

Comments in python

- Single line comment
- Multi line comment

In [2]:

```
1 # single line comment
2 """
3 Multi
4 line
5 comment
6 """
7 print('Good Afternoon all')
```

Good Afternoon all

Python Basics

In [3]:

```
1 # print a message
2 print('python programming')
```

python programming

In [4]:

```
1 print('This keerthi\'s pen')
```

This keerthi's pen

In [5]:

```
1 print("This keerthi's pen")
```

This keerthi's pen

In [8]:

```
1 print('c:docs\\navin')
2 print('c:docs\\navin')
```

c:docs
avin
c:docs\\navin

In [7]:

```
1 print(r'c:docs\\navin')
```

c:docs\\navin

In [12]:

```
1 myname = 'keerthi'
2 print(5 * myname )
```

keerthikeerthikeerthikeerthikeerthi

In [11]:

```
1 print(5*' keerthi')
```

keerthi keerthi keerthi keerthi keerthi

In [14]:

```
1 print('keerthi\nkeerthi\nkeerthi')
```

keerthi
keerthi
keerthi

In [15]:

```
1 print('Rohit\n'*5)
```

Rohit
Rohit
Rohit
Rohit
Rohit

Variables

- A python variable is a reserved memory location to store values.

In [16]:

```
1 a = 123 # single variable assignment
2 print(a)
```

123

In [17]:

```
1 a = 'apssdc'
2 print(a)
```

apssdc

In [25]:

```
1 x,y,z = 12,10,9 # multiple variable assignment with multiple values
2 print(x,y,z)
3 print(x, '\n', y, '\n', z)
```

12 10 9

12

10

9

In []:

```
1 m = 134
2 q=w=e=m # multiple variable assignment
3 print(w)
```

Data Types and Conversions

- int
- float
- string

In [26]:

```
1 n1 = 13
2 type(n1)
```

Out[26]:

int

In [27]:

```
1 n1 = 13.8
2 type(n1)
```

Out[27]:

float

In [28]:

```
1 n1 = 'apssdc'
2 type(n1)
```

Out[28]:

str

In [29]:

```
1 n1 = 13.8
2 print(int(n1))
```

13

In [30]:

```
1 n1 = 13
2 print(float(n1))
```

13.0

In [32]:

```
1 n1 = 13
2 n2 = 13.8
3 n3 = 'karthik'
4 print(type(n3))
5 print(type(n2))
6 print(type(n1))
7
```

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

In [33]:

```
1 n1=13
2 s = str(n1)
3 print(type(s))
```

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

Indentation

In [36]:

```
1 n1 = 13
2 n2 = 12
3 if n1 > n2:
4     print('n1 is greater than n2')
```

n1 is greater than n2

Reading input dynamically

In [40]:

```
1 x = input()
2 print('Given input is:',x)
3 print(type(x))
```

python

Given input is: python

<class 'str'>

In [41]:

```
1 x = input()
2 print('Given input is:',x)
3 print(type(x))
```

123

Given input is: 123

<class 'str'>

In [43]:

```
1 n1 = input()
2 n2 = input()
3 print('Addition is:',n1+n2)
```

12

13

Addition is: 1213

In [44]:

```
1 n1 = int(input())
2 n2 = int(input())
3 print('Addition is:',n1+n2)
```

12

13

Addition is: 25

In [45]:

```
1 n1 = int(input())
2 print('Given input is:')
```

apssdc

ValueError

Traceback (most recent call last)

<ipython-input-45-9ae0f939f026> in <module>

```
----> 1 n1 = int(input())
      2 print('Given input is:')
```

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

Operators

1. Arithmetic Operators
2. Assignment Operators
3. Comparison Operators
4. Logical Operators
5. Identity Operators
6. Membership Operators
7. Bitwise Operators

1. Arithmetic Operators

In [47]:

```
1 a,b = 5,3
2 print(a+b)
3 print(a-b)
4 print(a*b)
5 print(a/b)
6 print(a%b)
7 print(a//b) # Floor Division
8 print(a**b) # Exponentiation
```

```
8
2
15
1.6666666666666667
2
1
125
```

2. Assignment Operator

In [49]:

```
1 a = 12
2 print(a)
```

```
12
```

In [52]:

```
1 a+=5 # a=a+5
2 print(a)
```

```
27
```

In [53]:

```
1 b = 5
2 b -= 3 # b=b-3
3 print(b)
```

```
2
```

3. Comparison Operators

In [55]:

```
1 n1,n2 = 5,3
2 print(n1==n2)
3 print(n1!=n2)
4 print(n1>n2)
5 print(n1<n2)
6 print(n1>=n2)
7 print(n1<=n2)
```

False
True
True
False
True
False

4. Logical Operators

and, or, not

In [57]:

```
1 a = 5
2 print(a<6 and a>2)
3 print(a>6 and a>2)
```

True
False

In [66]:

```
1 print(a<6 or a>2)
2 print(a>6 or a>2)
```

True
True

In [62]:

```
1 (a<6 or a>2)
```

Out[62]:

True

In [63]:

```
1 not(a<6 or a>2)
```

Out[63]:

False

5. Identity Operators

is, is not

In [67]:

```
1 x,y = 5,3
2 print(x is y)
```

False

In [68]:

```
1 x,y = 5,5
2 print(x is y)
```

True

In [69]:

```
1 x,y = 5,3
2 print(x is not y)
```

True

6. Membership Operators

in, not in

In [70]:

```
1 x = ['apple', 'banana', 'grapes']
2 print('apple' in x)
```

True

In [71]:

```
1 print('papaya' not in x)
```

True

In [72]:

```
1 print('papaya' in x)
```

False

7. Bitwise Operators

&, |, ^, >>, <<, ~

In [73]:

```
1 5 & 3
```

Out[73]:

1

In [74]:

```
1 5 | 3
```

Out[74]:

7

In [75]:

```
1 5^3
```

Out[75]:

6

In [76]:

```
1 ~5
```

Out[76]:

-6

In [77]:

```
1 5 >> 3
```

Out[77]:

0

In [78]:

```
1 5<<3
```

Out[78]:

40

In []:

```
1
```