \text{slope of } \text{np.polyfit}(\text{year, ppm, 1})$$

```
rate=slope of np.polyfit(year, ppm, 1)
```

معدل النمو من ملاءمة خطية

الإجراء

1. شغّل الشيفرة الأولية (بضعة صفوف حقيقية من ملف NOAA).
2. نزّل ملف CSV كاملًا من NOAA (الرابط أدناه)، وتخطَّ الأسطر التي تبدأ بـ #، وحمّل كل السنوات.
3. ارسم ppm مقابل السنة مع محاور معنونة والمصدر.
4. لائم خطوطًا للفترتين 1960–1969 و2016–2025. كيف تغيّر المعدل؟

```
import csv, io
import numpy as np
text = """year,mean,unc
1959,315.98,0.12
1960,316.91,0.12
2020,414.21,0.12
2021,416.41,0.12
2022,418.53,0.12
2023,421.08,0.12
2024,424.61,0.12
2025,427.35,0.12
"""
rows = list(csv.DictReader(io.StringIO(text)))
yr = np.array([int(r["year"]) for r in rows])
ppm = np.array([float(r["mean"]) for r in rows])
print(round(ppm[-1] - ppm[0], 2), "ppm since 1959")
recent = yr >= 2020
slope = np.polyfit(yr[recent], ppm[recent], 1)[0]
print(round(slope, 2), "ppm per year, 2020-2025")
```

الناتج

```
111.37 ppm since 1959
2.65 ppm per year, 2020-2025
```

جدول البيانات

السنة	متوسط CO_2 السنوي (ppm)
1959	315.98
1960	316.91
2020	414.21
2023	421.08
2024	424.61
2025	427.35

قيم حقيقية من ملف المتوسطات السنوية لماونا لوا لدى NOAA GML، استُرجعت في 24 أكتوبر 2026.

مثال حسابي

1. 427.35 − 315.98 = 111.37 ppm منذ 1959.
2. معدل أفضل ملاءمة 2.65 ppm ≈ 2020–2025 في السنة.

السلامة

- لا شيء يتجاوز الاستخدام العادي للحاسوب.

بديل منزلي أو افتراضي

- Colab أو JupyterLite في متصفح.

NOAA GML: Mauna Loa CO_2 annual means (CSV) (https://gml.noaa.gov/webdata/ccgg/trends/co2/co2_annmean_mlo.csv) · Google Colab (دفاتر مجانية) (https://colab.research.google.com/)

الصلة بالكوكب: هذا هو السجل وراء منحنى كيلينغ، أطول قياس متواصل لارتفاع CO_2. NOAA GML: Trends in atmospheric CO_2, data (https://gml.noaa.gov/ccgg/trends/data.html)

☐ **أكملت هذا المختبر (عمليًا أو افتراضيًا)**

المختبر B: درجة الحرارة العالمية من NASA GISTEMP

الأسبوع 8

الهدف: حلّل شذوذات درجة الحرارة السنوية العالمية من ناسا: المتوسطات، وأدفأ سنة، والاتجاه.

$$\Delta T \text{ per decade} = 10 \times \text{slope}$$

```
Δ T\ per decade=10×slope
```

الاتجاه لكل عقد

الإجراء

1. شغّل الشيفرة الأولية مع الشذوذات السنوية (J–D) للفترة 2016–2025.
2. نزّل ملف CSV العالمي الكامل من GISTEMP؛ وتخطَّ سطر العنوان الأول؛ وتعامل مع القيم المفقودة ***.
3. ارسم 1880–2025 واحسب الاتجاه منذ 1975.
4. قارن ملاءمة على 10 سنوات بملاءمة على 50 سنة، واشرح لماذا تكون الاتجاهات القصيرة أكثر ضوضاء.

```
import numpy as np
years = np.arange(2016, 2026)
anom = np.array([1.01, 0.91, 0.85, 0.98, 1.01, 0.85, 0.89, 1.17, 1.28, 1.18])
print(round(anom.mean(), 3))
print(int(years[anom.argmax()]), anom.max())
print(round(np.polyfit(years, anom, 1)[0] * 10, 2), "C per decade (10-year fit)")
```

الناتج

```
1.013
2024 1.28
0.32 C per decade (10-year fit)
```

جدول البيانات

السنة	الشذوذ (°C مقارنة بـ 1951–1980)
2016	1.01
2020	1.01
2023	1.17
2024	1.28
2025	1.18

قيم حقيقية من NASA GISTEMP v4 (GLB.Ts+dSST، العمود J–D)، استُرجعت في 24 أكتوبر 2026. تحدّث ناسا التحليل، لذا قد تختلف التنزيلات اللاحقة قليلًا.

