What is Python?

INTRODUCTION TO PYTHON FOR DEVELOPERS

George Boorman
Curriculum Manager, DataCamp

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

Spotify

NETFLIX

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

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/@theoneofussocialclub

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

datacamp

The DataCamp interface

IPython Shell

Python Script

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

Performing calculations

- 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!

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)

script.py

Light Mode

```
# Addition
print(4 + 5)

# Subtraction
print
      print                    local
# Multiplication


# Division

```

Run Code

Submit Answer

IPython 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

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

```python
# 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