**Intro to Python.**

|  |
| --- |
| **Overview**  Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. |

**Python is used in a lot of industries:**

* Insurance - creating business insights with machine learning.
* Retail Banking - flexible data transformation and manipulation.
* Aerospace - Automating data analysis.
* Finance - Data mining to identify cross-sell opportunities.
* Business Services - API access to financial information.
* Hardware - Automating network administration.
* Healthcare - Predicting disease prognosis.
* Consulting Services - Bespoke web development.
* Software - Adding extensibility to legacy applications.
* Gaming - E.g Battlefield 2, Civilization IV, The Sims 4

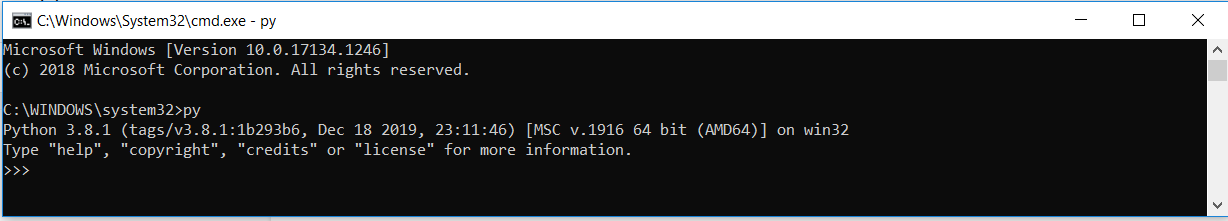
**What can Python do?**

* Python can be used on a server side to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping(a technique used to build something quickly).

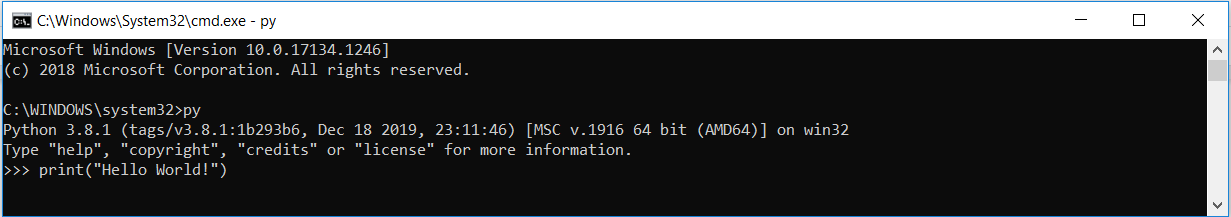
**The Python Command Line**

To test a short amount of code in python sometimes it is quickest and easiest not to write code in a file. This is made possible because Python can be run on the command line itself.

Type the following ***py*** on the command line:

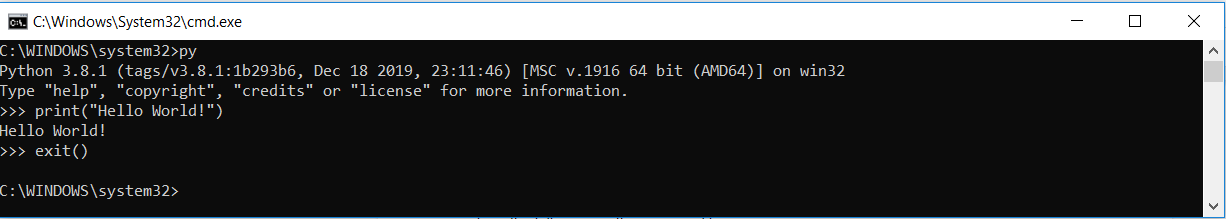


From there you can write any python statements (code), including hello world example from earlier.



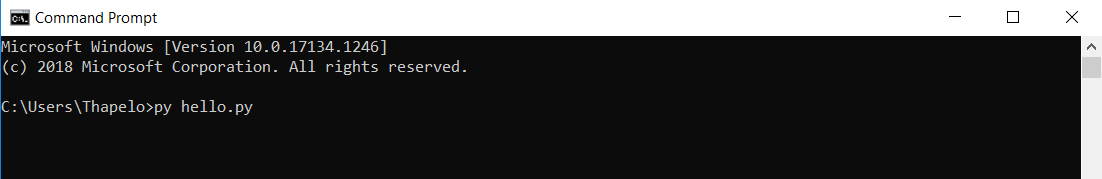
Which will write “Hello World!” in the command line.

