

Python First Program

```
In [1]: print('Welcome to Python')
```

Welcome to Python

```
In [2]: 5+6
```

```
Out[2]: 11
```

First we learn : Character Set

- All character small a-z, Capital A-Z, Digits 0-9, Special characters <, >, #, @, !, ^, & Characters are case sensitive

After Character Set We learn words

- Words are of two types one is Identifiers and Reserve words
- Identifiers are also have two types 1. Variables and 2. Constants
- In Python we rarely use constants. Technically here we can't use constants directly, but can use indirectly.
- Let us understand variables

For Identifiers we always havee some rules

- First Character is always letter not a digit
- Second character may be letter or digit
- No special character allowed in identifier's G#a% g-a@n) not allowed
- Expection : only underscore allowed anywhere in the identifier
- Length of Identifier

Take two variables and print sum

```
In [3]: a=5  
b=6  
print(a+b)
```

11

Take one variable have a value a then increment with 6 and then print the value

```
In [4]: a=5;
        a=a+6;
        print(a)
```

11

Take two variables and print sum. (Variable a and A)

```
In [5]: a=5;
        A=6;
        a+A
```

Out[5]: 11

Take two variables and print sum. All statements in a single line

```
In [6]: a=5; b=6; a+b      # Here we can understand terminator ;
```

Out[6]: 11

Take two variables using multiple assignment statement and print sum.

```
In [7]: a, b =5
        a+b
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-7-ddab96a99266> in <module>
----> 1 a, b =5
      2 a+b

TypeError: cannot unpack non-iterable int object
```

Above statement is incorrect you have to assign a value of a and

Take two variables using multiple assignment and print sum.

```
In [8]: ajay, Ajay =5, 6
        ajay+Ajay
```

Out[8]: 11

```
In [9]: a, b,c=5,6,7 # Three Variables
        a+b
```

Out[9]: 11

Multiple Assignment with Same Values as in first and second cell

```
In [10]: a=5 # Same Value to all variables
        b=5
        c=5
```

```
In [11]: a=b=c=5 # Same value to multiple variables with Assignment
        print(a)
```

5

```
In [12]: 1a=5 # # you can't start identifier with digit as per rule
```

```
File "<ipython-input-12-e960d5019be4>", line 1
      1a=5 # # you can't start identifier with digit as per rule
      ^
```

SyntaxError: invalid syntax

```
In [13]: a1=5 # Second character may be digit as per rule\n"
        a1
```

Out[13]: 5

```
In [14]: a)b=5 # No special character is allowed
```

```
File "<ipython-input-14-ae982bc09c90>", line 1
      a)b=5 # No special character is allowed
      ^
```

SyntaxError: invalid syntax

```
In [15]: a_b=5 # Exception _(underscore) is allwed anywhere within the identifier
        _ab=6
        ab_=7
        a_b+_ab*ab_
```

Out[15]: 47

```
In [16]: aaaaaaaaaaaaaaaaaaaaaaaaaa=7 # Identifier Length
        aaaaaaaaaaaaaaaaaaaaaaaaaa
```

Out[16]: 7

```
In [17]: aaaaaaaaaaaaaaaaaaaaaaaaaa
        aaaaaaaaaaaaaaaaaaaaaaaaaa=7
        aaaaaaaaaaaaaaaaaaaaaaaaaa
        aaaaaaaaaaaaaaaaaaaaaaaaaa
```

Out[17]: 7


```
In [24]: # Escape Sequence \t, \b, \n

print("Welcome to Python \n I am here to assist you \n Now we are starting this tour")
```

```
Welcome to Python
I am here to assist you
Now we are starting this tour
```

```
In [25]: print("Welcome to Python \tI am here to assist you \tNow we are starting this tour")
```

```
Welcome to Python      I am here to assist you      Now we are starting t
his tour
```

```
In [26]: print("Welcome to Python \bI am here to assist you \bNow we are starting this tour")
```

```
Welcome to Python  I am here to assist you  Now we are starting this tour
```

```
In [27]: eno=101
name="Gagan"
salary=100000.00
print(eno, '\t', name, salary)
```

```
101      Gagan 100000.0
```

```
In [28]: # Separator separate outputs with some character instead of space

print(1,2,3,4, sep='-')
```

```
1-2-3-4
```

```
In [29]: ##Similarly end with some character

print(1,2,3,4, sep='*', end='@')
```

```
1*2*3*4@
```

```
In [30]: # Formatting using index value in sequence

name = 'Ajay'
time = 5
print('Hello {}, we will meet at {} P.M.'.format(name,time))
```

```
Hello Ajay, we will meet at 5 P.M.
```

```
In [31]: # Formating using index value in {}

print('I am travelling by {0} and going to {1}. Will reach there by {2} P.M.'.
      format('Train','Chandigarh', 5))

# here {} we are specifying index number
```

