Learning Python The Precise Way

print('*' * 10)

The above statement will print 10 * as output.

O/P: **********

input()
For taking user input

E.g: WAP to greet an user

name= input('What is your name? ')
print('Hi ' + name)

o/p:
What is your name? Shahid
Hi Shahid

Variable Type changing

By default every value in Python is taken as string so we have to change the type from string to our desired type, we will visualize this through an example

E.g: WAP to calculate age

birth=input('Birth Year = ')
age= 2022 - **int(birth)**
print('Your Age is: ' ,age)

O/P:

Birth Year = 1999
Your Age is: 23

Now we clearly notice that how we calculated the age and changed the input value to integer
Similarly we can change to float,double or any other type we desire, by using float(),double(),.....

String Operations

There are a number of ways to define strings.

1. Using single quotes

2. Double quotes
3. Triple quotes(For writing multiple lines)

We will see this through an example.

E.g:
course="Python's Easy" → *using single quote in string*
course2='Python is "Not Easy" ' → *using double quote in string*

course3='''Python is really easy → *storing multiple lines*

programming language'''

print(course,course2,course3)

O/P:
Python's Easy Python is "Not Easy" Python is really easy

programming language

We can use **negative index** in char arrays(strings) in Python. Simply it will count from backwards.

E.g:
m='Ramanujan'
print(m[2],m[-1],m[-3],m[0])

O/P:
m n j R

E.g:2:
m='Ramanujan'
print(m[2],m[-1],m[-3],m[0])
print(m**[0:4]**) → *This line will print the characters starting from index 0 till 3*

O/P:
m n j R
Rama

E.g3:

m='Ramanujan'
print(m[2],m[-1],m[-3],m[0])
print(m[0:4])

print(m[0:]) → *This line will print the characters starting from index 0 till the end*

print(m[3:]) → *This line will print the characters starting from index 3 till the end*
print(m[:]) → *This line will print the whole string*
dup=m[:] → *This line will store the whole string in dup variable.*
print(dup)

o/p:

m n j R
Rama
Ramanujan
anujan
Ramanujan
Ramanujan

Formatted Strings

Formatted strings are particularly useful in situations where you dynamically generate some text with your variables.

Let's say we have two variables first name and last name. So first we set this to John, and last we set this to Smith.

fname='John'
lname='Smith'
msg= **f ' {fname} [{lname}] is a coder '** → *This is how we formatted our string*
print(msg)

o/p: John [Smith] is a coder

So to define formatted strings, prefix your strings with an F and then use curly braces to dynamically insert values into your strings.

String methods

replace(),upper(),lower(),find(), in(its a keyword which returns result in boolean)
We will illustrate the methods with an example.

E.g:

course='Python for Beginners'
print(course.upper())

```
print(course)  #to show that no changes are implemented on the variable
print(course.lower())
print(course.find('P'))    #returns index
print(course.find('for'))    #returns index of the first char
print(course.find('you'))    #returns -1 since its does not exist in the string
print(course.replace('Beginners','Absolute Beginners'))
print(course)            #to show that no changes are implemented on the variable
print('Mython' in course)
```

O/P:

```
PYTHON FOR BEGINNERS
Python for Beginners
python for beginners
0
7
-1
Python for Absolute Beginners
Python for Beginners
False
```

len() is used to calculate the no. of characters in string.

Arithmetic Operators

- \+ Addition
- \- Subtraction
- (*) Multiplication
- (**) Exponent
- / Division(returns whatever the result is)
- // Division(returns only the integer part)

Augmented operators can also be implemented like C or C++,
+= , -=, *= , /=

Operator Precedence

Parenthesis or brackets then, Exponential i.e 2 ** 3 then,
Multiplication or Division then, Addition or subtraction

Math Functions

https://docs.python.org/3/library/math.html

The above link will provide all the math functions which are included in the math module.

Technically in Python we have a handful of built in functions for performing mathematical operations, if you want to write a program that involves complex mathematical calculations, you need to import the math module.

A module in Python is a separate file with some reusable code. We use these modules to organize our code into different files. As a metaphor think of a super market. When you go to a super market you see different sections for fruits and vegetables, cleaning products, junk food and so on. Each section in the super market is like a module in Python. So in Python we have this math module which contains a bunch of reusable functions for performing mathematical calculations.
We use keyword **import** to include any module

E.g:

```
import math

print(math.ceil(2.9))
print(math.floor(2.5))
```

o/p:

```
3
2
```