# مقدمة

هل فكرت يومًا كيف تُبنى التطبيقات، أو كيف تتحكم الشركات في كميات هائلة من البيانات؟ أو ربما تساءلت كيف تعمل برامج الذكاء الاصطناعي؟ الجواب غالبًا يكون: البرمجة.

ومن بين جميع لغات البرمجة، تُعد لغة بايثون (Python) واحدة من أسهل وأقوى اللغات التي يمكن أن تبدأ بها رحلتك في هذا العالم. فهي لغة مرنة وسهلة الفهم، وتُستخدم في كل شيء تقريبًا: بناء المواقع، تحليل البيانات، الذكاء الاصطناعي، وحتى الألعاب.

تم تصميم بايثون لتكون بسيطة من حيث الكتابة والقراءة، وهذا يجعلها رائعة للمبتدئين، وفي نفس الوقت قوية بما يكفي للمشاريع الكبيرة. ولهذا السبب، يعتمد عليها كبار الشركات مثل Google وNetflix وInstagram.

## لماذا هذا الكتاب؟

هذا الكتاب مخصص لكل من يرغب في تعلم البرمجة بلغة بايثون من الصفر، حتى لو لم يكن لديه أي خلفية تقنية. سنبدأ معًا من الأساسيات خطوة بخطوة، ثم ننتقل إلى مواضيع متقدمة، مع الكثير من الأمثلة العملية.

## الخطوط العريضة لمحتوى الكتاب:

- التعرف على لغة بايثون
- سنتعرف على بايثون، وكيفية تنصيبها، وكتابة أول برنامج.
- الأساسيات
- سنتعلم المتغيرات، الشروط، الحلقات، والدوال.
- التعامل مع البيانات
- سنشرح القوائم، القواميس، والسلاسل النصية وغيرها.
- البرمجة الكائنية (OOP)
- سنتعرف على مفهوم الكائنات، الفئات، والوراثة.
- مشاريع عملية
- سنبني معًا تطبيقات بسيطة توضح كيفية استخدام ما تعلمناه.

- مقدمة إلى مجالات بايثون المتقدمة
- مثل تحليل البيانات، الذكاء الاصطناعي، وتطوير الويب.

## المحتوى :


| العنوان | رقم الصفحة |
| --- | --- |
| الفصل الاول : مقدمة في البرمجة وبايثون<br>• ما هي البرمجة؟<br>• لماذا نختار بايثون؟<br>• تثبيت بايثون وتجهيز بيئة العمل (IDLE أو VS Code) | |
| الفصل الثاني : أول خطواتك مع بايثون<br>• طباعة الجمل (print)<br>• استخدام التعليقات<br>• كتابة أول برنامج | |
| الفصل الثالث : المتغيرات وأنواع البيانات<br>• المتغيرات (Variables)<br>• الأعداد، النصوص، القوائم، القواميس، المنطقيات<br>• التحويل بين الأنواع | |
| الفصل الرابع : العمليات والشروط<br>• العمليات الحسابية والمنطقية<br>• الشروط (if, elif, else)<br>• تمارين بسيطة | |
| الفصل الخامس: الحلقات (Loops)<br>• حلقة for<br>• حلقة while<br>• استخدام break و continue | |
| الفصل السادس: القوائم في بايثون<br>• ما هي القائمة؟<br>• الوصول إلى العناصر<br>• إضافة وحذف العناصر من القائمة<br>• التعامل مع القوائم باستخدام الحلقات | |

| العنوان | رقم الصفحة |
| --- | --- |
| • تغيير العناصر في القائمة<br>• الترتيب التصاعدي<br>• الترتيب التنازلي<br>• تمارين الفصل السادس | |
| الفصل السابع: Tuples و Dictionaries في بايثون<br>• Tuples<br>• Dictionaries<br>• تمارين الفصل السابع | |
| الفصل الثامن: التعامل مع المستخدم (الإدخال والإخراج)<br>• استقبال البيانات من المستخدم (input)<br>• عرض النتائج على الشاشة (print)<br>• تنسيق النصوص عند الطباعة | |
| الفصل التاسع: الدوال (Functions) في بايثون<br>• ما هي الدالة؟<br>• الدوال مع المعاملات (Parameters)<br>• دوال ترجع قيمة (Return)<br>• معاملات اختيارية (قيم افتراضية)<br>• أكثر من قيمة مرجعة<br>• لماذا نستخدم الدوال؟<br>• تمارين الفصل التاسع | |
| الفصل العاشر: الكلاسات (Classes) والكائنات (Objects)<br>• ما هو الـ Class؟<br>• إضافة دالة __init__<br>• الوراثة inheritance<br>• تمارين الفصل العاشر | |
| الفصل الحادي عشر: التعامل مع الملفات في بايثون (File Handling)<br>• أنواع العمليات على الملفات<br>• فتح ملف: open<br>• قراءة من ملف | |

