$ python3 zero_to_builder.py

النسخة العربية

PYTHON

ZERO TO BUILDER

كتاب بايثون الذي يعلّمك أن **تبني**، لا أن تحفظ.

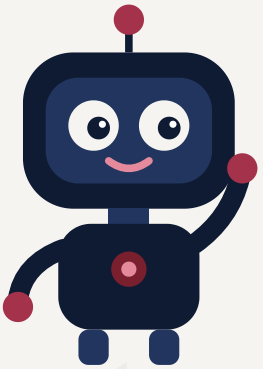
ماذا يوجد بالداخل؟

- شرح خطوة بخطوة
- 8 مشاريع حقيقية
- كويز في كل فصل
- تمارين بـ 3 مستويات
- امتحان نهائي
- مفتاح الإجابات + أكواد الحلول
- متحف الأخطاء
- نصائح للتعلم بالذكاء الاصطناعي
- الأكواد كاملة

مناسب للمبتدئين · بدون أي خبرة سابقة

Elmasryx

أهلاً! هيا نحوّلك إلى صانع حقيقي.



بِت · مرشدك

مرحباً، أنا **بِت**. سأكون معك في كل صفحة. وعندما يبدو شيء صعباً، سأقفز وأجعله بسيطاً. اتفقنا؟

بايثون هي اللغة التي تعمل خلف أدوات الذكاء الاصطناعي، وخلف البيانات التي تعتمد عليها الشركات، وخلف البرامج الصغيرة التي تنجز في **3 ثوانٍ** عملاً مملاً يأخذ **3 ساعات**. أتقنها، وستردّ لك الجميل:

وظيفة
وظائف الذكاء الاصطناعي والبيانات والأتمتة تستخدمها يومياً.

عمل حر
العملاء يدفعون لبرامج توفّر لهم ساعات.

ترقية في عملك
أتمت الأجزاء المملة، ولاحظ الفرق.

اصنع أدواتك
حوّل برنامجاً صغيراً إلى أداة تبيعها.

#1

بايثون هي اللغة الأكثر استخداماً بين من يتعلمون البرمجة: 71.8% منهم يستخدمونها.

المصدر: Stack Overflow Developer Survey 2025

Elmasryx

جاهز خلال 10 دقائق.

تحتاج شيئين مجانيين فقط: **بايثون** وهي اللغة، و**VS Code** وهو الدفتر الذي تكتب فيه. هذه كل القائمة.

1 نزّل بايثون

ادخل على **python.org/downloads**، ونزّل البرنامج، واتبع خطوات التثبيت.

2 نزّل VS Code

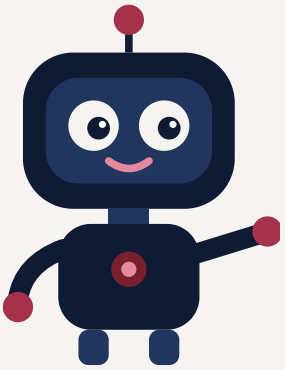
من **code.visualstudio.com**. افتحه، واضغط **Extensions**، وابحث عن **Python**، وثبّت إضافة Microsoft.

3 سلّم على بايثون

افتح **Terminal ← New Terminal** واكتب:

```
python --version    # Windows
python3 --version   # Mac / Linux
```

ظهر لك شيء مثل `Python 3.13` ؟ أنت جاهز.



نصيحة بِت

أصدقائي على ويندوز: أثناء التثبيت فعّلوا خيار **Add python.exe to PATH**. إن نسيته ستقضي منتصف الليل تبحث عن الأخطاء. اسألني كيف عرفت!

لا يوجد لابتوب الآن؟

ابحث عن "online Python editor"، وستقدر تشغّل كل أمثلة الكتاب من متصفح هاتفك.

سطر واحد.
برنامجك الأول.

أنشئ ملفاً اسمه `hello.py` في VS Code. النهاية `.py` تخبر الكمبيوتر أن هذا ملف بايثون. ثم اكتب:

HELLO.PY

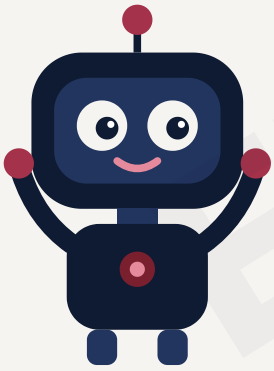
```
print(1)("Hello, world!"(2))
```

1. `print` أمر معناه: **"اعرض هذا على الشاشة".**
2. الكلام يُكتب بين **علامتي تنصيص**. هكذا يعرف بايثون أنه نص وليس أمراً.

