

Python (Essentials)

Brief Introduction: Python has quickly become one of the most popular programming languages because of its human-friendly syntax and efficiency, allowing programmers to write fewer lines of code than in other languages. It has many applications and is used in databases and web development, to name a few.

Important Note! If you want to practice coding Python, you can install Anaconda for Python 3.7 [here](#). When you open Anaconda-Navigator, launch Jupyter Notebook and you can practice coding. You can also install Python [here](#) and access it through your command prompt by calling python (for Windows) or python3 (for Mac), but if you choose this way, make sure you download the Python 3 version because Python 3 and 2 are quite different.

This lecture will cover some of the most basic topics that are essential to understand before moving onto the “hard stuff” (but really, everything in computer science is hard if you squint at it long enough). Some of this might be familiar to you, but look over it anyway because we’ve worked hard on it and would appreciate it if you appreciated our hard work.

Math Operations

Computer science, like with most STEM-related subjects, include mathematical calculations. Below is a list of the basic math operations you will likely come across quite often.

```
Input: 1+1          # addition
Output: 2
Input: 3-2          # subtraction
Output: 1
Input: 1 * 2        # multiplication
Output: 2
Input: 1 / 2.0      # division
Output: 0.5
Input: 2 ** 3       # ** denotes “to the power of” → exponents
Output: 8
Input: 5 % 2        # % = modulus → solves for the remainder
Output: 1
Input: 5 // 2       # // rounds down to the nearest whole number
Output: 2
```

Variables

Variables are very important in programming because they allow you to store information to be used later on in your code (the computer will allocate memory to store the variable). They also act as identifiers so that your code is easier for you and others to understand. In Python, you don’t have to assign a type to the variable; unlike in Java or C, Python variables can switch between types and are not immutable (meaning they can change types).

```
Input: var = 5 # sets the variable equal to a certain value
          # variables can not start with symbols (i.e. !$)
```

```
var      # calls the variable
Output: 5
```

Note: when you are naming a variable, you want to write it in lowercase (so num, not Num). If the variable is made up of multiple words, you can either use camel case, which is where you capitalize the first letter of every word after the first (e.g. softDev), or snake case, which is separating words by underscores (e.g. soft_dev).

Data Types

Type is the classification of what type of value a variable is and, depending on the type, what operations that variable has access to. There are two different data types: primitive (int, float, double, boolean, char, etc.) and reference types (strings, lists, etc.), although it's not so important in Python as it would be in Java since, as stated above, variables in Python are mutable.

Here's a list of the five most common types you'll encounter:

```
int (1,5, 67)
float (3.14, 9.283)
boolean (true or false)
str ("banana", "jamba juice")
list (["Iron Man", 85, true])
```

Note: Because data type isn't as important in Python, lists do not have to contain one type only (like it does in Java). So, you can have a string, integer, and boolean in one list; it's magical.

Commenting

Commenting makes your program more legible and comprehensible to other programmers (including yourself!). You'll want to comment your code often so that you can refer back to the comments whenever you need without having to read through hundreds of lines of code.

The *inline comment* (#) in python is a single line of code you can use to leave notes for yourself. (*Note: # only works for one-line comments. If you want to make multi-line comments, use triple quotes as shown below.*)

```
# this is an inline comment
```

The *multi-line comment* is generally used when writing program and function descriptions and is not usually used to comment lines of code. You'll want to use the inline comment for that.

```
"""
Description: This program does a,b, and c and its end goal is x, y, and z.
Author: Emma
Date: Summer 2020
This is a multi-line comment.
"""
```

Strings

The String type is one of the most versatile types because it is a reference type (referring to an *object*). This means that Strings have access to methods like `.isalpha()` and `.capitalize()`, which checks if the string is made of alphabetic characters and capitalizes the first letter of the string respectively.

```
Input: "String" # doesn't matter whether you use ' or " around your phrase
Output: 'String'
```