Whenever you are done in the python command line, you can simply type the following ***quit()*** *or* ***exit()*** to quit the python command line interface.



**Execute Python Syntax**

Python syntax can be executed by writing directly in the Command Line, or by creating a python file with the .py file extension, and running it in the command line.



**Python Indentation**

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages uses punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are optional. It has fewer syntactic exceptions and special cases than C or Pascal

Indentation refers to the spaces at the beginning of the code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code.

A block of statements can be executed from top to bottom unless a jump/goto statement is used.

Python will give you an error if you skip the indentation. The number of spaces is up to you as a programmer, but it has to be at least one. You have to use the same number of spaces in the same block of code, otherwise Python will give you a syntax error when run in the interpreter. The python guidelines recommend the use of 4 spaces per indentation level.

**Statement and control flow**

Python’s statements include (among others)

* The ***assignment*** statement (token ‘=’, the equal sign), eg message=’Hello World!’. In Python, variables are created when you assign a value to it
* The **if** statement, which conditionally executes a block of code, along with ***else*** and ***else-if***.
* The ***for*** statement, which iterates over an iterable object.
* The ***while*** statement, which executes a block of code as long as its condition is true.
* The ***try*** statement, which allows exceptions to be raised.
* The ***def*** statement, which defines a function or a method.

We will cover statements later on in this module.

**Python Variables**

In Python, variables are containers used to store a value. Python has no command for declaring a variable, as opposed to other languages like Java, C#, C++.

An example of a variable in python looks like this: *x = 5, y = ‘Test’*

The value of a variable can be of any data type (e.g String, Numeric etc). Variables are created when first assigned before being referenced. The data type (String, integer etc) of a variable is determined by Python.

A variable can have a short name (like x and y) or a more descriptive name (age, car\_name, total\_volume). Rules for Python variables are:

* A variable must start with a letter or the underscore character
* A variable name cannot start with a number.
* A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and \_)
* Variable names are case-sensitive (age, Age and AGE are different variables)

There is a python coding convention/guidelines that are recommended by Python and is available by clicking [this](https://www.python.org/dev/peps/pep-0008/) link.

**Good variable names**

* Choose a meaningful name instead of short name. Roll\_no is better than rn.
* Maintain the length of a variable name. Roll\_no\_of\_a\_student is too long.
* Be consistent with your casing; roll\_no or RollNo. Read the python recommended [style guide](https://www.python.org/dev/peps/pep-0008/) for more information.
* Begin a variable name with an underscore(\_) character for a special case.

Also remember that variable names are case-sensitive.

An example of variables are listed below:

*x = 5*

*y = "John"*

*print(x)*

*print(y)*

After running the code in the command line interpreter, the variable x and y will be assigned a value of *5* and *john* respectively.

**Assign Value to Multiple Variables**

Python allows you to assign values to multiple variables in one line.

**Example**:

x, y, z = "Orange", "Banana", "Cherry"

print(x)

print(y)

print(z)

Also you can assign the same value to multiple variables on one line.

**Example:**

**x = y = z = "Orange"**

**print(x)**

**print(y)**

**print(z**

**Comments**

Python has commenting capability for the purpose of in-code documentation. Comments start with #, and Python will render the rest of the line as a comment. Block comments generally apply to some (or all) code that follows them, and are indented to the same level as that code.

**Python block comment Examples**.



**Multiline comment syntax**

Here is a syntax on how you can do multiple comments in python.



**Output Variables**

The Python print statement is often used to output variables. To combine both text and a variable, Python uses the *+* character:

**Example:**

**Message = "awesome"**

**print("Python is " + message)**

**TASK 1**

**Write a hello world program in the interpreter.**

**Instructions:**

Open the python interpreter and write a piece of code which will output *Hello World!* To the command prompt.

Remember we use the print statement to write anything to the command prompt.

**Expected output:**

*Hello World!*

**Task 2**

**Write a python program which accepts accepts the radius of a circle from the user and compute the area.**

Remember the area of a circle is computed using this formula, **pi \* r**2

**Task 3**

**Using the same input, calculate the diameter and circumference of the circle and display your answer. Add some comments to your program. Demonstrate your understanding of single line comments and multi-line comments.**