مثال حسابي

1. المتوسط 2016–2025 = 1.013 °C.
2. ملاءمة عشر سنوات تعطي نحو 0.32 °C لكل عقد، لكن عشر سنوات فترة قصيرة وكثيرة الضوضاء؛ استخدم فترات أطول للاتجاه طويل الأمد.

السلامة

- لا شيء يتجاوز الاستخدام العادي للحاسوب.

بديل منزلي أو افتراضي

- Colab أو JupyterLite.

NASA · NASA GISTEMP v4 global means (CSV) (https://data.giss.nasa.gov/gistemp/tabledata_v4/GLB.Ts+dSST.csv) GISS: GISTEMP v4 surface temperature analysis (/https://data.giss.nasa.gov/gistemp)

الصلة بالكوكب: إعادة إنتاج سجل ناسا بنفسك هي أفضل إجابة عن سؤال «كيف نعرف؟» NASA GISS: GISTEMP v4 surface temperature analysis (/https://data.giss.nasa.gov/gistemp)

أكملت هذا المختبر (عمليًا أو افتراضيًا)

المختبر C: نهر عبر محطة قياس USGS

الأسبوع 9

الهدف: حلّل التدفق اليومي من محطة USGS على نهر بوتوماك عند ليتل فولز: الوحدات، والإحصاءات، والانحسار، والحجم.

$$V = \sum_{i} Q_i \times 86{,}400 \text{ s}$$

V=_iQ_i×86,400\ s

الحجم من متوسطات التدفق اليومية

الإجراء

1. شغّل الشيفرة الأولية مع عشرة أيام حقيقية (1–10 يونيو 2025).
2. اطلب سنة كاملة من USGS Water Services (format=rdb) للموقع 01646500 والمعامل 00060، وحلّل الملف المفصول بعلامات الجدولة.
3. ارسم التدفق اليومي، وجد الذروة، واحسب الحجم السنوي بوحدة km³.
4. اكتب اختبارات لدالة تحويل الوحدات الخاصة بك.

```
import numpy as np
cfs = np.array([44700, 40200, 29100, 22700, 17300, 14100, 12600, 12200, 14300, 22600])
m3s = cfs * 0.0283168
print(round(m3s.mean(), 1), round(np.median(m3s), 1))
drops = np.diff(cfs)
print(int(drops.min()), "cfs biggest one-day drop")
print(round(m3s.sum() * 86400 / 1e6, 1), "million m3 in 10 days")
```

الناتج

```
650.7 564.9
-11100 cfs biggest one-day drop
562.2 million m3 in 10 days
```

جدول البيانات

التاريخ (2025)	متوسط التدفق اليومي (cfs)
1 يونيو	44,700
3 يونيو	29,100
5 يونيو	17,300
8 يونيو	12,200
10 يونيو	22,600

قيم يومية حقيقية معتمدة من موقع USGS رقم 01646500، استُرجعت في 24 أكتوبر 2026.

مثال حسابي

1. المتوسط 650.7 m³/s، والوسيط 564.9 m³/s (ذروة 1 يونيو ترفع المتوسط).
2. أكبر انخفاض في يوم واحد كان 11100 cfs؛ ومرّ نحو 562 مليون m³ في عشرة أيام.

السلامة

- لا شيء يتجاوز الاستخدام العادي للحاسوب.

بديل منزلي أو افتراضي

- Colab أو JupyterLite.

USGS: Potomac River near Washington, DC (Little Falls), site 01646500 (https://waterdata.usgs.gov/monitoring-location/USGS-01646500/) · USGS Water Services (واجهة برمجة البيانات) (https://waterservices.usgs.gov/)

الصلة بالكوكب: تدعم محطات القياس النهرية مثل هذه إمدادات مياه الشرب، والإنذار بالفيضانات، والتدفقات البيئية. USGS: Potomac River near Washington, DC (Little Falls), site 01646500 (https://waterdata.usgs.gov/monitoring-location/USGS-01646500/)

☐ **أكملت هذا المختبر (عمليًا أو افتراضيًا)**

المختبر D: مقذوف مع سحب هوائي

الأسبوع 14

الهدف: وسّع محاكاة المقذوف بسحب تربيعي، وجد أفضل زاوية إطلاق مع ازدياد السحب.

$$\vec{a} = -g\hat{y} - k\,|\vec{v}|\,\vec{v}$$

a=-g y-k | v| v

السحب التربيعي (k لكل متر)

