What is Python

Python is simple and easy

English like language

Free and Open Source

High Level Language

Developed by Guido Vam Rossum

Python can be used on a server to create web applications.

CHARACTER SET

Alphabets: All capital (A-Z) and small (a-z) alphabets.

Digits: All digits 0-9.

Special Symbols: Python supports all kind of special symbols like, ” ‘ l ; : ! ~ @ # $ % ^ ` & * ( ) _ + – = { } [ ] \ .

White Spaces: White spaces like tab space, blank space, newline, and carriage return.

Other: All ASCII and UNICODE characters are supported by Python that constitutes the Python character set.

Python Comment

Comments can be used to explain Python code.

Comments can be used to make the code more readable.

Comments can be used to prevent execution when testing code.

```
#This is a single comment

-----------------------------------
"""
This is a comment
written in
more than just one line
"""
```

Variable:

IT Is a simple name given to the memory location in program

Name="IT PLUS SOLUTION"

Age=5

Price=55.5

RULES OF IDENTIFIERS

1. Identifiers can be combination of uppercase and lowercase letters, digits or an underscore(_). So **myVariable**, **variable_1**, **variable_for_print** all are valid python identifiers.
2. An Identifier can not start with digit. So while **variable1** is valid, **1variable** is not valid.
3. We can't use special symbols like !,#,@,%,$ etc in our Identifier.
4. Identifier can be of any length.

Data Types:

Integer
Float
String
Boolean
None

age=34
pi=3.14
Complex_num=10+2j
A=True
name="Divya"

```python
print(type(age))
print(type(pi))
print(type(complex_num))
print(type(A))
print(type(name))
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'bool'>
<class 'str'>
```

KEYWORDS

It is reserved word in Python

True and False (T and Fi s in upper case)

and	else	in	return
as	except	is	True
assert	finally	lambda	try
break	false	nonlocal	with
class	for	None	while
continue	from	not	yield
def	global	or	
del	if	pass	
elif	import	raise	

Types of Operator

Arithmatic Operator : + - * / % // **

Relational/Comparison Operator : ==,!=,<=,>=,<,>

Logical Operator :and or not

Assignment Operator :==,+=,-=,*=,/=,%=,**=

Type of Conversion

```
a, b = 1, 2.0
sum = a + b
```

```
#error
a, b = 1, "2"
sum = a + b
```

TYPE CASTING

Function	Description
int(y [base])	It converts *y* to an integer, and Base specifies the number base. For example, if you want to convert the string in decimal numbers then you'll use 10 as base.
float(y)	It converts *y* to a floating-point number.
complex(real [imag])	It creates a complex number.
str(y)	It converts *y* to a string.
tuple(y)	It converts *y* to a tuple.
list(y)	It converts *y* to a list.
set(y)	It converts *y* to a set.
dict(y)	It creates a dictionary and *y* should be a sequence of (key, value) tuples.
ord(y)	It converts a character into an integer.
hex(y)	It converts an integer to a hexadecimal string.
oct(y)	It converts an integer to an octal string

Input in Python

Input() -Statement is used to accept value (using keyword)from user

input( ) #result for input( ) is always a str

int (input( )) #int

float (input( )) #float

STRING INDEXING

D I V Y A
0 1 2 3 4

Str=“Divya”

Str[0] is ‘D’

Str[1] is ‘I’

STRING

Accessing Parts of a String

str[starting_idx : ending_idx] #ending idx is not included

str = “Riya Mayur Shinde”

str[1 : 4] is “**iya**”

str[: 4] is same as str[0 : 4]

str[1 :] is same as str[1 : len(str)]

String Function

str = “I am a coder.”

- str.**endsWith**(“er.“) #returns true if string ends with substr

- str.**capitalize**( ) #capitalizes 1st char

- str.**replace(** old, new **)** #replaces all occurrences of old with new str.**find(** word **)** #returns 1st index of

1st occurrence str.**count**(“am“) #counts the occurrence of substr in string

CONDITIONAL STATEMENT

```
if(condition):
        S tatement1
 elif(condition):
        S tatement2
 else:
        StatementN
```

CONDITIONAL STATEMENT

GRADE STUDENTS BASED ON STUDENT

marks >=90,grade="A"

90 >marks >=80,grade="B"

80 >marks >=70,grade="C"

70 >marks,grade="D"

LIST IN PYTHON

Lists are used to store multiple items in a single variable.

Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.

mylist = ["apple", "banana", "cherry"]

student = [”Karan”, 85, “Delhi”]   #student[0], student[1]..

student[0] = “Arjun”   #allowed in python

len(student)   #returns length

LIST SLICEING
Similar to string slicing

list_name[starting_idx : ending_idx] #ending idx is not included

marks = [87, 64, 33, 95, 76]

marks[1 : 4] is [64, 33, 95]

marks[: 4] is same as marks[0 : 4]

marks[1 :] is same as marks[: len(marks)]

marks[-3 : -1] is [33,95]

LIST METHOD

list = [2, 1, 3]

list.**append**(4) #adds one element at the end *[2, 1, 3, 4]*

list.**sort**( ) #sorts in ascending order *[1, 2, 3]*

list.**sort(** reverse=True **)** #sorts in descending order *[3, 2, 1]*

list.**reverse( )** #reverses list *[3, 1, 2]*

list.**insert**(idx, el) #insert element at index list = [2, 1, 3, 1]

list.**remove(**1**)**   #removes first occurrence of element

list.**pop(** idx **)**   #removes element at idx