Python Documentation

1. Introduction

Python is a high level general-purpose programming language. It uses a multi-paradigm approach, meaning it supports procedural, object-oriented, and some functional programming constructs.

It was created by Guido van Rossum as a successor to another language (called ABC) between 1985 and 1990, and is currently used on a large array of domains like web development, desktop applications, data science, DevOps, and automation/productivity.

Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python's license is administered by the Python Software Foundation.

See also

- Python on Wikipedia

2. What you should already know

Python Install

Many PCs and Macs will have python already installed.

To check if you have python installed on a Windows PC, search in the start bar for Python or run the following on the Command Line (cmd.exe):

```
C:\Users\Your Name>python --version
```

To check if you have python installed on a Linux or Mac, then on linux open the command line or on Mac open the Terminal and type:

```
python --version
```

If you find that you do not have Python installed on your computer, then you can download it for free from the following website: https://www.python.org/

3. Python and C++

Python and C++ both are the most popular and general-purpose programming languages. They both support Object-Oriented Programming (OPP) yet they are a lot different from one another. In this article, we will discuss how Python is different from C++.

Difference Between C++ vs Python

The following table shows the key difference between Python and C++ programming languages.

Parameters	Python	C++
Code	Python has fewer lines of code	C++ tends to have long lines of code
Syntax	It uses short-hand syntax and has numerous short-hand structural iterators. It requires 'self' as a parameter to any class instance method. Some of the short-hand syntaxes are confusing (e.g.	It has a stiff learning curve as it has lots of predefined syntaxes and structures. C++ uses implicitly 'this' to refer to class

	`rkwargs`) but rarely is anything in Python completely esoteric.	instances. Some syntax in C++ is extremely esoteric.
Compilation	Python is interpreted	C++ is precompiled
Speed	It is slower since it uses an interpreter and also determines the data type at run time	It is faster once compiled as compared to python
Efficiency	Specialized formatting not common in other languages, script-like language, OOP features, code reuse through libraries	C-like syntax, powerful OOP features and operator overloading, best compile-time optimizer
Nature	It is dynamically typed.	It is statically typed.
Functions	Python Functions do not have restrictions on the type of the argument and the type of its return value	In C++, the function can accept and return the type of value which is already defined.
Scope of Variable	Variables are accessible even outside the loop.	The scope of variables is limited within the loops.
Extension	Python programs are saved with the .py extension.	C++ programs are saved with the .cpp extension
Popularity	It has huge community support. When it comes to popularity, beginner and novice programmers tend to turn to Python.	It also has dedicated followings online. But only the people who have some experience in the field show a lot of interest in C++.
Garbage Collection	It supports garbage collection.	It doesn't support garbage collection, but it can be implemented.
Rapid Prototyping	Rapid Prototyping is possible, with easy project setup, live interpreter.	Rapid Prototyping is possible, but project setup can be complicated, live interpreter through IRC bot.
Application Domain	Web development, data analysis, scientific computations, etc.	Game development, embedded systems, etc.
Variable Declaration	Variables are declared by simply writing their name with their datatype.	While declaring a variable it is necessary to mention its datatype.
Installation on Windows	Difficult to install on Windows.	Easy to install on Windows.

4. Hello world

To get started with writing Python, open Jupyter Notebook and write your first "Hello world" Python code:

```
print("Hello, World!")
```

5. Variables

Creating Variables

Python has no command for declaring a variable.

```
x = 5
y = "John"
print(x)
print(y)
```

6. Data Types

Global Variables

Variables that are created outside of a function (as in all of the examples in the previous pages) are known as global variables.

Global variables can be used by everyone, both inside of functions and outside.

```
x = "awesome"
def myfunc():
print("Python is " + x)
myfunc()
```

7. Dictionaries

Dictionaries are used to store data values in key:value pairs.

A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

As of Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

Dictionaries are written with curly brackets, and have keys and values:

```
thisdict = {
"brand": "Ford",
"model": "Mustang",
"year": 1964
}
print(thisdict)
```

8. Built-in Math Functions

Python has a set of built-in math functions, including an extensive math module, that allows you to perform mathematical tasks on numbers.

The min() and max() functions can be used to find the lowest or highest value in an iterable:

```
x = min(5, 10, 25)
y = max(5, 10, 25)
print(x)
print(y)
```

The abs() function returns the absolute (positive) value of the specified number:

```
x = abs(-7.25)
print(x)
```

The pow(x, y) function returns the value of x to the power of y (x^y).

```
x = pow(4, 3)
print(x)
```

Return the value of 4 to the power of 3 (same as 4 * 4 * 4).

9. Reference

- All the documentation in this page is taken from various sources including official Python documentation and tutorials.