



CHALLENGES

MEDIA

SQUADS

INACTIVITY

CLUSTER

STATISTICS

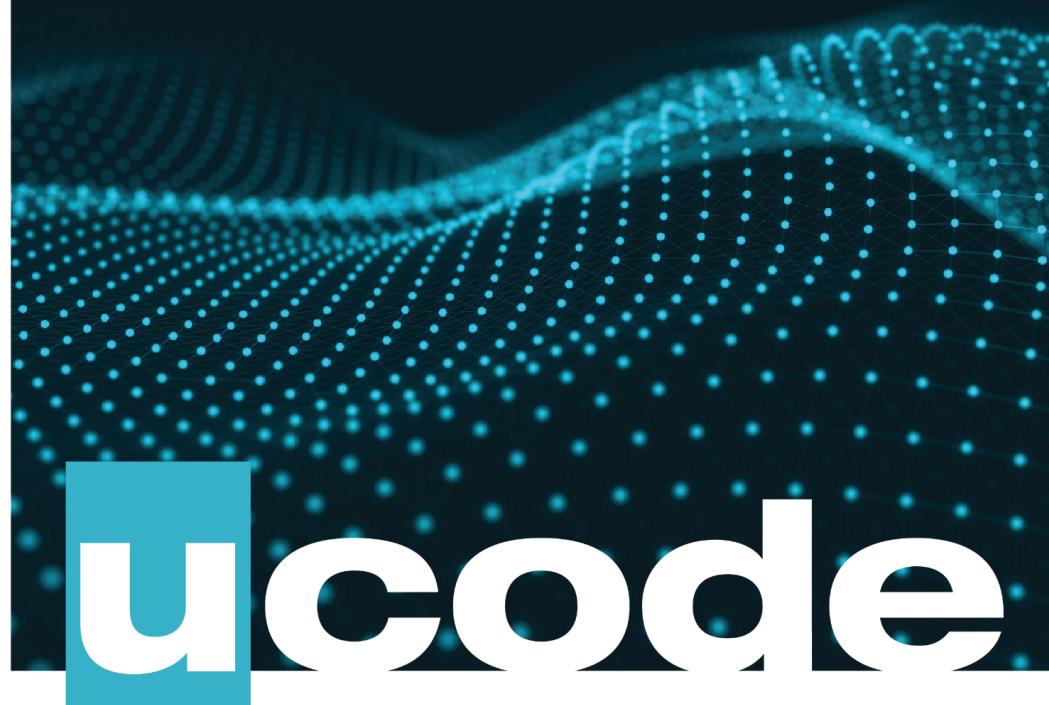


# Sprint 00

Marathon Python



April 9, 2021



## Contents

Engage .....	2
Investigate .....	3
Act Basic: Task 00 > Hello! . . . . .	5
Act Basic: Task 01 > Data types . . . . .	6
Act Basic: Task 02 > Variables . . . . .	7
Act Basic: Task 03 > Concatenate . . . . .	9
Act Basic: Task 04 > Bridge of Death . . . . .	10
Act Basic: Task 05 > Math operations . . . . .	12
Act Basic: Task 06 > Casting . . . . .	14
Act Basic: Task 07 > Condition . . . . .	16
Act Advanced: Task 08 > What is the address? . . . . .	18
Act Advanced: Task 09 > Get rid of it . . . . .	20
Act Advanced: Task 10 > Do op . . . . .	21
Act Advanced: Task 11 > Bot . . . . .	22
Share . . . . .	24

**ucode**

## Engage

### DESCRIPTION

Welcome to the Python Sprint 00!

Welcome to the `Python` programming language!

Python is one of those rare languages that can claim to be both simple and powerful. You will be surprised how easy it is to concentrate on the solution to the problem rather than the syntax and structure of the language you are programming in. Python can be used for things like:

- backend (or server-side) web and mobile app development
- desktop app and software development
- processing big data and performing mathematical computations
- writing system scripts (creating instructions that tell a computer system to "do" something)

But don't let Python's broad range scare you. Python is an easy to learn, in-demand programming language that can exponentially increase your chances of getting hired and increasing your income in a matter of months.