الإجراء

1. شغّل الشيفرة الأولية، التي تبحث في الزوايا الصحيحة من 20° إلى 60°.
2. تحقق من أن k = 0 يعطي 45° والمدى الدقيق.
3. نصّف dt وتأكد أن النتائج بالكاد تتغير.
4. ارسم المدى مقابل الزاوية لكل قيمة k.

```
import math
def fly(v0, deg, k=0.0, dt=0.001, g=9.81):
    vx, vy = v0 * math.cos(math.radians(deg)), v0 * math.sin(math.radians(deg))
    x = y = 0.0
    while y >= 0:
        v = math.hypot(vx, vy)
        vx -= k * v * vx * dt
        vy -= (g + k * v * vy) * dt
        x += vx * dt
        y += vy * dt
    return x
for k in (0.0, 0.01, 0.05):
    best = max(range(20, 61), key=lambda a: fly(20, a, k))
    print(k, best, round(fly(20, best, k), 1))
```

الناتج

```
0.0 45 40.8
0.01 43 31.4
0.05 39 17.8
```

جدول البيانات

السحب k (1/m)	أفضل زاوية	المدى (m)
0	45°	40.8
0.01	43°	31.4

السحب k (1/m)	أفضل زاوية	المدى (m)
0.05	39°	17.8

محسوب بالشيفرة الأولية (v_0 = 20 m/s، dt = 0.001 s).

مثال حسابي

1. دون سحب تكون أفضل زاوية 45° والمدى 40.8 m (القيمة الدقيقة 40.77 m).
2. يقصّر السحب المدى ويخفض أفضل زاوية.

السلامة

- لا شيء يتجاوز الاستخدام العادي للحاسوب.

بديل منزلي أو افتراضي

- Colab أو JupyterLite.

Google Colab (دفاتر مجانية) (https://colab.research.google.com/) · JupyterLite: بايثون في متصفحك (Pyodide) (https://jupyter.org/try-jupyter/lab/)

الصلة بالكوكب: السحب هو الفيزياء نفسها التي تحدد استهلاك الطائرة للوقود (مقدمة في هندسة الطيران والفضاء).

☐ **أكملت هذا المختبر (عمليًا أو افتراضيًا)**

مجموعات تدريب للامتحان

وقّت نفسك: نحو 10 دقائق لكل مسألة. ثم قارن بالحلول الكاملة.

مجموعة تمارين منتصف الفصل (الوحدات 1–5)

1. ماذا يطبع print(7 // 2, 7 % 2)؟

   1. القسمة الصحيحة والباقي.

   الإجابة: 3 1

2. اكتب حلقة تجمع الأعداد من 1 إلى 100.

   1. total = 0; for n in range(1, 101): total += n.

   الإجابة: 5050

3. اكتب stall_speed(W, rho, S, cl_max=1.6) وقيّمها عند 10791 N و1.227 kg/m³ و16.2 m².

1. (2W/(ρSC))**0.5 .1.

الإجابة: 26.0 m/s

4. إذا كان co2 = [418.53, 421.08, 424.61, 427.35]، فاحسب التغيرات من سنة إلى أخرى.

1. [b - a for a, b in zip(co2, co2[1:])].

الإجابة: [2.55, 3.53, 2.74]

5. لماذا يحتاج float("") إلى try/except عند قراءة البيانات؟

1. إنه يطلق ValueError.

الإجابة: لتخطي الصفوف التالفة أو تسجيلها بدل توقف البرنامج.

مجموعة التدريب النهائية (كل الوحدات)

1. حوّل 44700 cfs إلى m³/s.

1. 0.0283168 ×.

الإجابة: ≈ 1,265.8 m³/s

2. كم خطوة بحث ثنائي تلزم لمليون عنصر مرتب؟

1. ⌈log₂ 10⁶⌉.

الإجابة: 20

3. كم خطوة تنصيف تجعل [0, 2] أصغر من 10⁻⁶؟

1. 2/2ⁿ < 10⁻⁶.

الإجابة: 21

4. ما المدى الدقيق لمقذوف بسرعة 20 m/s بزاوية 45° (g = 9.81)؟

1. v₀²/g.

الإجابة: 40.77 m

5. أربع خطوات أويلر لـ P += 0.5P(1 − P/1000) بدءًا من 100؟

1. 145.0, 207.0, 289.1, ...

الإجابة: 391.8

6. اكتب اختبارًا لـ f_to_c.

1. assert f_to_c(212) == 100، وغيرها.

الإجابة: أزواج معروفة: 0 → 32، 100 → 212، 40− → 40−.

ورقة الصيغ

$$7//2 = 3,\ 7\,\%\,2 = 1,\ 2^{10} = 1024$$

$$\bar{x} = \tfrac{1}{n}\sum x_i$$

$$\text{rate} = \Delta y / \Delta x$$

$$y \approx mx + b\ \text{(np.polyfit)}$$

$$1\ \text{cfs} = 0.0283168\ \text{m}^3/\text{s}$$

$$\text{binary search} \approx \lceil \log_2 n \rceil$$

$$n_{bisect} = \lceil \log_2((b-a)/\text{tol}) \rceil$$

$$v \mathrel{+}= a\,\Delta t,\ x \mathrel{+}= v\,\Delta t$$

$$R = v_0^2 \sin 2\theta / g$$

$$P \mathrel{+}= hrP(1 - P/K)$$

الثوابت

الثابت	القيمة
1 قدم مكعبة في الثانية (cfs)	0.0283168 m³/s
عدد الثواني في اليوم	86,400
CO_2 في ماونا لوا، المتوسط السنوي لعام 1959	315.98 ppm
CO_2 في ماونا لوا، المتوسط السنوي لعام 2025	427.35 ppm
شذوذ GISTEMP لعام 2024 (مقارنة بـ 1951–1980)	1.28 °C
GM للأرض	3.986 × 10¹⁴ m³/s²
g	9.81 m/s²

المصدر: NOAA GML: Trends in atmospheric CO_2, data (https://gml.noaa.gov/ccgg/trends/data.html)

أدوات ومراجع مجانية

شغّل بايثون مجانًا في متصفحك وابحث عن المعلومات هنا.

- Google Colab (دفاتر مجانية) (https://colab.research.google.com/): دفاتر مجانية مثبت فيها NumPy وMatplotlib مسبقًا.
- JupyterLite: بايثون في متصفحك (Pyodide) (https://jupyter.org/try-jupyter/lab/): بايثون في المتصفح، دون حساب، مدعوم بـ Pyodide.
- درس بايثون التعليمي الرسمي (https://docs.python.org/3/tutorial/): الدليل التعليمي المرجعي.

- MIT OpenCourseWare 6.100L: Introduction to CS and Programming Using Python (https://ocw.mit.edu/courses/6-100l-introduction-to-cs-and-programming-using-python-fall-2022/): مقرر MIT التمهيدي الكامل.
- Harvard CS50's Introduction to Programming with Python (https://cs50.harvard.edu/python/): مقرر بايثون المجاني من هارفارد.

مسرد المصطلحات

الخوارزمية
طريقة دقيقة خطوة بخطوة.

المصفوفة
حاوية NumPy للحسابات العددية السريعة.

التأكيد
فحص يوقف البرنامج إذا كان خاطئًا.

Big-O
كيف تنمو الكلفة مع حجم المدخلات.

التنصيف
إيجاد الجذور بتنصيف فترة.

CSV
ملف قيم مفصولة بفواصل.

القاموس
ربط مفتاح → قيمة.

الحالة الحدية
مدخل غير معتاد عند حدّ.

أويلر-كرومر
خطوات زمنية مستقرة: السرعة أولًا.

الاستثناء
خطأ يُطلق أثناء التنفيذ.

الدالة
كتلة شيفرة مسماة قابلة لإعادة الاستخدام.

القائمة
تسلسل مرتب قابل للتغيير.

الحلقة
تنفيذ متكرر.

الدفتر
مستند من خلايا شيفرة قابلة للتشغيل.

Pyodide
بايثون مُصرَّفة لتعمل في متصفح الويب.

المحاكاة
دفع نموذج إلى الأمام في الزمن.

Traceback
تقرير بمكان حدوث الخطأ.

اختبار الوحدة
فحص صغير لدالة واحدة.

المتغير
اسم مرتبط بقيمة.

f-string
سلسلة نصية منسقة حرفية.

معايير التقييم الذاتي

لا تُمنح درجة ولا ساعات معتمدة. استخدم هذا لتقيّم بصدق أين أنت في كل وحدة.

المستوى	كيف يبدو
4 · إتقان	يكتب دوال واضحة ومختبرة؛ ويتحقق من النتائج مقابل إجابات معروفة؛ ويشرح اختياراته.
3 · متمكن	البرامج تعمل مع أخطاء عرضية تُكتشف بالاختبار.
2 · قيد التطور	يستطيع تعديل الأمثلة لكنه يجد صعوبة في الكتابة من الصفر.
1 · مبتدئ	ينسخ الشيفرة دون فهم؛ راجع الوحدة.

الأوزان المقترحة إذا أردت درجة

الجزء	الوزن
تمارين	20%
مختبرات البرمجة	35%
مجموعة تدريب منتصف الفصل	20%
مجموعة التدريب النهائية	25%

استهدف المستوى 3 أو أعلى في كل وحدة قبل الانتقال إلى الدورة التالية.

دورات ذات صلة

- الرسم الهندسي / التصميم بالحاسوب ENGR 120
- التفاضل والتكامل 3 MATH 210
- الكيمياء العامة + 1 المختبر CHEM 110 + CHEM 110L
- كل الدورات