

PYTHON

What is python?

programming language created by GUIDO VAN ROSSUM in 1991.

What python can do ?

create web apps(frontend as well as backend)

can connect to database systems and read & modify files.

handle big data

perform complex mathematics

software development etc.

Why python?

Simple syntax

Python runs on interpreter(line by line execution of code, less time to analyze code.)

In [8]:

```
# Note: Python is a case sensitive language.
```

Variables

Values can be retrieve by using variables in python. Variables are used to store values. Python gives importance to values than variables.

In [3]:

```
# FOR EXAMPLE :  
# Here a and b are variables and some values are assigned to them.  
a=5  
b=3.14
```

Data Types in python

Data types are the classification of data items. It represents the kind of value that tells what operations can be performed on a particular data.

Data types in python includes:

1.Integer

2.Float

3.string

4.Bool

5.List

6.Tuple

7.Dictionary

8.Sets

In [6]:

```
a=2
b=3.14
c='Kriti'
d= True
e=[1,2,3]
f=(1,2,3)
g={'1':'python','2':'c++'}
h={1,2,3}
```

In [7]:

```
# type() function : It returns the data type of the value
# assigned to the variable.
```

```
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
print(type(g))
print(type(h))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
<class 'bool'>
<class 'list'>
<class 'tuple'>
<class 'dict'>
<class 'set'>
```

In [12]:

```
# id() function : It returns address of the variable.  
  
print(id(a))  
  
'''  
In python address is given to the value not the variable therefore  
the number of address that can be assigned to a variable doesn't depend  
on the variable, it depends on the number of values assigned.  
  
'''  
  
a=4  
i=4  
id(a)==id(i) '''It returns true as the value assigned to a and i are  
same therefore, the address of a and i is same.'''
```

1806770727312

Out[12]:

True

Keywords and Identifiers

Keywords in python : special reserved words, or keywords, that have specific meanings and restrictions around how they should be used. False ,else , in ,True, is,for,while,etc.

Identifiers in python: An identifier is a name given to entities like class, functions, variables, etc. It helps to differentiate one entity from another.

Rules for writing identifiers : 1. Identifiers can be a combination of letters in lowercase (a to z) or uppercase (A to Z) or digits (0 to 9) or an underscore _. Names like myClass, var_1 and print_this_to_screen, all are valid example. 2. An identifier cannot start with a digit. 1variable is invalid, but variable1 is a valid name. 3. Keywords cannot be used as identifiers. 4. We cannot use special symbols like !, @, #, \$, % etc. in our identifier

Taking Input

```
a1=input()
```

By default python takes string as input.

Type Conversion

It is the conversion of object from one data type to another data type.

Implicit Type Conversion is automatically performed by the Python interpreter. Python avoids the loss of data in Implicit Type Conversion.

Explicit Type Conversion is also called Type Casting, the data types of objects are converted using predefined functions by the user.

In Type Casting, loss of data may occur as we enforce the object to a specific data type.

In [10]:

```
a=input()
print(type(a))
b=input()
print(type(b))
```

```
1
<class 'str'>
3.12
<class 'str'>
```

In [8]:

```
a=int(input())
print(type(a))
```

```
3
<class 'int'>
```

In [7]:

```
a=float(input())
print(type(a))
```

```
3.12
<class 'float'>
```

In [6]:

```
a=str(input())
print(type(a))
```

```
pyhton
<class 'str'>
```

Output

For giving output we use print and eval function.

In [12]:

```
a= str(input("Enter a string :"))
print("OUTPUT :",a) #output
```

```
Enter a string :This code will print output
OUTPUT : This code will print output
```

In [20]:

```
# eval()
# If we write-
print('2+3')
# It will return the same value as it is a string. Whereas, If we write-
eval('2+3')
# It will evaluate the expression in string format and return
# the sum of the numbers in string.
```

2+3

Out[20]:

5

In [41]:

```
# end :

print("This is a python code",end=" *@")
# It will print *@ at the end of the statement.
```

This is a python code *@

STRINGS

Strings are immutable i.e. we can not change the elements in string once assigned.

For example:

If we take string as Python and now we want to change the letter y to letter z.

string=python if we want to make the string= pzthon it is not possible.It'll generate an error.

In [39]:

```
# Defining a string.
string = 'hello!'
# Taking string as user input
string1=str(input("String1 :"))
```

String1 :hello!