But first, you need to start with the basics. Therefore, in this challenge, you will get acquainted with variables, numbers, strings, operators, and other topics.

Let's start!

**BIG IDEA**

Introduction to Python.

**ESSENTIAL QUESTION**

How to start programming with Python?

**CHALLENGE**

Learn the basics of Python.

Sprint 00 | Marathon Python > 2



## Investigate



**GUIDING QUESTIONS**

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find answers, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- What are Python scripts?
- How to execute a `Python` script?
- What does it mean when people say that Python is an `interpreted programming language`?
- What areas of programming is Python best suited for?

## GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have a limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- What is it? Why is Python so popular?
- Watch The Story of Python.
- Prepare your environment for development. Make sure that you have Python with a 3.8 version, or higher. If not consider downloading it from the official Python website.
- Find some resources with articles about Python to gain the latest info about it (e.g. Medium).
- Learn about different Python interpreters.
- Open the terminal and create a Python script called main.py containing a line import this. Then, run the command python3 main.py. Here is how to do this in the console:

```
>touch main.py  
>echo 'import this' > main.py  
>cat main.py  
import this  
>python3 main.py  
The Zen of Python, by Tim Peters  
  
...  
>
```

- Enjoy the zen.
- Then let's try the same thing in the Python Interpreter. Stay in the terminal and do the following:

```
>python3  
>>> import this  
The Zen of Python, by Tim Peters  
  
...  
>>>
```

Sprint 00 | Marathon Python > 3



- You got exactly the same result as when running the script.
- Attentively watch and investigate learning videos available on the challenge page. Try to repeat all actions.
- Clone your git repository issued on the challenge page in the LMS.
- Proceed with tasks.

## ANALYSIS

Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Be attentive to all statements of the story.
- All tasks are divided into Act Basic and Act Advanced. You need to complete all basic tasks to validate the Sprint. But to achieve maximum points, consider accomplishing advanced tasks also.
- Analyze all information you have collected during the preparation stages. Try to define the order of your actions.
- Perform only those tasks that are given in this document.

\* Submit only those files that are described in the story. Only useful files allowed.

- Submit only those files that are described in the story. Only useful files allowed, garbage shall not pass!
- Run the scripts using `python3`.
- Make sure that you have `Python` with a `3.8` version, or higher.
- Use the standard library available after installing `Python`. You may use additional packages/libraries that were not previously installed only if they are specified in the task.
- Complete tasks according to the rules specified in the `PEP8` conventions.
- The solution will be checked and graded by students like you. `Peer-to-Peer learning`.
- Also, the challenge will pass automatic evaluation which is called `Oracle`.
- If you have any questions or don't understand something, ask other students or just Google it.



Sprint 00 | Marathon Python &gt; 4

## Act Basic: Task 00



### NAME

Hello!

### DIRECTORY

`t00_hello/`

### SUBMIT

`hello_world.py`

### LEGEND

- Who's that then?
- I dunno, must be a king.
- Why?
- He hasn't got sh\*\* all over him.

-- Monty Python and the Holy Grail

### BEFORE YOU BEGIN

In order to complete this task, you must be able to answer the following questions:

- How to print text using Python?
- How to run Python scripts?
- What is the difference between single and double quotes in Python?

### DESCRIPTION

Create a script that prints the message below to the console:

'Hello world! It is I, Arthur, son of Uther Pendragon, from the castle of Camelot.'

See how your script must work in the section **CONSOLE VIEW**.

**CONSOLE VIEW**

```
>python3 hello_world.py
Hello world! It is I, Arthur, son of Uther Pendragon, from the castle of Camelot.
>
```

**SEE ALSO**

[Python print\(\) Function](#)  
[Single and Double Quotes | Python](#)

**ucode**

Sprint 00 | Marathon Python > 5

## Act Basic: Task 01



**NAME**  
Data types

**DIRECTORY**  
t01\_data\_types/

**SUBMIT**  
data\_types.py

**BEFORE YOU BEGIN**

In order to complete this task, you must be able to answer the following questions:

- What are data types?
- What are data types for?
- What data types are there in Python?
- What is the difference between values: 0, '0', and 0.0?