| العنوان | رقم الصفحة |
| --- | --- |
| • الكتابة في ملف<br>• استخدام with (الطريقة الأفضل)<br>• تمارين الفصل الحادي عشر | |
| الفصل الثاني عشر: التعامل مع الأخطاء (Exceptions Handling)<br>• ما هو الاستثناء (Exception)؟<br>• استخدام try و except<br>• لماذا نستخدم التعامل مع الأخطاء؟<br>• تمارين الفصل الثاني عشر | |
| الفصل الثالث عشر: المكتبات (Libraries) في بايثون<br>• ما هي المكتبة؟<br>• استيراد مكتبة: import<br>• مكتبة random - التوليد العشوائي<br>• مكتبة datetime - الوقت والتاريخ<br>• مكتبة time - الانتظار والزمن<br>• تثبيت مكتبة خارجية: pip<br>• تمارين الفصل الثالث عشر | |
| خاتمة | |
| اسئلة متنوعة | |
| المصادر | |

# الفصل الأول

# مقدمة في البرمجة وبايثون

# الفصل الأول: مقدمة في البرمجة وبايثون

## ما هي البرمجة؟

البرمجة ببساطة هي طريقة نستخدمها لإعطاء الأوامر للحاسوب ليقوم بأشياء معينة. مثلًا، يمكننا أن نطلب منه أن يجمع لنا رقمين، أو يعرض رسالة، أو يخزن بيانات، أو حتى يبني لعبة كاملة!

الكمبيوتر لا يفهم لغتنا كبشر، لكنه يفهم اللغات البرمجية التي نكتب بها الأوامر بطريقة معينة ومنظمة.

## لماذا أتعلم البرمجة؟

- البرمجة أصبحت اليوم مهارة مهمة في كثير من المجالات مثل:
- تطوير تطبيقات الهاتف والمواقع
- تحليل البيانات والإحصائيات
- الذكاء الاصطناعي والتعلم الآلي
- أتمتة المهام وتوفير الوقت
- إنشاء ألعاب إلكترونية

وأكثر بكثير!

والأجمل من هذا؟ أن تعلم البرمجة يشبه تعلّم لغة جديدة، كلما تمرّنت عليها، صارت أسهل وأمتع!

## لماذا نختار لغة بايثون؟

بايثون (Python) هي واحدة من أسهل اللغات البرمجية من حيث الفهم والكتابة، لذلك ينصح بها كثير من المبرمجين للمبتدئين. إليك بعض مميزاتها:

- سهلة القراءة والكتابة: أقرب للغة الإنسان من لغات برمجة أخرى.
- مجانية ومفتوحة المصدر.
- تعمل على جميع أنظمة التشغيل (ويندوز، لينكس، ماك).
- تُستخدم في مجالات كثيرة: تحليل البيانات، الذكاء الاصطناعي، الويب، الألعاب، وغيرها.
- مجتمع ضخم من المستخدمين: أي مشكلة تواجهك، ستجد لها حلولًا بسهولة.

## كيف أبدأ ببايثون؟

لبدء استخدام بايثون، تحتاج إلى خطوتين بسيطتين:

### 1. تثبيت بايثون

- ادخل إلى الموقع الرسمي: https://www.python.org
- اختر "Download Python" حسب نظام التشغيل الذي تستخدمه.

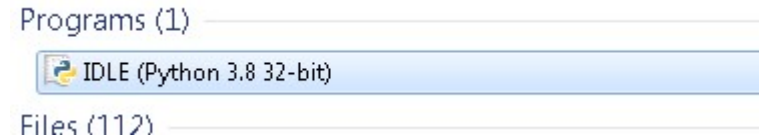
- بعد تحميل الملف، افتحه واتبع خطوات التثبيت.