اضغط زر ▶ أعلى الشاشة، و...

النتيجة

```
Hello, world!
```



بِت

لقد أمرت الكمبيوتر بشيء، **ونفّذه!** كل مبرمج على هذا الكوكب بدأ بهذا السطر بالضبط. أهلاً بك في النادي.



ركن الأخطاء

نسيت علامتي التنصيص؟ سيظن بايثون أن `Hello` اسم يجب أن يعرفه:

```
NameError: name 'Hello' is not defined
```

الحل: أعد النص بين علامتي التنصيص.

الفصل 01

01 أساسيات بايثون

كل تطبيق على هاتفك يفعل ثلاثة أشياء. وبعد هذا الفصل، برامجك ستفعلها أيضاً:

يعرض
يُظهر أشياء على الشاشة

يتذكر
يحتفظ بالمعلومات

يستمع
يأخذ إجابات المستخدم

`print` اعرض أي شيء، حتى الحسابات

ماذا يفعل؟
يعرض نصوصاً وأرقاماً ونتائج على الشاشة.

لماذا تحتاجه؟
هكذا يكلّمك برنامجك، وهكذا تتأكد أن الكود يعمل.

BASICS.PY

```
# Python skips this line. It's a note for you.  (1)
print("I'm learning Python!")
print("2 + 3 =", (2) 2 + 3)(3)
```

1. السطر الذي يبدأ بـ `#` هو **تعليق**: ملاحظة لك، وبايثون يتجاهلها.
2. **الفاصلة** تعرض أكثر من شيء في سطر واحد.
3. حسابات بدون علامتي تنصيص؟ بايثون **يحسبها** لك.

النتيجة

```
I'm learning Python!
2 + 3 = 5
```

دورك

كم دقيقة في الأسبوع؟ لا تستخدم الآلة الحاسبة، دع بايثون يحسبها: `print(7 * 24 * 60)`

المتغيّر صندوق عليه ملصق.



بِت

الملصق على الصندوق هو **الاسم**، وما بداخله هو **القيمة**. اذكر الاسم لاحقاً، وسيفتح لك بايثون الصندوق.

PROFILE.PY

```
name = "Sara"
age = 24
age = age + 1      # happy birthday!
print(name, "is now", age)
```

1. الاسم. ممنوع المسافات، فاكتب `user_name` وليس `user name`.
2. علامة `=` معناها **"ضع هذا بالداخل"**، وليست "يساوي" كما في الرياضيات.
3. القيمة. يمكنك تغييرها في أي وقت، كما فعلنا في عيد الميلاد.

النتيجة

```
Sara is now 25
```

4 أنواع يمكن أن يحملها الصندوق

"Sara"	24	1.68	True
`str`	`int`	`float`	`bool`
نص، دائماً بين علامتي تنصيص.	أرقام صحيحة.	أرقام عشرية.	نعم أو لا: True أو False.

استمع للمستخدم

اجعل برنامجك يدير محادثة.

برنامجك
What’s your name?

المستخدم
Omar

برنامجك
How old are you?

المستخدم
21

برنامجك
Hi Omar! In 10 years you’ll be 31.

FUTURE.PY

```
name = input(1)("What's your name? ")
age = int(2)(input("How old are you? "))
print(f"Hi {name}! In 10 years you'll be {age + 10}.")(3)
```

1. `input` يسأل سؤالاً و**ينتظر** المستخدم حتى يكتب.
2. الإجابات تصل دائماً كـ**نص**. و `int()` تحوّل النص إلى رقم لتقدر تحسب به.
3. ضع `f` قبل علامتي التنصيص، ثم أي متغير بين `{ }` ، وبايثون يملأ الفراغ.



ركن الأخطاء

النصوص والأرقام لا تختلط. `"5" + "5"` تعطيك `55` وليس `10` ، لأن بايثون يلصق النصوص ولا يجمعها. وإن نسيت `int()` سترى:

```
TypeError: can only concatenate str (not "int") to str
```