**DESCRIPTION**

Create a script that prints values in different data types:

- integer with a value 50
- float with a value -42.00001
- string with a value 'Monty Python'
- boolean with a value True

Your output must be like in the **CONSOLE VIEW**.

**CONSOLE VIEW**

```
>python3 data_types.py
50
-42.00001
Monty Python
True
>
```

**SEE ALSO**

The Python Standard Library Data Types  
Python Data Types



Sprint 00 | Marathon Python > 6

## Act Basic: Task 02



**NAME**  
Variables

**DIRECTORY**  
t02\_variables/

**SUBMIT**  
var.py

**LEGEND**

- All right, but apart from the sanitation, the medicine, education, wine, public order, irrigation, roads, the fresh-water system, and public health, what have the Romans ever done for us?
- Brought peace?
- Oh, peace? SHUT UP!

-- Monty Python's Life of Brian

**BEFORE YOU BEGIN**

In order to complete this task, you must be able to answer the following questions:

- What is a variable in programming?
- What is the advantage of using variables to store values?
- How to define a variable?
- How to determine the data type of a variable?
- How to import a variable from a script?
- What characters can you use in a variable name?
- Which of these cannot be the first character of a variable name: uppercase letter, digit, lowercase letter, underscore?

There are good and bad ways to name a variable, but the main goal is to make your code readable and understandable. Find out what are the best practices of variable naming, and what are the conventions for variables in Python, according to PEP 8.

**DESCRIPTION**

Create a script that:

- defines the following variables:
  - an integer called `my_int` with the value `1`
  - a float called `my_float` with the value `2.123`
  - a string called `my_str` with the value `'Tis but a scratch.'`
  - a boolean called `my_bool` with the value `False`



- prints the types of each of these variables using the function `type()` in the correct order

Your output must look like the **CONSOLE VIEW**. You can also find the output in the **resources** for the challenge.

The **PYTHON INTERPRETER** section demonstrates a way to check if your script works correctly on the example of the float. As you can see, we have imported the variables from the script. Once imported, the script was executed and generated the same output as the **CONSOLE VIEW**, but that's not what we're interested in. The next command: `>>> type(my_float)` outputs the type of the variable `my_float`. If your script has defined the variable correctly, the output must be `<class 'float'>`. You can test other variables using the same approach.

#### CONSOLE VIEW

```
>python3 var.py
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
>
```

#### PYTHON INTERPRETER

```
>python3
>>> from var import my_int, my_float, my_str, my_bool
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
>>> type(my_float)
<class 'float'>
>>> my_float
2.123
>>>
```

#### SEE ALSO

[Python Variables](#)  
[Python Data Types](#)



## Act Basic: Task 03

### NAME

Concatenate

### DIRECTORY

t03\_concatenate/

### SUBMIT

concatenate.py

### BEFORE YOU BEGIN

- How to concatenate strings using an operator? What operator is it?
- What other methods of concatenation are available in Python?

### DESCRIPTION

Create a script that:

- contains a variable `first_name` with a value 'Monty'
- contains a variable `last_name` with a value 'Python'
- contains a variable `full_name` that must store the result of concatenation of `first_name` and `last_name` connected with a space ' '
- prints the variable `full_name`

Your output must be like in **CONSOLE VIEW**.

### CONSOLE VIEW

```
>python3 concatenate.py  
Monty Python  
>
```

### SEE ALSO

[Python String Concatenation](#)

Sprint 00 | Marathon Python > 9

**ucode**

## Act Basic: Task 04

**NAME**  
Bridge of Death

**DIRECTORY**  
`t04_bridge_of_death/`

**SUBMIT**  
`input.py`

**BEFORE YOU BEGIN**

In order to complete this task, you must be able to answer the following questions:

- How to get input from the user?
- How to format a string with f-strings?

Let's try to apply the new knowledge.  
Start the Python Interpreter and repeat the following steps (follow line by line, include your name).

**PYTHON INTERPRETER**

```
>python3
>>> name = input('Enter your name: ')
Enter your name: ucoder
>>> f'Hello {name}'
'Hello ucoder'
>>> quit()
>
```

**DESCRIPTION**

Create a script that:

- reads several variables from the `stdin` using the `input()` function
  - name
  - quest
  - color
- prints the entered information

In the **CONSOLE VIEW** example, the user-entered values are:

- Sir Lancelot of Camelot on the question 'What is your name?'
- to seek the Holy Grail on the question 'What is your quest?'
- blue on the question 'What is your favorite color?'