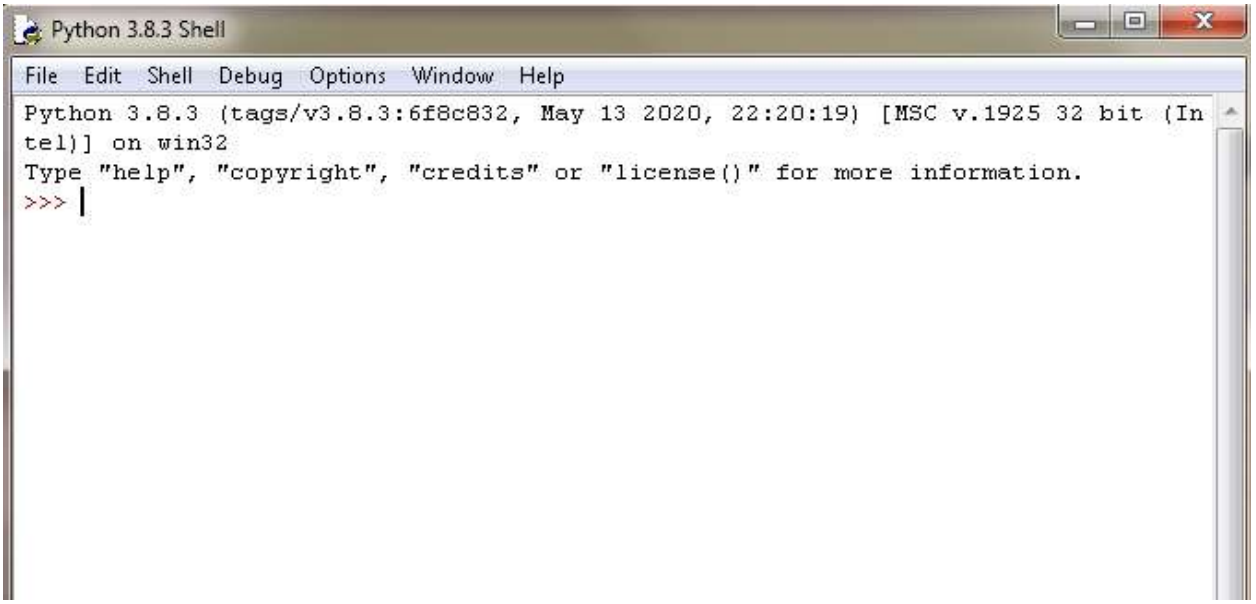
- تأكد من تفعيل خيار "Add Python to PATH" أثناء التثبيت.

## 2. اختيار برنامج لكتابة الكود

يمكنك كتابة الكود في عدة برامج، وأشهرها:

- IDLE: يأتي مع بايثون تلقائيًا.




- VS Code: محرر قوي وسهل الاستخدام.

- Thonny: موجه للمبتدئين وسهل جدًا.

سنبدأ في هذا الكتاب باستخدام IDLE لأنه بسيط ومناسب للبداية.

### كتابة أول برنامج لك

افتح برنامج IDLE ثم من القائمة اختر File→New File

### وأكتب السطر التالي

```
print("Hello World!!")
```

ثم من القائمة اختر Run -->Run Module أو قم بالضغط على مفتاح F5 .

### ستظهر لك الرسالة

```
Hello World!!
```

### مبروك 🎉 لقد كتبت أول برنامج بلغة بايثون!

### 🩹 أمثلة بسيطة على أوامر بايثون

### طباعة جمل مختلفة

```
print("Hello World!!")
print("Programming is Fun!!")
print("Python is Wonderfull")
```

```
```

## طباعة أرقام وعمليات حسابية

```
print(3 + 5)

print(2 - 10)

print(2 * 4)

print(3 / 9)
```

## طباعة أكثر من شيء في نفس السطر

```
print(6 + 4 ,"مجموع 4 + 6 هو")
```

## رسائل الخطأ ومعناها .

```
>>> print(hello world)
SyntaxError: invalid syntax
```

1- Syntax Error

تعني هذه الرسالة بأنه يوجد خطأ في طريقة كتابة الاوامر

ولتصحيح الخطأ السابق نكتب

```
print("Hello World")
```

كما نرى بأن الخطأ كان بسبب عدم وضع علامتى التنصيص- علامة التنصيص تخبر البايثون بان هذه رسالة نصية .

```
>>> print("Hello World!!)

SyntaxError: EOL while scanning string literal
```

في الكود السابق فان المبرمج نسى أن يقوم باغلاق الرسالة بعلامة التنصيص

في نهاية الجملة- فهي لازمة لاكمال النص .

## كتابة سطور متعددة وطباعتها . (''') Multiline String

يمكننا طباعة سطور متعددة باستخدم علامة التنصيص الفردية 3 مرات قبل بداية الجملة واكمال النص واغلاقة ايضا ب3 علامات تنصيص فردية كما هو موضح بالشكل التالي .

```
>>> print('''This is to test
multi line messages
using triple single quotes ''')
This is to test
multi line messages
using triple single quotes
>>>
```

## تكرار السلاسل الحرفية :

ماذا اذا اردنا تكرار حرف أو جملة عدد من المرات .

يمكننا استخدام علامة (*) لتكرار الجمل المكلوب طباعتها كما يلي .

```
>>> print(10*'ABC-')
ABC-ABC-ABC-ABC-ABC-ABC-ABC-ABC-ABC-ABC-
>>>
```

## تمارين بسيطة (جرّبها بنفسك!)

### تمرين 1:

اكتب كودًا يطبع اسمك، وعمرك، وهوايتك المفضلة.

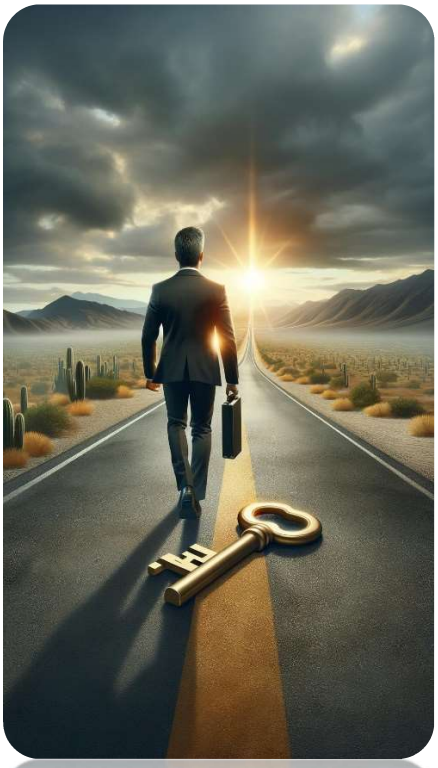
### تمرين 2:

اطبع العملية التالية: "إذا كان عندك 3 تفاحات واشتريت 2، كم يكون المجموع؟"

حاول أن تطبع  النتيجة: مع شرح بسيط.

### تمرين 3:

اكتب كودًا يطبع نتيجة جمع رقمين من اختيارك



# الفصل الثاني

# أول خطواتك مع بايثون

# الفصل الثاني: أول خطواتك مع بايثون

في هذا الفصل، سنبدأ بالتعرف على الأساسيات التي نحتاجها لكتابة برامج بلغة بايثون. لا تقلق إذا كانت هذه أول مرة تبرمج فيها، سنتدرّج خطوة بخطوة.

## ✨ 1. الدالة ()print: صديقك الأول

كما رأينا في الفصل السابق، نستخدم الدالة print لعرض أي شيء على الشاشة. يمكننا طباعة نصوص، أرقام، أو حتى نتائج عمليات حسابية.

```
print("أهلًا وسهلًا!")

print(2025)

print(3 + 5 ,"= 3 + 5")
```

## 💬 2. التعليقات (Comments)

التعليقات هي جمل نكتبها في الكود، لكنها لا تُنفذ. نستخدمها لشرح ما يفعله الكود، وتبدأ بعلامة #.

```
#هذا السطر يطبع رسالة ترحيب

print("مرحبًا بك في بايثون")
```

التعليقات مهمة جدًا، خصوصًا عندما تبدأ مشاريع طويلة، أو لمراجعة كودك لاحقًا.

