

# Hello Python!

INTRODUCTION TO PYTHON



Hugo Bowne-Anderson

Data S

# Exercise

Exercise < Light Mode

## Your first Python code

It's time to run your first Python code!

Head to the code and hit the run code button to see the output.

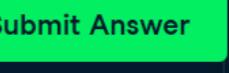
Instructions 100 XP

- Hit the run code button to see the output of `print(5 / 8)`.

Take Hint (-30 XP)

`script.py`

```
1 # Hit run code to see the output!
2 print(5 / 8)
```

 Run Code  Submit Answer

IPython Shell Slides

In [1]:

# How you will learn

The screenshot shows a learning interface for a Python course. On the left, a sidebar titled "Exercise" contains the heading "Python as a calculator". It explains that Python is suited for basic calculations and provides examples of addition, subtraction, multiplication, and division. Below this, a list of tasks is provided, and a "Take Hint (-30 XP)" button is available. The main area features a code editor titled "script.py" with the following code:

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

The code editor includes navigation buttons (←, →), a "Course Outline" link, and a "Light Mode" toggle. Below the code editor is a "Python Shell" window showing the command "In [1]:". At the bottom right are "Run Code" and "Submit Answer" buttons.

# Python



- General purpose: build anything
- Open source! Free!
- Python packages, also for data science
  - Many applications and fields

# IPython Shell

## Execute Python commands

The screenshot shows a Python exercise interface. On the left, there's a sidebar with navigation links: 'Learn / Courses / Introduction to Python'. Below that is a section titled 'Exercise' with the title 'Python as a calculator'. It contains text about Python being suited for basic calculations and examples of addition, subtraction, multiplication, and division. A 'Instructions' section lists four tasks: 'Print the sum of 5 + 5.', 'Print the result of subtracting 5 from 5.', 'Multiply 3 by 5.', and 'Divide 10 by 2.'. A 'Take Hint (-30 XP)' button is available. On the right, the main area has a dark theme with a 'script.py' file open. The code is as follows:

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

Below the code are buttons for 'Run Code' and 'Submit Answer'. At the bottom, there's an 'IPython Shell' section with the prompt 'In [1]:'.

## Execute Python commands

# IPython Shell

The screenshot shows a Python exercise interface. At the top, there's a navigation bar with 'Learn / Courses / Introduction to Python' and a 'Course Outline' button. Below that is a section titled 'Exercise' with a sub-section 'Python as a calculator'. It contains text about Python being suited for calculations and some examples. A 'Light Mode' toggle is also present. On the left, there's a 'Instructions' section with a '100 XP' badge and a list of tasks: 'Print the sum of 5 + 5.', 'Print the result of subtracting 5 from 5.', 'Multiply 3 by 5.', and 'Divide 10 by 2.'. A 'Take Hint (-30 XP)' button is available. The main area shows a code editor with a script named 'script.py' containing code for addition, subtraction, multiplication, and division. At the bottom are 'Run Code' and 'Submit Answer' buttons. A large green box highlights the 'IPython Shell' window at the bottom, which has 'In [1]:' and a text input field.

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

In [1]:

# IPython Shell

The screenshot shows a Python exercise interface. On the left, there's a sidebar with navigation links: 'Learn / Courses / Introduction to Python' and a 'Course Outline' button. Below that is a section titled 'Exercise' with the sub-section 'Python as a calculator'. It contains text about Python being suited for basic calculations and some examples. A 'Instructions' section lists four tasks: 'Print the sum of 5 + 5.', 'Print the result of subtracting 5 from 5.', 'Multiply 3 by 5.', and 'Divide 10 by 2.'. A 'Take Hint (-30 XP)' button is available. On the right, there's a main workspace. At the top, it says 'script.py' and has a 'Light Mode' toggle. The code editor shows the first line of code: '1'. Below the editor are three buttons: a blue circular arrow, a grey 'Run Code' button, and a green 'Submit Answer' button. At the bottom, there's an 'IPython Shell' section with the text 'In [1]:'.

# Python Script

- Text files - .py
- List of Python commands
- Similar to typing in IPython Shell

The screenshot shows a Python script editor interface. The main area displays a file named `script.py` with the following content:

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

The code editor has a green border around the script area. Below the editor is an `IPython Shell` window showing the prompt `In [1]:`.