The full output can be found in the challenge **resources** in the LMS.

Don't hardcode the output, values must only be accepted via `input()`.

Sprint 00 | Marathon Python &gt; 10



**CONSOLE VIEW**

```
>python3 input.py
BRIDGEKEEPER: Stop.
Who would cross the Bridge of Death must answer me these questions three.
BRIDGEKEEPER: What is your name?
Sir Lancelot of Camelot
BRIDGEKEEPER: What is your quest?
to seek the Holy Grail
BRIDGEKEEPER: What is your favorite color?
```

```
blue
-----
Traveler info:
Your name is Sir Lancelot of Camelot
Your quest is to seek the Holy Grail
Your favorite color is blue
-----
BRIDGEKEEPER: Right. Off you go.
>
```

#### SEE ALSO

[Python input\(\) Function](#)  
[How To Use f-strings to Create Strings in Python 3](#)



Sprint 00 | Marathon Python > 11

## Act Basic: Task 05



#### NAME

Math operations

#### DIRECTORY

t05\_math\_operations/

#### SUBMIT

math.py

#### BEFORE YOU BEGIN

In order to complete this task, you must be able to answer the following questions:

- What arithmetic operators exist in Python?
- What is the difference between these two division operators: `/` and `//`?
- How can this operator `%` help determine whether a number is odd or even?

**DESCRIPTION**

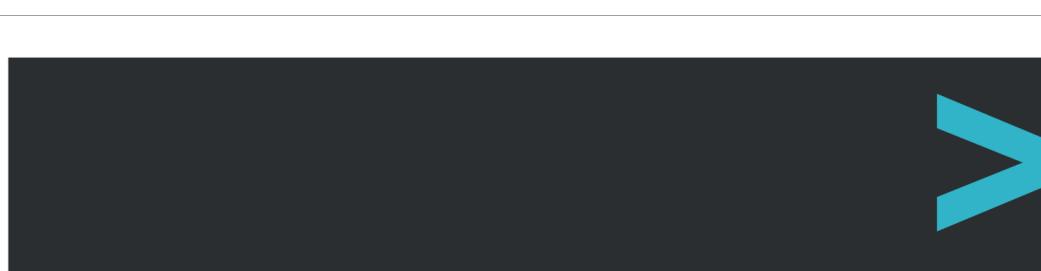
Create a script that (the first two requirements were done for you):

- asks to enter two numbers in sequence
- stores values in variables `a` and `b` respectively, and casts them to type `integer` (you can find an example of casting below in the **SYNOPSIS**)
- performs primitive math operations on them and stores the result of each operation in variable `c`
  - addition (+)
  - subtraction (-)
  - multiplication (\*)
  - division (/)
  - modulus (%)
  - exponent (\*\*)
  - floor division (//)
- prints the result for each operation

You must write your implementation with operations and their output to get the result as in the **CONSOLE VIEW**.

Use `f-strings` to format strings.

**Sprint 00 | Marathon Python > 12**



**SYNOPSIS**

```
a = input('Enter the first number: ')
b = input('Enter the second number: ')
a = int(a)
b = int(b)

# write your code here
```

**CONSOLE VIEW**

```
>python3 math.py
Enter the first number: 7
Enter the second number: 4
7 + 4 = 11
7 - 4 = 3
7 * 4 = 28
7 / 4 = 1.75
7 % 4 = 3
7 ** 4 = 2401
7 // 4 = 1
>python3 math.py
Enter the first number: 33
Enter the second number: 4
33 + 4 = 37
33 - 4 = 29
33 * 4 = 132
33 / 4 = 8.25
```

```
33 / 4 = 8.25
33 % 4 = 1
33 ** 4 = 1185921
33 // 4 = 8
>
```

#### SEE ALSO

[Python Arithmetic Operators Example](#)  
[f-Strings](#)  
[Python Casting](#)



Sprint 00 | Marathon Python > 13



















