

الوحدة 6: الدوال في بايثون (Python Functions)

الدالة هي كتلة من الأوامر البرمجية لا تعمل إلا عندما يتم استدعاؤها.

يمكن للدالة أن تُرجع قيمة (return)، كما تساعد في تجنب تكرار الكود في البرامج.

أ. إنشاء دالة (Creating a Function)

يتم تعريف الدالة في بايثون باستخدام الكلمة المفتاحية:

```
def
```

ثم اسم الدالة، ثم قوسان.

مثال:

```
def my_function():  
  
    print("Hello from a function")
```

هذا المثال ينشئ دالة اسمها my_function، وعند استدعائها تطبع:

```
Hello from a function
```

ملاحظة مهمة:

الكود داخل الدالة يجب أن يكون مزاحًا بمسافة (Indentation) لأن بايثون تعتمد على المسافة لتحديد الكتل البرمجية.

ب. استدعاء الدالة (Calling a Function)

لا تعمل الدالة إلا إذا تم استدعاؤها باستخدام اسمها متبوعًا بقوسين:

مثال:

```
def my_function():
```

```
print("Hello from a function")  
  
my_function()
```

يمكن استدعاء الدالة عدة مرات:

```
def my_function():  
  
    print("Hello from a function")  
  
my_function()  
  
my_function()  
  
my_function()
```

ج. قواعد تسمية الدوال (Function Names)

أسماء الدوال تتبع نفس قواعد تسمية المتغيرات في بايثون:

- ✓ يجب أن يبدأ الاسم بحرف أو شرطة سفلية _
- ✓ يمكن أن يحتوي الاسم على حروف وأرقام وشرطة سفلية فقط
- ✓ الأسماء حساسة لحالة الأحرف

مثل:

myFunction يختلف عن myfunction

أمثلة لأسماء صحيحة:

- ✓ calculate_sum()
- ✓ _private_function()
- ✓ myFunction2()

من الأفضل اختيار أسماء تصف وظيفة الدالة بوضوح.

د. لماذا نستخدم الدوال؟ (Why Use Functions?)

بدون الدوال قد نضطر لتكرار نفس الكود مرات عديدة. تخيل أنك بحاجة لتحويل درجات الحرارة من فهرنهايت إلى مئوية عدة مرات في برنامجك. بدون دوال، ستحتاج إلى كتابة نفس الكود مرارًا وتكرارًا:

مثال بدون دوال، تكرار مزعج:

```
temp1 = 77

celsius1 = (temp1 - 32) * 5 / 9

print(celsius1)

temp2 = 95

celsius2 = (temp2 - 32) * 5 / 9

print(celsius2)

temp3 = 50

celsius3 = (temp3 - 32) * 5 / 9

print(celsius3)
```

باستخدام الدوال يصبح الكود نظيف وقابل لإعادة الاستخدام:

```
def fahrenheit_to_celsius(fahrenheit):

    return (fahrenheit - 32) * 5 / 9

print(fahrenheit_to_celsius(77))

print(fahrenheit_to_celsius(95))

print(fahrenheit_to_celsius(50))
```

هـ. قيم الإرجاع (Return Values)

يمكن للدالة إرجاع قيمة باستخدام كلمة:

```
return
```

عندما تصل الدالة إلى `return` تتوقف عن التنفيذ وترجع القيمة.

مثال:

```
def get_greeting():  
  
    return "Hello from a function"  
  
message = get_greeting()  
  
print(message)
```

يمكن استخدام القيمة المُعادة مباشرة:

```
def get_greeting():  
  
    return "Hello from a function"  
  
print(get_greeting())
```

إذا لم تحتوي الدالة على `return` فإنها تُرجع القيمة `None` تلقائيًا.