



<https://apssdc.in>

APSSDC

Andhra Pradesh State Skill Development Corporation



Python Basics

Today Objectives

- Output & Input
- Variables in Python
- Type Conversions in Python
- Operators in Python
- Conditional Statements

In [1]: ▶ 1 `print("Hello World")`

Hello World

In [3]: ▶ 1 `print("Today's date is",29)`

Today's date is 29

In [4]: ▶ 1 `print("Today's date is",29, sep = '-')`

Today's date is-29

In [5]: ▶ 1 `print(1,2,3,4,5,6, sep = '\t')`

1 2 3 4 5 6

In [6]: ▶ 1 `print(1,2,3,4,5,6, sep = 'abc')`

1abc2abc3abc4abc5abc6

In [7]: ▶ 1 `print("Hello World")`
2 `print("Hello World")`
3 `print("Hello World3")`

Hello World
Hello World
Hello World3

```
In [8]: ▶ 1 print("Hello World", end = '\t')
          2 print("Hello World")
          3 print("Hello World3")
```

```
Hello World    Hello World
Hello World3
```

Variables

It is the named memory location which holds some value

Properties of declaring a variable

- It can contain AZaz09_
- It shouldn't start with a number
- No special characters should included in variable name except _
- Keywords are not allowed
- Case Sensitive

```
In [9]: ▶ 1 a = 10
          2 A = 5
```

```
In [12]: ▶ 1 a = 55
           2 _a = 2256
           3 # 1a = 66
           4 # a-b = 23+66
           5 _ = 625
           6 a9 = 'Python'
           7 a9_ = 'APSSDC'
```

Data Types in Python

- Primary Data Types -> int, float, complex, string, boolean
- Secondary Data Types -> Containers -> List, Tuple, Dictionary, set

```
In [13]: ▶ 1 a = 55
           2 b = 55.66
           3 c = 5 + 6j
           4 d = True
           5 e = 'a+5/.,;'
           6 f = ['python', 65, 5+6j, 55.665]
           7 g = ('python', 65, 5+6j, 55.665)
           8 h = {'Name': 'Python'}
           9 i = {1,2,3,4}
```

```
In [14]: 1 print(type(a))
```

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

```
In [15]: 1 a = 55.66  
2  
3 print(type(a))
```

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

```
In [17]: 1 print(type(c), type(i))
```

```
<class 'complex'> <class 'set'>
```

```
In [18]: 1 a = input()
```

```
123
```

```
In [19]: 1 abc = 556.6
```

```
In [20]: 1 print(type(a))
```

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

```
In [21]: 1 b = input()
```

```
[1, 2, 3]
```

```
In [22]: 1 print(b, type(b))
```

```
[1, 2, 3] <class 'str'>
```

Type Conversions

```
In [25]: 1 a = int(input())  
2  
3  
4 print(a, type(a))
```

```
123
```

```
123 <class 'int'>
```

```
In [26]: ▶ 1 a = int(input())
          2
          3
          4 print(a, type(a))
```

256.66

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-26-f4932f7a7800> in <module>
----> 1 a = int(input())
      2
      3
      4 print(a, type(a))
```

ValueError: invalid literal for int() with base 10: '256.66'

```
In [28]: ▶ 1 a = float(input())
          2
          3
          4 print(a, type(a))
```

125.336
125.336 <class 'float'>

Number System

- Decimal -> 10 -> 0-9
- Binary -> 2 -> 0,1
- Octal -> 8 -> 0-7
- Hexadecimal -> 16 -> 0-F

```
In [29]: ▶ 1 a = 55
          2
          3 print(bin(a), oct(a), hex(a))
```

0b110111 0o67 0x37

```
In [30]: ▶ 1 print(int('110111', 2), int('67', 8), int('37', 16))
```

55 55 55

```
In [31]: ▶ 1 print(ord('a'))
```

97

In [32]:



```
1 print(chr(97))
```

a

Operators in Python

- Arithmetic Operator -> +, -, *, /, %, //, *
- Logical -> and, or, not
- Comparison/relational -> <, >, <=, >=, ==, !=
- Bitwise Operator -> &(and), |(or), ~(Not), ^(XOR), >>(right shift), <<(left shift)
- Assignment Operator -> +=, -=, /=, %=, //=, **=
- Identity operators -> is, not is
- Membership operators -> in, not in

In [35]:



```

1 a = 4
2 b = 10
3
4
5 print(a+b, a-b, a*b, a/b, a % b, a//b, a ** 2, sep = '->')
```

14->-6->40->0.4->4->0->16

and (inp1 * inp2)

inp1	inp2	inp1 and inp2
T	T	T
T	F	F
F	T	F
F	F	F

or (inp1 + inp2)

inp1	inp2	inp1 or inp2
T	T	T
T	F	T
F	T	T
F	F	F

not

inp1	not inp1
F	T
T	F

```
In [40]: ▶ 1 a = 1
          2 b = 0
          3 c = '0'
          4
          5 print(a and b, a or b, not a, a and c, type(a and c), not c, sep = '->')

0->1->False->0-><class 'str'>->False
```

```
In [39]: ▶ 1 print(c and a, a or c)

1 1
```

```
In [41]: ▶ 1 a = 5
          2 b = 6
          3 c = 0
          4
          5 print(a and b)

6
```

```
In [42]: ▶ 1 print(a and c, c and a)

0 0
```

```
In [43]: ▶ 1 a = 5
          2 b = 6
          3
          4 print(a < b, a > b, a == b)

True False False
```

Bitwise Operators

&, |, ~, ^

```
a = 4
b = 10
```

```
0000 -> 0 -> and
1110 -> 14 -> or
```

```
~a
4 -> 0100
    0001
-0101 -> *5
```

```
In [44]: ▶ 1 a = 4
           2 b = 10
           3
           4 print(a & b, a | b, ~a)
```

0 14 -5

###

XOR -> A'B + AB'

inp1	inp2	inp1 and inp2
T	T	F
T	F	T
F	T	T
F	F	F

a = 4 -> 0100

b = 10 -> 1010

1110 -> 14

```
In [45]: ▶ 1 print(a ^ b)
```

14

Left Shift and right shift

a = 4 -> 0100

a << 2 -> 010000 -> 16

a << 2 -> 0001 -> 1

```
In [46]: ▶ 1 a = 4
           2
           3 print( a << 2, a >> 2)
```

16 1

```
In [47]: ▶ 1 a = 5
           2
           3 a = a + 10
           4 print(a)
           5 a += 10
           6 print(a)
           7 a -= 5
           8 print(a)
           9 a **= 2
          10 print(a)
```

```
15
25
20
400
```

Membership Operators

```
In [48]: ▶ 1 a = 'Python'
           2 b = 'Py'
           3 c = 'py'
           4
           5 print(a in b, a in c, a not in c)
```

```
False False True
```

Identity Operators

```
In [61]: ▶ 1 a = 5
           2 b = 5.5
           3 c = 5
           4 d = [1,2,3]
           5 e = [1,2,3]
           6 f = d
```

```
In [55]: ▶ 1 print(f)
```

```
[1, 2, 3]
```

```
In [56]: ▶ 1 print(a == c, d == e, d == f)
```

```
True True True
```



```
In [58]: ▶ 1 print(id(a))
          2 print(id(c))
          3 print(id(d))
          4 print(id(e))
          5 print(id(f))
```

```
140719521806240
140719521806240
2762794741312
2762794754112
2762794741312
```

```
In [60]: ▶ 1 d.append('Python')
          2
          3 print(d)
          4 print(f)
```

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

```
In [64]: ▶ 1 print(d,e,f)
          2 print(d is e, d is f, e is f, d is not e)
```

```
[1, 2, 3] [1, 2, 3] [1, 2, 3]
False True False True
```