On the left side of the interface, there is a sidebar with the following sections:

- Exercise**:
  - Python as a calculator**: A brief introduction stating "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
  - 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`.
- Take Hint (-30 XP)**

# Python Script

The screenshot shows a Python script exercise interface. At the top left, there's a navigation bar with icons for 'Learn', 'Courses', and 'Introduction to Python'. Below it, a sidebar on the left is titled 'Exercise' and contains sections for 'Python as a calculator', 'Instructions', and a 'Take Hint (-30 XP)' button. The main area has a dark theme with a light header bar showing 'script.py', a 'Light Mode' toggle, and a 'Run Code' button. The code editor shows the first line of a Python script: '1 4'. Below the editor is an 'IPython Shell' window with the prompt 'In [1]:'. The 'Instructions' section contains four tasks:

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

A yellow '100 XP' badge is visible above the instructions. The 'Submit Answer' button is highlighted in green at the bottom right of the editor area.

# Python Script

The screenshot shows a Python script exercise interface. At the top left, there's a navigation bar with 'Learn / Courses / Introduction to Python'. Below it, a sidebar on the left contains sections like 'Exercise' (selected), 'Python as a calculator' (with a brief description), 'Instructions' (with a yellow '100 XP' badge), and a 'Take Hint (-30 XP)' button. The main area has a dark theme with a 'script.py' file open. The code in the file is just '1'. To the right of the code editor are three buttons: a blue 'Run Code' button, a white 'Submit Answer' button with a green border, and a grey 'Copy' button. At the bottom, there's an 'IPython Shell' section labeled 'In [1]:'.

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

The screenshot shows a Python exercise interface. At the top left, there's a navigation bar with a logo, 'Learn / Courses / Introduction to Python'. Below it, a sidebar on the left is titled 'Exercise' and contains sections for 'Python as a calculator', 'Instructions', and a 'Take Hint (-30 XP)' button. The main area has tabs for 'script.py' and 'IPython Shell'. The 'script.py' tab is active, displaying the following code:

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

Below the code editor are buttons for 'Run Code' and 'Submit Answer'. The 'IPython Shell' tab shows the prompt 'In [1]:'.

# Exercise

## 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 `4 + 5`.
- Print the result of subtracting `5` from `5`.
- Print the result of multiplying `3` by `5`.
- Print the result of dividing `10` by `2`.

 Take Hint (-30 XP)

#### script.py

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

 Light Mode



Run Code

Submit Answer

IPython Shell

Slides

In [1]:

# Let's practice!

INTRODUCTION TO PYTHON

## Variables and Types

INTRODUCTION TO PYTHON



Hugo Bowne-Anderson  
Data Scientist at DataCamp

# Variable

- Specific, case-sensitive name
  - Call up value through variable name
- 1.79 m - 68.7 kg

```
height = 1.79
```

```
weight = 68.7
```

```
height
```

```
1.79
```

# Calculate BMI

```
height = 1.79
```

```
weight = 68.7
```

```
height
```

```
1.79
```

weight

BMI = \_\_\_\_\_height<sup>2</sup>

```
68.7 / 1.79 ** 2
```

```
21.4413
```

```
weight / height ** 2
```

```
21.4413
```

```
21.4413
```

```
bmi = weight / height ** 2  
bmi
```

# Reproducibility

```
height = 1.79  
weight = 68.7  
bmi = weight / height ** 2  
print(bmi)
```

```
21.4413
```

# Reproducibility

```
height = 1.79  
weight = 74.2 # <-  
bmi = weight / height ** 2  
print(bmi)
```

```
23.1578
```

# Python Types

```
type(bmi)
```

float

```
day_of_week = 5  
type(day_of_week)
```

int

# Python Types (2)

```
x = "body mass index"  
y = 'this works too'  
type(y)
```

```
str
```

```
z = True  
type(z)
```

bool

## Python Types (3)

2 + 3

5

'ab' + 'cd'

'abcd'

- Different type = different behavior!