Strings can be saved as variables, and to print the string in the command prompt you use the `print()` function (it's probably the easiest thing you will ever encounter anywhere).

```
Input: x = "hello"
      x
Output: 'hello'
```

```
Input: print(x) # if you want the phrase without the quotes, use the print
function
Output: hello
```

Print formatting can be done in two ways by either using the `.format()` function or using `"%."` Both work, though `.format()` came with a more recent update.

```
Input: # the format function allows for variable substitutions
      name = 'Sherry'
      age = 13
      print("My name is { } and I am { } years old.".format(name, age))
Output: My name is Sherry and I am 13 years old.
```

```
Input: name = 'Sherry'
      age = 13
      print("My name is %s and I am %d years old." % (name, age))
Output: My name is Sherry and I am 13 years old.
```

What's important to note about the `"%"` way is that it requires knowledge of types. The `"name"` variable is a string, so you have to add the `s` in `"%s"` to indicate that. Since `"age"` is an integer, you use `"%d"`. If we were talking about money, which would be a float number, you'd use `"%f"`.

String Indexing is a handy tool that allows you to manipulate a string; you can take out a single character or cut the string to have the part that you need. It's dead useful and you'll probably use it often.

```
Input: string = "abcdefg"
      string[0] # the first letter of the string corresponds to the index 0
Output: 'a'
```

```
Input: string[0:] # because there is no stated upper limit, the output will
                  # start at the first letter and contain the entire string
Output: 'abcdefg'
```

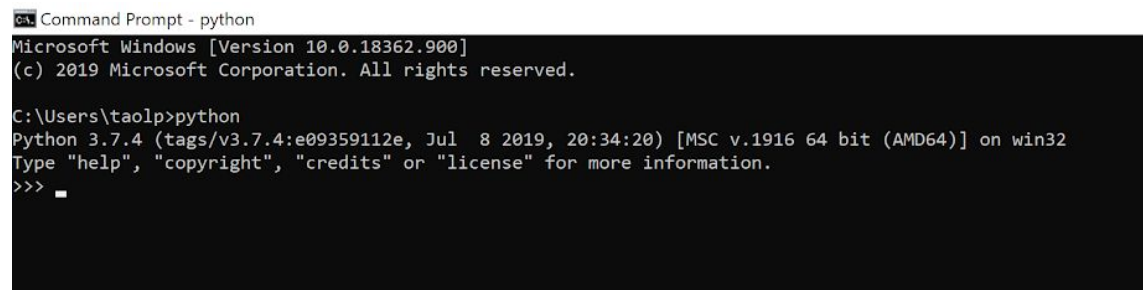
```
Input: string[:3]      # in this case, the third index corresponds to the letter
                        # 'd,' but in this function, the upper limit is not
                        # included in the output

Output: 'abc'
```

“Hello, World!”

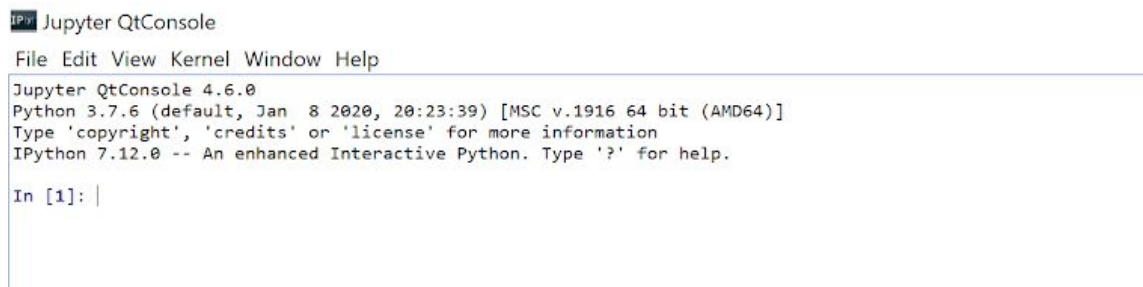
The “Hello, World!” program is the most famous program you will ever know because it is usually every programmer’s first line of code, and it’s literally one line! Below is a step-by-step guide to help you take your first step into the world of programming (or for a hint of self-satisfaction if that’s what you’re looking for).

