**PYTHON BASICS**

Python was created by Guido Van Rossum in 1991.

It is a general purpose programming language.

Interpreted language.

In Python, code is executed line by line, whenever an error encoundered the execution stops there.

C, C++, Java are compiled

Dynamically typed language.

a = 10

(it dynamically checks the data type of the variable by the value assigned to it)

We doesn’t need to specify the data type of the variable explicitly.

Here a is an integer variable

Object Oriented Programming language

Fastest growing language.

Python syntax are easy compared to other languages

a = 10

b = 20

print(a+b)

Python syntax is more compressed. The number of lines is less is compared to other programming languages.

In C :

#include<stdio.h>

void main()

{

System.out.println(“Hello World!”);

}

In Python:

print(“Hello World!”)

A single line is required to print, less lines of code compared to other languages.

We don’t need to manage the curly braces here , intendation matters.

**History of Python**

* Python laid it’s foundation in 1980’s.
* Python was introduced by Guido Van Rossum in 1989.
* The implementation of Python was started in December 1989,at CWI(Centrum Wiskunde & Informatica) in Netherland.
* First version of Python was introduced in 1991.
* In February 1991, van Rossum published the code (labelled version 0.9.0).
* Python is influenced by following programming languages.

ABC language

Modula-3 language(van Rossum early worked on these languages)

* In 1994, Python 1.0 was released with new feature which map, filter, lambda.
* Python 2.x added some features like comprehensions, garbage collection.
* After Python 2.x Python 3.x released on 2008.
* Python latest version Python 3.12.2

**Features of Python**

* Easy to learn and implement:

Very easy syntax

* Open Source.

Can download form python.org and its packages are free to use.

* Broad standard library.

Django, numpy, matplotlib….you can find a large number of libraries for every use. For web development django, if you go for machine learning numpy, pandas etc . There is a wast collection of libraries for you.

* Cross platform.

Can operate on different platforms

Windows, Linux ,…

* Work on interpreter logic.
* Multi paradigm language.

Can follow multiple programming paradigms. You can treat Python in Object Oriented way, procedural as your wish.

* High level programming.
* Extendable language.

You can use Python with other programming languages.

* Expressive programming language.

Just by seeing the syntax you can understand what the code means.It is that simple syntaxed programming language.

print(“Hello Athira!”)

* GUI programming language.

**Applications of Python**

* Network programming
* Data Analysis
* Robotics
* Website and application development(Django framework)
* Desktop application(tinker)
* Games development
* Web scraping
* Data visualization
* Scientific calculation
* Machine learning and Artificial Intelligence
* 3D Application development
* Audio and video software development

**Companies using Python**

Google

YouTube(for recommendations)

Instagram( Image and video processing)

NASA(scientific calculation)

Netflix(to make better video buffering)

Facebook(to handle large amount of data)

Quora(for reading writing readability)

Dropbox

uTorrent( for peer to peer connection)

We doesn’t have semicolon in Python.

**How to Download & Install Python on Windows**

You can download Python from it’s official site python.org

Remember to select Add Python to path

pip , which is used to download and install other Python packages.

**Python Identifiers**

* Starts with letter A to Z or a to z or an underscore followed by followed by letters, digits.
* No special characters @, #, $ allowed in identifiers.
* Do not start with digits, can use digits inbetween or end.
* Case sensitive programming language.
* Class names start with an uppercase letter.
* Keywords are reserved and can’t be used as identifiers.

**Python Keywords**

and exec not

assert finally or

break for pass

class from print

continue global raise

def if return

del import try

elif in while

else is with

except lambda yield

**Variables**

* Name which is used to refer memory location.
* Variables used to hod value.
* Don’t need to specify the type of the variable.

**Data Types**

Immutable Mutable

-Integer -list

-float -set

-boolean -dictionary

-string

-tuple

type() function returns the type of the variable.

a = 10 - int

name = “Athira” - string

amount = 12.5 - float

data = (1,2,”hello”,2.3, True) - tuple

data = [1,2,”hello”, 2.4, True] - list

data = {1,2,”hello”,2,4, True} - set

info = {“name”:”ammu”, “age”:10} - dictionary

**Operators**

Arithmetic operators

Comparison operators

Assignment operatos

Logical operatos

Membership operators

Identity operators

**Arithmetic operators**

Addition

Substraction

Multiplication

Division

Remainder(%)

21/10 = 1 - remainder

Exponent(\*\*)

2\*\*3 = 8

Floor division(//)

The quotient

21//10 = 2

**id() –**

a = 10

b = 10

print(id(a), id(b)) - same memory allocation for same value.

**Comparison Operators**

Comparison operators returns true or false values

==

!=

<=

>=

>

<

**Assignment Operators**

**=**

**+=**

**-=**

**\*=**

**/=**

**%=**

**\*\*=**

**//=**

**Shorthand operators**

**a = 10**

**a+=1**

**Logical Operators**

and

or

not

**Membership Operators**

**in**

It is evaluated to be true if the first operand is found in the second operand.

(list, tuple, dictionary)

**not in**

It is evaluated to be true if the first operand is not found in the second operand.

(list, tuple, dictionary)

a = “hello”

print(“e” in a)

print(“o” not in a)

**Identity Operators**

is

It is evaluated to be true if the referense present at both sides point to the same object.

is not

It is evaluated to be true if the referense present at both sides do not point to the same object.

a = 10

b = 10

print(a is b)

print(id(a))

print(id(b))

-5 - 256 (values get reassigned)

a = [1, 2, 3]

b = [1, 2, 3]

print(a is b) - false

In the case of immutable data types reassign happens but not in mutable data types.

**Conditional Statements**

**If**

If condition

# code to be executed

**if-else**

if condition:

#code to be executed

else:

#code to be executed

**if-elif-else**

if #expression 1:

#code to be executed

elif expression2:

#code to be executed

elif expression 3:

#code to be executed

else:

#code to be executed

**Nested if**

if condtion1:

#code to be executed

if condition:

#code to be executed

else:

#code to be executed