I am travelling by Train and going to Chandigarh. Will reach there by 5 P.M.

```
In [32]: here {} we are specifying index number but not in sequence

print('I am travelling by {1} and going to {2}. Will reach there by {0} P.M.'.
      format('Train','Chandigarh', 5))
```

```
File "<ipython-input-32-6bed7f812767>", line 1
      here {} we are specifying index number but not in sequence
          ^
```

SyntaxError: invalid syntax

```
In [33]: # here {} we are specifying variables inplace of index

print('Hello {name}, {greeting}'.format(greeting = 'Goodmorning', name='Ajay'
))
```

Hello Ajay, Goodmorning

```
In [34]: # here {} we are specifying variables inplace of index. But if you don't have
          values for all variable then you will get error

print('Hello {name}, {greeting}'.format(greeting = 'Goodmorning', name))
```

```
File "<ipython-input-34-027890ffa0f7>", line 3
      print('Hello {name}, {greeting}'.format(greeting = 'Goodmorning', name))
                                                ^
```

SyntaxError: positional argument follows keyword argument

```
In [35]: # Specifying field width

x = 12.3456789
print('The value of x is %3.2f', x)

# Specify % with field width and variable both otherwise you will get output
like this
```

The value of x is %3.2f 12.3456789

```
In [36]: x = 12.3456789
print('The value of x is %3.2f' %x)
```

The value of x is 12.35

```
In [37]: print('The value of x is %5.2f'%x)
```

The value of x is 12.35

```
In [38]: x = 12.3456789
print('The value of x is %10.2f' %x)
```

The value of x is 12.35

```
In [39]: y = 12
print('The value of x is %2d' %y)
```

The value of x is 12

```
In [40]: y = 12
print('The value of x is %8d' %y)
```

The value of x is 12

```
In [41]: y = 12.3456
print('The value of x is %8d' %y)
```

The value of x is 12

```
In [42]: print("Multiplication of {} and {} is {}".format(x,y), x*y)
```

Multiplication of 12.3456789 and 12.3456 is 152.41481342784

Input Statements : input function : input()

```
In [43]: a=input()
```

25

```
In [44]: a
```

```
# when you are giving input using input function(means by keyboard), your input always be of string input, \n",
# for that we will use type casting"
```

```
Out[44]: '25'
```

```
In [45]: a+2
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-45-a917fe5125e3> in <module>
----> 1 a+2
```

TypeError: can only concatenate str (not "int") to str

```
In [46]: b=int(input()) # typecast(input()), this will take integer inputs
```

45

In [47]: `b+2`

Out[47]: 47

In [48]: `c=input('Please Enter the value of c :')`

In these interactive input statements there is no need of seprate print statements

Please Enter the value of c :682

In [49]: `c`

Out[49]: '682'

In [50]: `name = input('Please Enter Your Name :')`

`print('Hello ', name)`

`print('Hello ', name , 'How Are you?')`

`print('Hello '+ name) # you can also take + inplace of , compare both statements`

`print('Hello '+ name + 'How Are you?')`

Please Enter Your Name :Gagan Deep

Hello Gagan Deep

Hello Gagan Deep How Are you?

Hello Gagan Deep

Hello Gagan DeepHow Are you?

Multiple Input in a single statement :

Using split() method and

Using List comprehension

Using split() method

In [51]: `#Multiple Input : Using split() method and Using List comprehension`
`a, b = input("Enter a two values: ").split()`

Enter a two values: 23 45

In [52]: `print('Value of a is ' + a + ' Value of b is ' + b)`

Value of a is 23 Value of b is 45

In [53]: `print(a+b) # String Addition or Concatanation`

2345


```
In [54]: a, b = input("Enter a two values: ").split(',')
```

Enter a two values: 235, 456

```
In [55]: print('Value of a is ' + a + ' Value of b is ' + b)
```

Value of a is 235 Value of b is 456

```
In [56]: i,j = map(int, input("Enter the values: ").split())
```

Enter the values: 23 45

```
In [57]: i+j
```

Out[57]: 68

```
In [58]: x, y = [int(x) for x in input("Enter two values: ").split()]
```

Enter two values: 23 45

```
In [59]: print('Value of x is ' + x + ' Value of y is ' + y + 'and the Sum is ' + x+y)
# Check Error what is telling you
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-59-7a12fed44979> in <module>
----> 1 print('Value of x is ' + x + ' Value of y is ' + y + 'and the Sum is
' + x+y)
```

TypeError: can only concatenate str (not "int") to str

```
In [60]: print('Value of x is ', x , ' Value of y is ', y , 'and the Sum is ' , x+y)
```

Value of x is 23 Value of y is 45 and the Sum is 68

```
In [ ]:
```