

What is Python?

INTRODUCTION TO PYTHON FOR DEVELOPERS



George Boorman
Curriculum Manager, DataCamp



What we will learn

- **No prior knowledge required**
- What Python is and why it is popular for programming
- How to execute code using the DataCamp UI
- Code comments and data types
- Performing calculations
- Creating and manipulating variables
- Building conditional workflows
- For and while loops

What is Python?



- Open-source: free to use
- Packages: use other's code to avoid starting from scratch
- Syntax resembles natural language

The swiss army knife of programming languages

Use-cases



- Task automation
- Web apps
- Artificial Intelligence (AI)
 - Web scraping
 - Content generation/summary
 - Image recognition

¹ <https://unsplash.com/@dmjdenise>

Python in Big Tech



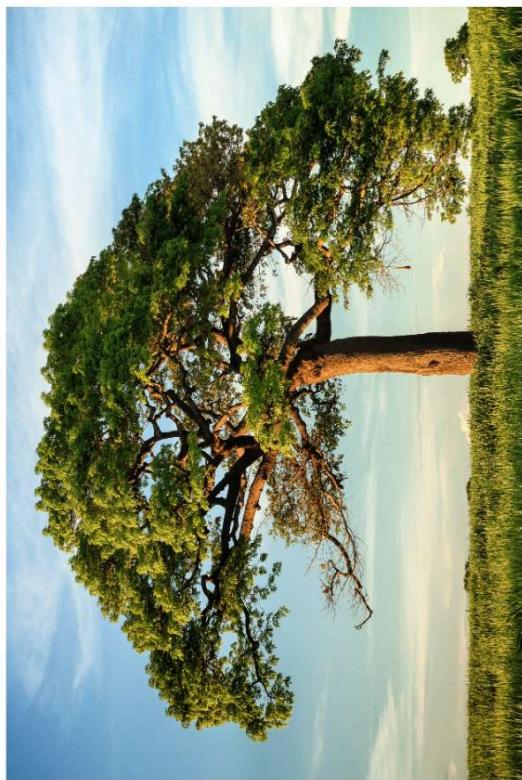
NETFLIX

¹ <https://instagram-engineering.com/what-powers-instagram-hundreds-of-instances-dozens-of-technologies-adf2e22da2ad>



INTRODUCTION TO PYTHON FOR DEVELOPERS

Not always fit-for-purpose



- Difficult to cut down with a swiss army knife
- Easier with an axe or chainsaw
- Not suitable:
 - Mobile apps - Swift or Kotlin
 - Hardware programming - C

¹ <https://unsplash.com/@theoneoffussocialclub>

 datacamp

INTRODUCTION TO PYTHON FOR DEVELOPERS

Speed of development

script.py

```
1 my_age = 34
```

- Uses natural language
- Easier to create and understand code

Speed of development



- Python is slow(er) than some other languages
- But Python is easier to use
 - so it can be quicker for development

Let's practice!

INTRODUCTION TO PYTHON FOR DEVELOPERS

How to run Python code

INTRODUCTION TO PYTHON FOR DEVELOPERS



George Boorman
Curriculum Manager, DataCamp



The DataCamp interface

The screenshot shows the DataCamp Python exercise interface. At the top, there's a navigation bar with 'Exercise' and 'Light Mode'. Below it, the code editor shows a script named 'script.py' with the following content:

```
script.py
1 # Addition
2 print(4 + 5)
3
4 # Subtraction
5 print
6 print
7 # Multiplication
8
9
10 # Division
11
```

Below the code editor, there's a yellow '100 XP' badge. To the right, a green 'Submit Answer' button is visible. On the left, under 'Instructions', there's a bulleted list of tasks:

- Print the sum of `4 + 5`.
- Print the result of subtracting `5` from `5`.
- Multiply `5` by `5`.
- Divide `10` by `2`.

At the bottom, there's a 'Take Hint (-50 XP)' button. On the far right, a terminal window titled 'Python Shell' shows the command 'In [1]:'.

IPython Shell

The screenshot shows a Python exercise interface. At the top, there are navigation buttons for 'Learn / Courses / Introduction to Python' and a 'Course Outline'. Below this is a toolbar with icons for back, forward, and search, along with a 'Light Mode' switch.

The main area is titled 'Exercise' and contains the following content:

- script.py**:

```
1 # Addition
2
3 # Subtraction
4
5 # Multiplication
6
7 # Division
8
9
10
11
```
- Python as a calculator**:

Python is perfectly suited to do basic calculations. It can do addition, subtraction, multiplication and division.
- The code in the script gives some examples.
- Now it's your turn to practice!
- Instructions**:

100 XP

 - Print the sum of `5 + 5`.
 - Print the result of subtracting `5` from `5`.
 - Multiply `3` by `5`.
 - Divide `10` by `2`.
- Take Hint (-30 XP)**

To the right of the main content is an 'IPython Shell' window with the text 'In [1]:' and a large green rectangular highlight.

Python Script

The screenshot shows a web-based Python script editor interface. At the top, there are navigation icons: back, forward, course outline, and a search bar. On the right, there are light mode/dark mode switches and a 'Submit Answer' button.

script.py

```
1 # Addition
2
3 # Subtraction
4
5
6 # Multiplication
7
8
9 # Division
10
11
```

Exercise

Python as a calculator

Python is perfectly suited to do basic calculations. It can do addition, subtraction, multiplication and division.

The code in the script gives some examples.

Now it's your turn to practice!

Instructions

100 XP

- Print the sum of `5 + 5`.
- Print the result of subtracting `5` from `5`.
- Multiply `3` by `5`.
- Divide `10` by `2`.

Take Hint (-30 XP)

IPython Shell

In [1]:

Code comments

```
# This is a code comment
```

- Code is read by more than it is written
- Code is shared with others
- Comments help people (including us) to understand our code



Performing calculations

The screenshot shows a DataCamp exercise interface. At the top left, there's a yellow button labeled "100 XP". Below it, a grey bar contains the word "Instructions". The main area contains a list of tasks:

- Print the sum of `5 + 5`.
- Print the result of subtracting `5` from `5`.
- Multiply `3` by `5`.
- Divide `10` by `2`.

Below the tasks is a yellow button labeled "Take Hint (-50 XP)". To the right of the tasks are three buttons: "Run Code" (grey), "Submit Answer" (green), and a small icon (grey). On the far right, there's an "IPython Shell" section with the text "In [1]:".

Performing calculations

The screenshot shows a Python exercise interface. At the top, there are navigation buttons for 'Learn', 'Courses', and 'Introduction to Python'. Below that, it says 'Exercise' and 'script.py'. The main area contains the following text:

```
Python as a calculator
Python is perfectly suited to do basic calculations. It can do addition, subtraction, multiplication and division.
```

The code in the script gives some examples:

```
Now it's your turn to practice!
```

Instructions (100 XP)

- Print the sum of `4 + 5`.
- Print the result of subtracting `5` from `5`.
- Multiply `3` by `5`.
- Divide `10` by `2`.

Take Hint (-30 XP)

Python Shell

In [1]:

At the bottom right, there are 'Run Code' and 'Submit Answer' buttons. The 'Submit Answer' button is highlighted in green.

- Use `print()` to generate output from script

More calculations

Exercise

Python as a calculator

Python is perfectly suited to do basic calculations. It can do addition, subtraction, multiplication and division.

The code in the script gives some examples.

Now it's your turn to practice!

script.py

```
1 # Addition
2 print(4 + 5)
3
4 # Subtraction
5 print
6 print
7 # Multiplication
8
9
10 # Division
11
```

Instructions

Take Hint (-50 XP)

Run Code

Submit Answer

Python Shell

In [1]:

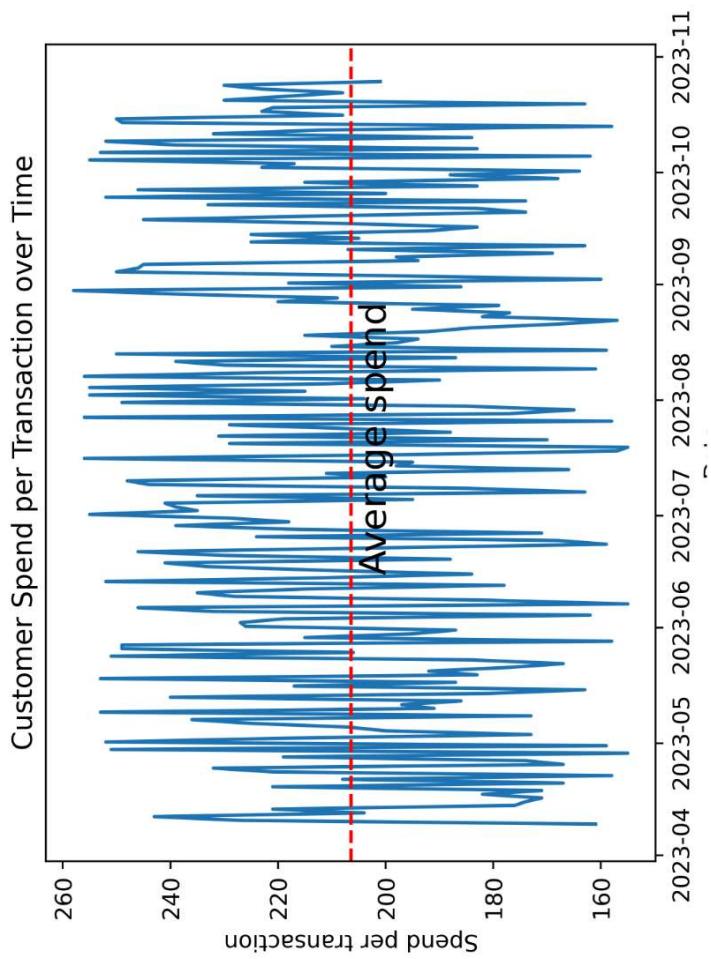
Calculations cheat sheet

Syntax	Action	Example	Output
*	Multiply	4 * 10	40
+	Addition	7 + 9	16
-	Subtract	23 - 4	19
/	Division	27 / 3	9
**	Power	3 ** 2	9
%	Modulo	7 % 4	3



Numbers in programming

- Age verification
- Average value over a time period



Let's practice!

INTRODUCTION TO PYTHON FOR DEVELOPERS

Variables and data types

INTRODUCTION TO PYTHON FOR DEVELOPERS



George Boorman
Curriculum Manager, DataCamp



Variables

```
# Create the customer_age variable with a value of 25  
customer_age = 25
```

- **variable_name** : Name of the variable (case-sensitive)

- **=** : For assigning a value
- **value** : The value to assign to the variable

Why use variables?

- Avoid having to retype the same information
- Can update variables if conditions change

```
# Original value of customer_age  
customer_age = 25  
  
# Update customer_age to 26  
customer_age = 26
```



Integers

```
# This is an integer
```

```
180
```

```
# This is also an integer
```

```
customer_id = 180
```

- No need to tell Python what data type the variable is!

```
type(customer_id)
```

```
<class 'int'>
```



FLOATS

```
# Decimal values  
customer_average_spend = 55.28  
  
# Check the data type  
type(customer_average_spend)
```

```
<class 'float'>
```



Strings

```
# Single quotes
customer_name = 'George Boorman'

# Double quotes also works
customer_name = "George Boorman"

# Check the data type
type(customer_name)
```

```
<class 'str'>
```



Booleans

```
# Define active and inactive customer variables  
active_customer = True  
inactive_customer = False
```

```
# Check active_customer data type  
type(active_customer)
```

```
<class 'bool'>
```

- Case sensitive: true and false won't work

Variable naming

```
# Define total_spend
```

```
total_spend = 3150.96
```

```
# Try to define using a space
```

```
total spend = 3150.96
```

```
SyntaxError: invalid syntax
```

```
# Try to name a variable starting with a number
```

```
2023_spend = 3150.96
```

```
SyntaxError: invalid decimal literal
```

Case conventions

```
# Snake case
```

```
total_spend = 3150.96
```

```
# CamelCase
```

```
TotalSpend = 3150.96
```

- Both are acceptable - use personal preference

Calculations with variables

```
# Define num_transactions  
num_transactions = 57  
  
# Calculate total_spend by multiplying two variables  
total_spend = num_transactions * customer_average_spend  
  
# Print two variables  
print(customer_name, total_spend)
```

George Boorman 3150.96



Let's practice!

INTRODUCTION TO PYTHON FOR DEVELOPERS