String Slicing

-5 -4 -3 -2 -1

H E L L O

0 1 2 3 4

It is of two types +ve indexing and -ve indexing. +ve indexing starts from 0 and -ve indexing starts from -1

In [43]:

```
string='python program'  
string[:] # print the whole string
```

Out[43]:

'python program'

In [44]:

```
string[1:4]  
# prints starting from index number 1 upto index number 3  
#(excludes index number 4,takes(n-1))
```

Out[44]:

'yth'

In [45]:

```
string[::-1] # Reverse the string
```

Out[45]:

'margorp nohtyp'

In [56]:

```
# String Concatenation  
  
str1='Python'  
str2='program'  
  
op=str1+str2  
print(op)  
  
op2=str1+" "+str2 # prints the space between the two strings  
print(op2)  
  
op3=str1*5  
print(op3)
```

Pythonprogram
Python program
PythonPythonPythonPythonPython

string methods

https://www.w3schools.com/python/python_strings_methods.asp
(https://www.w3schools.com/python/python_strings_methods.asp)

LISTS

In [58]:

```
# Defining lists

list1=[] # creates an empty list
list2=[1,2,3]
list3=[1,'kriti',3.14,True]
# List can have all type of data types, It is mutable.
# List is an ordered data type.
# It allows duplicates.
```

In [59]:

```
# len() function : It gives length of the list.
len(list2)
```

Out[59]:

3

In [60]:

```
type(list3)
```

Out[60]:

list

In [66]:

```
# list constructor : list()
li=list("1234")
li1=list(('1','2','3'))
print(li)
print(li1)
```

```
['1', '2', '3', '4']
['1', '2', '3']
```

In [67]:

```
# Accessing List items : List items are accessed by index numbers

li[2] # 2 is the index number of list li.
```

Out[67]:

'3'

In [69]:

```
list_1=['python','is','a','programming','language']
list_1[0:3]
#it will access list items from index 0 to index 2
#as it includes (n-1)th index at last and excludes nth index.
```

Out[69]:

```
['python', 'is', 'a']
```

In [71]:

```
# Changing list items

list_1=['python','is','a','programming','language']
list_1[2]='an'
list_1[3]='easy'
list_1
```

Out[71]:

```
['python', 'is', 'an', 'easy', 'language']
```

In [81]:

```
# Adding items in list

list=[1,2,3,4]
list.append((5,6))    # append() adds the item at the end of the list.
print("Adding list by append method :",list)

list.extend((7,8))    # extend() also adds the item at the end.
print("Adding item in list by extend method :",list)

list.insert(0,10)
# insert() adds item at the specified position in list.
# insert(a,b) here a is the index number at which we want
# to add element and b is the element.
print("Adding item in list by insert method :",list)
```

Adding list by append method : [1, 2, 3, 4, (5, 6)]

Adding item in list by extend method : [1, 2, 3, 4, (5, 6), 7, 8]

Adding item in list by insert method : [10, 1, 2, 3, 4, (5, 6), 7, 8]

In [85]:

```
# Deleting items in list

list=[1,2,3,4]
list.remove(2)    # remove() method deletes the given element.
#That is the value to be deleted is passed as the parameter.
print("Deleting item by remove method :",list)

list.pop()    # pop() method deletes the last item from list by default.
print("Deleting item by remove method :",list)

list.pop(0)
# index number is passed as a parameter in pop method
# to delete an item from specific position.
print("Deleting item by pop method :",list)
```

Deleting item by remove method : [1, 3, 4]

Deleting item by remove method : [1, 3]

Deleting item by pop method : [3]

List Methods

https://www.w3schools.com/python/python_lists_methods.asp
(https://www.w3schools.com/python/python_lists_methods.asp).

Practice Questions

- 1.What is the output of `print(2%6)`
- 2.What is the output of `print(2 * 3 ** 3 * 4)`
- 3.What is the output of the following code `x = 6 y = 2 print(x ** y) print(x // y)`
- 4.What is the output of the following assignment operator `y = 10 x = y += 2 print(x)`
- 5.What is the value of the following Python Expression `print(36 / 4)`
- 6.What is the output of the following String operations `str1 = 'Welcome' print (str1[:6] + ' PYnative')`
- 7.string='hello python' capitalize the first word of the given string using string method.
- 8.string='1234567890' what is the output of `string[0:10:2]`
- 9.Take a string as an input from user and then reverse it.
- 10.copy a list using copy function `List = ['a', 'b', 'c', 'd']`

- 11.What is the output of the following list function?

```
sampleList = [10, 20, 30, 40, 50]
sampleList.pop()
print(sampleList)
sampleList.pop(2)
print(sampleList)
```

- 12.What is the output of the following list function?

```
sampleList = [10, 20, 30, 40, 50]
sampleList.append(60)
print(sampleList)
sampleList.append((70,80))
print(sampleList)
```

- 13.Write the difference between `extend()` and `append()`.
- 14.Take two lists as user input and add them.
- 15.print the following list: `[1,2,[3,4],5,6,[7,8,9]]`

In []:

