Introduction:

Why Python? Python Concepts, Dynamic vs. Static Types, Procedural vs. Object-Oriented Programming, Comparing Programming Languages: C,C++,JAVA, C#, Python, Python Interpreter

Up to now C programming language was known as Mother Language to learn any programming language.

Python is a simple, general purpose, high level, and object-oriented programming language.

General Purpose means: We can use python for multiple purposes. For data science applications, machine learning, web application, desktop application and many more can be done with python. Therefore python is a general purpose programming language.

High Level Language: Machine level language and Assembly level language, are low level language, which is not easily understandable language. These are the languages are able to communicate directly with the computer. These are not human understandable languages.

There are some languages which are understandable by human OR programmer. These are human understandable languages. Exa., C, C++, JAVA, Python etc.

Python is an interpreted scripting language also. **Guido Van Rossum** is known as the founder (father) of Python programming in **1989** while working at **National Research Institute at Netherland.** But python language is available in market since Feb 20th 1991 for general purpose usage by public.

This is highly recommended programming language to start learning programming language as a first language.

Consider the C Program to display “Hello World”

```
#include<stdio.h>
#include<conio.h>
void main()
{
        printf(“Hello World”);
}
```

Consider the Java Program to display “Hello World”

```
public class Demo
{
        public static void main(String args[])
        {
                System.out.println(“Hello World”);
        }
}
```

Now, If we consider the “Hello World” using python then:

```
print(“Hello World”)
```

Just 1 line is enough in contrast with JAVA and C language with around 8-10 line of code. Here, you don’t need to write class, main method, etc.

Consider the C program to add two numbers:

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a = 10;
        int b = 10;
        printf(a+b);
}
```

Consider the JAVA program to add two numbers:

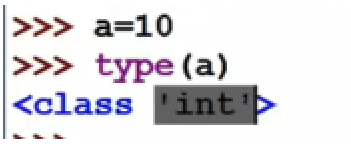
```
public class Demo
{
        public static void main(String args[])
        {
                int a=10;
                int b=20;
                System.out.println(a+b);
        }
}
```

Now, consider the program to add two numbers using python programming language.

```
a, b = 10, 20
print(a+b)
```

Biggest advantage of python is, we can write complex and big code in simpler way.

If you want to use any variable, then data type of that variable must be declared priory in C OR JAVA type of language. They are having certain fixed structure. That’s why C and JAVA languages are called as Statically Typed Programming language. Static means at the beginning we must need to define types. Whereas, in case of python no such need of specifying type is there.



```
>>> a=10
>>> type(a)
<class 'int'>
```

It automatically consider type based on the value given to variable. There is no need to add even semicolons here.

Consider one more case:

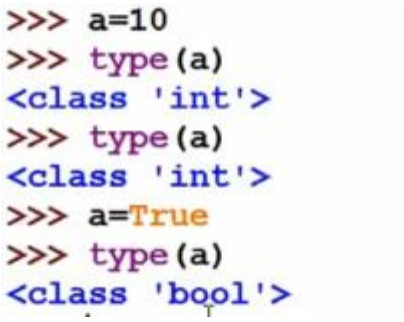
Consider the JAVA program:

```
public class Demo
{
        public static void main(String args[])
        {
                int a=10;
                a=True;
        }
}
```

This will show error like incompatible type.

Now, consider in case of python.

```
a=10
a=True
```



```
>>> a=10
>>> type(a)
<class 'int'>
>>> type(a)
<class 'int'>
>>> a=True
>>> type(a)
<class 'bool'>
```

We are not in need to specify type of variable and that’s why python is dynamically typed programming language. For this dynamic programming language, there are some other features also to make it most recommended language.

Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

Python 2 vs. Python 3

In most of the programming languages, whenever a new version releases, it supports the features and syntax of the existing version of the language, therefore, it is easier for the projects to switch in the newer version. However, in the case of Python, the two versions Python 2 and Python 3 are very much different from each other.

A list of differences between Python 2 and Python 3 are given below:

1. Python 2 uses **print** as a statement and used as print "something" to print some string on the console. On the other hand, Python 3 uses **print** as a function and used as print("something") to print something on the console.
2. Python 2 uses the function raw_input() to accept the user's input. It returns the string representing the value, which is typed by the user. To convert it into the integer, we need to use the int() function in Python. On the other hand, Python 3 uses input() function which automatically interpreted the type of input entered by the user. However, we can cast this value to any type by using primitive functions (int(), str(), etc.).
3. In Python 2, the implicit string type is ASCII, whereas, in Python 3, the implicit string type is Unicode.
4. Python 3 doesn't contain the xrange() function of Python 2. The xrange() is the variant of range() function which returns a xrange object that works similar to Java iterator. The range() returns a list for example the function range(0,3) contains 0, 1, 2.
5. There is also a small change made in Exception handling in Python 3. It defines a keyword **as** which is necessary to be used.

Why the name given like: Python

At the same time he began implementing Python, Guido van Rossum was also reading the published scripts from Monty Python’s Flying Circus (a BBC comedy series from the seventies, in the unlikely case you didn’t know). It occurred to him that he needed a name that was short, unique, and slightly mysterious, so he decided to call the language Python.

Early on there was popular TV show over BBC namely Monty Python’s Circus. They decided to keep such type of funny name to his language.

While developing this language, **he borrowed almost all the features from other programming language.**

1. Functional programming from C
2. OOP from C++
3. Scripting languages features from Perl and Shell script
4. Modular programming features from Modula-3

So, Python is Functional, Object oriented, scripting (python script) and modular language. Modular means dividing function into small small modules.

In python most of the syntax borrowed from C and ABC language.

Where we can use python?

1. We can develop Desktop Application (e.g., Calculator)
2. Web Application – In python we have Django framework to develop web application, there is also flask to develop web application
3. To develop Database related applications
4. For Networking applications
5. In developing Games
6. Data Analysis
7. Machine Learning applications
8. AI applications
9. For IoT applications
10. For Chatbot, etc……

Which companies are using python?

Top most companies in the world including: Google, Youtube, Dropbox, NASA and many more….

PROCEDURAL ORIENTED PROGRAMMING	**OBJECT ORIENTED PROGRAMMING**
In procedural programming, program is divided into small parts called **functions**.	In object oriented programming, program is divided into small parts called **objects**.
Procedural programming follows **top down approach**.	Object oriented programming follows **bottom up approach**.
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Adding new data and function is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way for hiding data so it is **less secure**.	Object oriented programming provides data hiding so it is **more secure**.
In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.

PROCEDURAL ORIENTED PROGRAMMING	**OBJECT ORIENTED PROGRAMMING**
In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
Procedural programming is based on **unreal world**.	Object oriented programming is based on **real world**.
Examples: C, FORTRAN, Pascal, Basic etc.	Examples: C++, Java, Python, C# etc.

Python Features

Python provides lots of features that are listed below.

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

5) Free and Open Source

Python language is freely available at offical web address.The source-code is also available. Therefore it is open source.

6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support

Graphical user interfaces can be developed using Python.

10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.

Python Vs Other languages:

Java

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. For example, a Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program. Because of the run-time typing, Python's run time must work harder than Java's. For example, when evaluating the expression a+b, it must first inspect the objects a and b to find out their type, which is not known at compile time. It then invokes the appropriate addition operation, which may be an overloaded user-defined method. Java, on the other hand, can perform an efficient integer or floating point addition, but requires variable declarations for a and b, and does not allow overloading of the + operator for instances of user-defined classes.

For these reasons, Python is much better suited as a "glue" language, while Java is better characterized as a low-level implementation language. In fact, the two together make an excellent combination. Components can be developed in Java and combined to form applications in Python; Python can also be used to prototype components until their design can be "hardened" in a Java implementation. To support this type of development, a Python implementation written in Java is under development, which allows calling Python code from Java and vice versa. In this implementation, Python source code is translated to Java bytecode (with help from a run-time library to support Python's dynamic semantics).

C++

Almost everything said for Java also applies for C++, just more so: where Python code is typically 3-5 times shorter than equivalent Java code, it is often 5-10 times shorter than equivalent C++ code! Anecdotal evidence suggests that one Python programmer can finish in two months what two C++ programmers can't complete in a year. Python shines as a glue language, used to combine components written in C++.

C#

Below is the list of points describing the difference between Python vs C# Performance

1) C# is statically written whereas Python is a dynamically written language. C#, being a static language usually contains a build/compile step.

2) You would possibly be able to write a program in Python with less variety of lines then a corresponding program in C#. Python is extremely compatible with inter-language programs.

3) Python is the winner in easy learning, cross-platform development, the convenience of open supply libraries

4) C# is a winner in development method, tools, performance, language evolution speed, and its customary libraries.

5) Python is healthier in readability, C# has additional consistent syntax.

6) Python is a more dynamic language than C#.