# Let's practice!

## INTRODUCTION TO PYTHON

**Exercise**

### Variable Assignment

In Python, a variable allows you to refer to a value with a name. To create a variable `x` with a value of `5`, you use `=`, like this example:

```
x = 5
```

You can now use the name of this variable, `x`, instead of the actual value, `5`.

Remember, `=` in Python means *assignment*, it doesn't test equality! Try it in the exercise by replacing `_____` with your code.

**Instructions** 100 XP

- Create a variable `savings` with the value of `100`.
- Check out this variable by typing `print(savings)` in the script.

**Take Hint (-30 XP)**

**script.py**

```
1 # Create a variable savings
2 savings = 100
3
4 # Print out savings
5 print(savings)
```

**Run Code** **Submit Answer**

**IPython Shell** **Slides**

In [1]:

## Exercise

Light Mode

### Calculations with variables

You've now created a savings variable, so let's start saving!

Instead of calculating with the actual values, you can use variables instead.

How much money would you have saved four months from now, if you saved \$10 each month?

#### Instructions

100 XP

- Create a variable `monthly_savings`, equal to `10` and `num_months`, equal to `4`.
- Multiply `monthly_savings` by `num_months` and assign it to `new_savings`.
- Print the value of `new_savings`.

 Take Hint (-30 XP)

script.py

```
1 # Create the variables monthly_savings and num_months
2
3
4 monthly_savings = 10
5 num_months = 4
6 # Multiply monthly_savings and num_months
7 new_savings = monthly_savings *num_months
8
9 # Print new_savings
10 print(new_savings)
```



Run Code

Submit Answer

IPython Shell

Slides

In [1]:

## Exercise

Light Mode

### Other variable types

In the previous exercise, you worked with the integer Python data type:

- `int`, or integer: a number without a fractional part. `savings`, with the value `100`, is an example of an integer.

Next to numerical data types, there are three other very common data types:

- `float`, or floating point: a number that has both an integer and fractional part, separated by a point. `1.1`, is an example of a float.
- `str`, or string: a type to represent text. You can use single or double quotes to build a string.
- `bool`, or boolean: a type to represent logical values. It can only be `True` or `False` (the capitalization is important!).

#### Instructions

100 XP

- Create a new float `half` with the value `0.5`

script.py

```
1 # Create a variable half
2
3 half = 0.5
4 # Create a variable intro
5 intro = "Hello! How are you?"
6
7 # Create a variable is_good
8 is_good = True
```



Run Code

Submit Answer

IPython Shell

Slides

In [1]:

## Exercise

Next to numerical data types, there are three other very common data types:

- `float`, or floating point: a number that has both an integer and fractional part, separated by a point. `1.1`, is an example of a float.
- `str`, or string: a type to represent text. You can use single or double quotes to build a string.
- `bool`, or boolean: a type to represent logical values. It can only be `True` or `False` (the capitalization is important!).

### Instructions

100 XP

- Create a new float, `half`, with the value `0.5`.
- Create a new string, `intro`, with the value `"Hello! How are you?"`.
- Create a new boolean, `is_good`, with the value `True`.

 Take Hint (-30 XP)

## script.py

```
1 # Create a variable half
2
3 half = 0.5
4 # Create a variable intro
5 intro = "Hello! How are you?"
6
7 # Create a variable is_good
8 is_good = True
```



Run Code

Submit Answer

IPython Shell Slides

In [1]:

## Exercise

Light Mode

### Operations with other types

Variables come in different types in Python. You can see the type of a variable by using `type()`. For example, to see type of `a`, execute: `type(a)`.

Different types behave differently in Python. When you sum two strings, for example, you'll get different behavior than when you sum two integers or two booleans.

Time for you to test this out.

#### Instructions 1/2

50 XP

- Add `savings` and `new_savings` and assign it to `total_savings`.  
Use `type()` to print the resulting type of `total_savings`.

- Calculate the sum of `intro` and `intro` and assign the result to `doubleintro`.  
Print out `doubleintro`. Did you expect this?

script.py

```
1 savings = 100
2 new_savings = 40
3
4 # Calculate total_savings using savings and new_savings
5 total_savings = savings + new_savings
6 print(total_savings)
7
8 # Print the type of total_savings
9 print(type(total_savings))
```



Run Code

Submit Answer

IPython Shell

Slides

In [1]: