Numbers

June 28, 2022

1 Numbers

1.0.1 Declaring and assigning 'Integers' in python

[1]:
```
x = 1
print(x)
print(type(x))
```

```
1
<class 'int'>
```

[2]:
```
#LongLength
y = 8823997732182756892129
print(y)
print(type(y))
```

```
8823997732182756892129
<class 'int'>
```

[3]:
```
#Negetive Integer
z = -125634
print(z)
print(type(z))
```

```
-125634
<class 'int'>
```

[4]:
```
a = 0b1100 #Binary
print(a)
print(type(a))
```

```
12
<class 'int'>
```

[5]:
```
b = 0o1100 #Octal
print(b)
print(type(b))
```

```
576
<class 'int'>
```

[6]:

```
c = 0x1100 #Hexadecimal
print(c)
print(type(c))
```

```
4352
<class 'int'>
```

1.0.2 Declaring and assigning “Float Numbers” in Python

[7]:

```
x = 12.3
print (x)
print(type(x))
```

```
12.3
<class 'float'>
```

[8]:

```
y = 12.9829379485794548679
print (y)
print(type(y))
```

```
12.982937948579455
<class 'float'>
```

[9]:

```
z = -18.96
print (z)
print(type(z))
```

```
-18.96
<class 'float'>
```

[10]:

```
a = 2e5
print (a)
print(type(a))
```

```
200000.0
<class 'float'>
```

1.0.3 Declaring and assigning “Complex Numbers” in Python

[11]:

```
x = -5j      #a+bj = a is real part , b is imaginary
print(x)
print(type(x))
```

```
(-0-5j)
<class 'complex'>
```

[12]:

```
y = 2 + 4j
print(y)
print(type(y))
```

```
(2+4j)
<class 'complex'>
```

[13]:

```
z = 22j
print(z)
print(type(z))
```

```
22j
<class 'complex'>
```

Inbuilt Functions Related to Numbers in Python

[14]:

```
print(abs(-9))
```

```
9
```

[15]:

```
print(round(5.683359,2))
```

```
5.68
```

[16]:

```
print(pow(5,3))
```

```
125
```

[17]:

```
print(min(11,20,50,2,1,5))
```

```
1
```

[18]:

```
print(max(11,20,50,2,1,5))
```

```
50
```

Modules Related To Numbers : Random ; Fraction ; Decimal ; Math

[19]:

```
import math
```

[20]:

```
print(math.sqrt(9))
```

```
3.0
```

[21]:

```
print(math.exp(10))
```

```
22026.465794806718
```

[22]:

```
print(math.log10(1000))
```

```
3.0
```

[23]:

```
print(math.factorial(6))
```

```
720
```

[24]:

```
#Constants
print(round(math.pi,2))
```

```
3.14
```

floor( ) and ceil( ) function These two methods are part of python math module which helps in getting the nearest integer values of a fractional number.

floor( ) :

It accepts a number with decimal as parameter and returns the integer which is smaller than the number itself.

[25]:

```
x,y,z = 21 , -23.6 , 14.2
```

[26]:

```
print("The value of ",x, "on applying floor() function is:", math.floor(x))
print("The value of ",y, "on applying floor() function is:", math.floor(y))
print("The value of ",z, "on applying floor() function is:", math.floor(z))
```

```
The value of  21 on applying floor() function is: 21
The value of  -23.6 on applying floor() function is: -24
The value of  14.2 on applying floor() function is: 14
```

ceil( ) :

It accepts a number with decimal as parameter and returns the integer which is greater than the number itself.

[27]:

```
print("The value of ",x, "on applying ceil() function is:", math.ceil(x))
print("The value of ",y, "on applying ceil() function is:", math.ceil(y))
print("The value of ",z, "on applying ceil() function is:", math.ceil(z))
```

```
The value of  21 on applying ceil() function is: 21
The value of  -23.6 on applying ceil() function is: -23
The value of  14.2 on applying ceil() function is: 15
```