Step 1: Open up your command prompt/Anaconda workspace.



```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\taolp>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

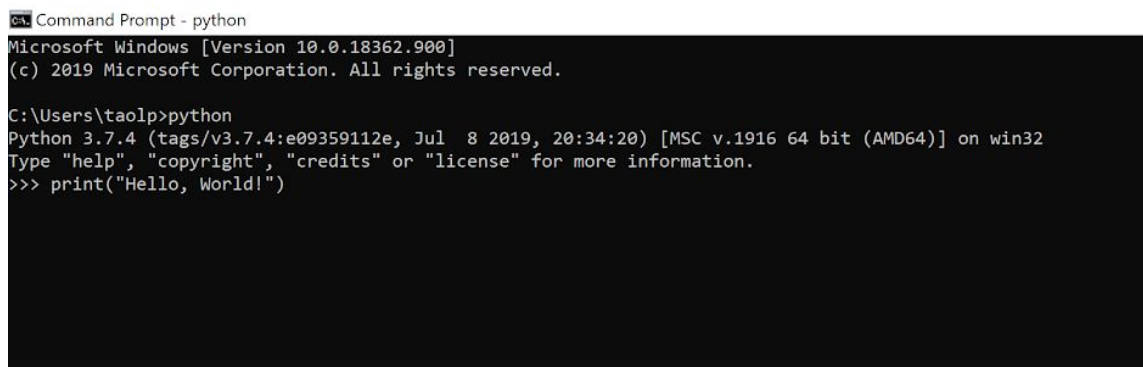


```
Jupyter QtConsole
File Edit View Kernel Window Help

Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

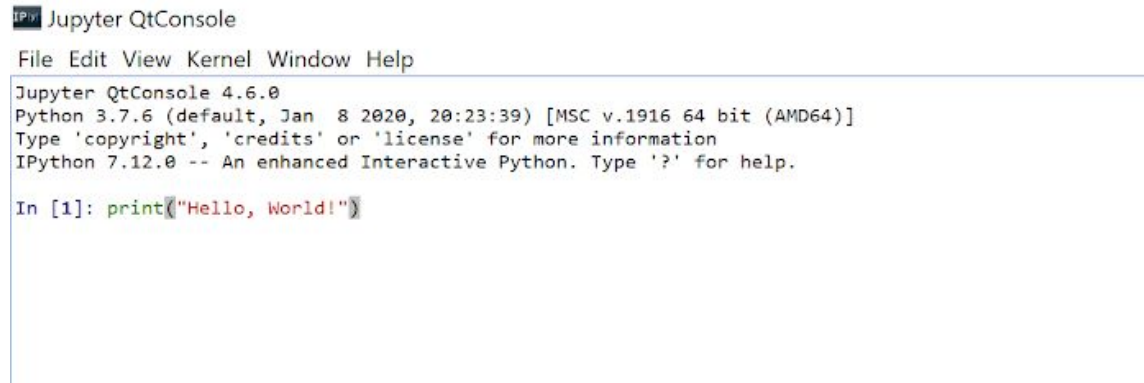
In [1]: |
```

Step 2: In your console, type in `print("Hello, World!")`. No semicolon is necessary because Python is neat that way.