## 📦 3. المتغيرات (Variables)

المتغير هو اسم نختاره لتخزين معلومة نستخدمها لاحقًا. مثل صندوق نضع فيه رقم أو كلمة.

```
name="Samy"

age=16

print("Name is : ",name)

print("age is : ", age)
```

## أنواع البيانات الشائعة:

| النوع | مثال | الاستخدام |
|---|---|---|
| نصوص (Strings) | “Welcome” | كلمات وجمل. |
| أعداد صحيحة (Integers) | 10 | عد الارقام |
| اعداد عشرية (Floats) | 3.14 | قيم غير صحيحة |
| منطقية (Boolean) | True or False | منطقية |

## 4. العمليات الحسابية

يمكنك استخدام بايثون كآلة حاسبة!

| العملية | مثال | الاستخدام |
|---|---|---|
| + | 5+2 | الجمع |
| - | 3-7 | الطرح |
| * | 5*9 | الضرب |
| / | 8/2 | القسمة |
| ** | 2**3 | الاس (قوة) |
| // | 7//2 | القسمة الصحيحة |
| % | 5%2 | باقي القسمة |

## مثال عملي :

```
num1=10

num2=3

print("Addition ",num1+num2)

print("Subtract",num1-num2)

print("Divide ",num1/num2)

print("Reminder",num1%num2)
```

الناتج :

```
Addition  13

Subtract 7

Divide  3.3333333333333335

Reminder 1
```

## تمارين الفصل الثاني

### تمرين 1:

اكتب كودًا يخزن اسمك وعمرك في متغيرين، ثم يطبع الجملة:

### تمرين 2:

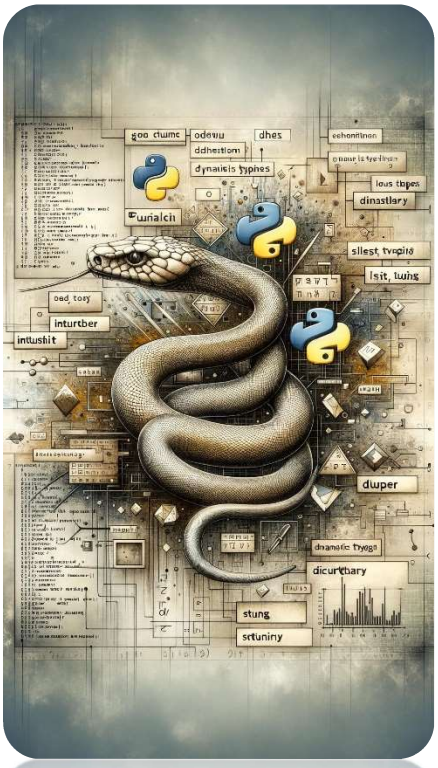
اختر رقمين، واطبع نتائج العمليات الأربعة (الجمع، الطرح، الضرب، القسمة) بينهما.

### تمرين 3:

اكتب تعليقًا بجانب كل سطر يشرح ما يفعله الكود التالي:

### تمرين 4 (تحدي صغير):

احسب مساحة مستطيل طوله 8 وعرضه 5، واطبع  النتيجة:.



# الفصل الثالث

# المتغيرات وأنواع البيانات

# الفصل الثالث: المتغيرات وأنواع البيانات

في هذا الفصل، سنتعرف أكثر على المتغيرات، وأنواع البيانات التي يمكن تخزينها داخلها، مثل النصوص، الأرقام، والقوائم. المتغيرات هي جزء أساسي من أي برنامج، وسنستخدمها في كل خطوة لاحقة.

## 📦 ما هو المتغير؟

المتغير هو اسم نستخدمه لتخزين قيمة (مثل رقم أو نص)، ويمكننا استخدامه لاحقًا في العمليات أو الطباعة أو أي شيء آخر.

مثال:

```
name="Samy"

age=16

print("Name is : ",name)

print("age is : ", age)
```

في هذا المثال:

- "Samy" هي قيمة نصية.
- 16 هي قيمة عددية.

## 🧠 قواعد تسمية المتغيرات

- يجب أن يبدأ بحرف أو شرطة سفلية _ (وليس رقمًا).
- لا يمكن أن يحتوي على فراغات (استخدم _ بدلًا من المسافة).
- لا تستخدم كلمات محجوزة مثل if, for, print.