**Python**

Python is an interpreter, 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.

**Where we use python-**

Business Applications, Software Development, Education, Machine Learning and Artificial Intelligence, Data Science and Data Visualization.

**Let’s get started**

>> **Variables**- Where we have to store the values are called variables. For ex:- x=10. Here x-> variable and 10 is value.

>> **Data Types**- Types where we can store the data like string, integer, float etc.

Text Type: str

Numeric Types: int, float, complex

Sequence Types: list, tuple, range

Mapping Type: dict

Set Types: set, frozenset

Boolean Type: bool

Binary Types: bytes, bytearray, memoryview

**For ex-**

X=” Hello World” str

x = 20 int

x = 20.5 float

x = 1j complex

x = ["apple", "banana", "cherry"] list

x = ("apple", "banana", "cherry") tuple

x = range(6) range

x = {"name" : "John", "age" : 36} dict

x = {"apple", "banana", "cherry"} set

x = frozenset({"apple", "banana", "cherry"}) frozenset

x = True bool

x = b"Hello" bytes

x = bytearray(5) bytearray

x = memoryview(bytes(5)) memoryview

**>> Python casting-**

There may be times when you want to specify a type on to a variable. This can be done with casting. There are 3 types to do-

>> int() - constructs an integer number from an integer literal, a float literal or a string literal (providing the string represents a whole number)

>> float() - constructs a float number from an integer literal, a float literal or a string literal (providing the string represents a float or an integer)

>> str() - constructs a string from a wide variety of data types, including strings, integer literals and float literals.

**for ex-**

**int-**

x = int(1) # x will be 1

y = int(2.8) # y will be 2

z = int("3") # z will be 3

**float**

x = float(1) # x will be 1.0

y = float(2.8) # y will be 2.8

z = float("3") # z will be 3.0

w = float("4.2") # w will be 4.2

str

x = str("s1") # x will be 's1'

y = str(2) # y will be '2'

z = str(3.0) # z will be '3.0'

**>> python String-**

We can't concatenate the string with the numbers ( it can be any string). To do this, we have to use FORMAT keyword. Below is the example-

s=200

d= print(" the number is " +s)

print (d) ->>>output--> TypeError: can only concatenate str (not "int") to str

**The correct way to do is as mentioned below-**

s=200

d= " the number is {}"

print(d.format(s)) **-->> output-** the number is 200.

we can work with multiple numbers like-

s=200

e=300

f=400

g=s+e+f

d=" the number is {},{} and {} and the addition of 3 numbers are {} "

print(d.format(s,e,f,g)) **-->> output**- the number is 200,300 and 400 and the addition of 3 numbers are 900