```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\taolp>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
```

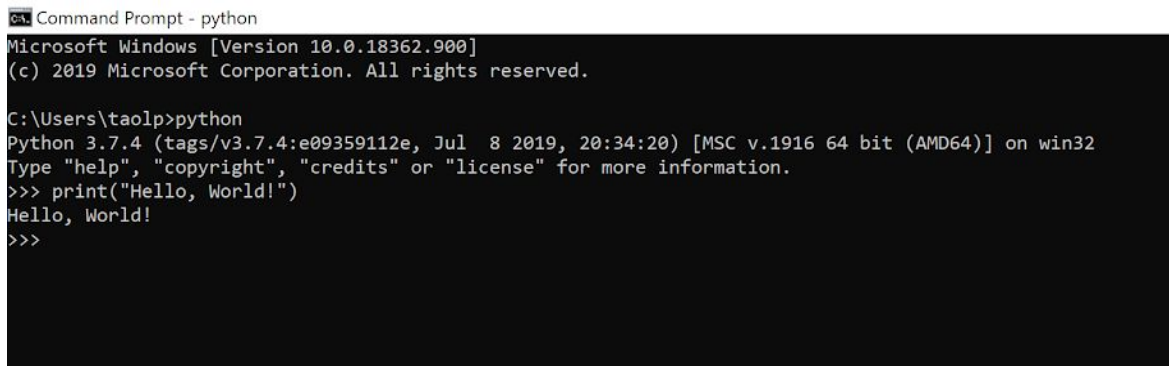
A screenshot of a Jupyter QtConsole window. The title bar says "Jupyter QtConsole". The menu bar includes "File", "Edit", "View", "Kernel", "Window", and "Help". The console text shows the Jupyter version (4.6.0), Python version (3.7.6), and IPython version (7.12.0). It prompts the user to type 'copyright', 'credits', or 'license' for more information. The first input is `In [1]: print("Hello, World!")`, which has been executed, but the output is not visible in this screenshot.

```
Jupyter QtConsole
File Edit View Kernel Window Help

Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

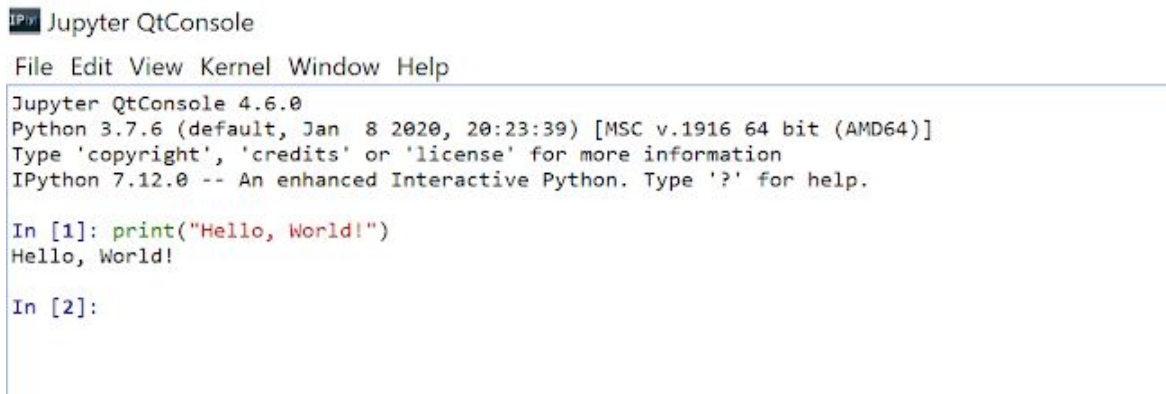
In [1]: print("Hello, World!")
```

Step 3: Watch the magic unfold and do your best to grasp the sheer greatness that you have just witnessed, but it's ok if you don't quite manage; we certainly didn't. Congratulations! You've written your very first line of code! (Or, you have revisited fond memories of a time when this was your first time coding.)

A screenshot of a Windows Command Prompt window titled "Command Prompt - python". It shows the execution of the Python command in a standard Windows environment. The output of the `print("Hello, World!")` command is visible.

```
Command Prompt - python
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\taolp>python
Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("Hello, World!")
Hello, World!
>>>
```

A screenshot of a Jupyter QtConsole window, similar to the first one. It shows the same initial text, but now the output of the first command is visible: `Hello, World!`. The second input prompt `In [2]:` is also visible.

```
Jupyter QtConsole
File Edit View Kernel Window Help

Jupyter QtConsole 4.6.0
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 7.12.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("Hello, World!")
Hello, World!

In [2]:
```

This concludes the very first lesson of Code Club! We hope you learned lots, and the next lesson will hopefully be out sometime next week or the week after that.
- Emma and Nicole <3