In [1]:
```
print("Hello World")
```
```
Hello World
```

In [1]:
```
#Addition
print(5+3)
```
```
8
```

In [2]:
```
#Subtraction
print(5-3)
```
```
2
```

In [3]:
```
#Multiplication
print(5*3)
```
```
15
```

In [4]:
```
#Division
print(5/3)
```
```
1.6666666666666667
```

In [6]:
```
#Modulo
print(5%3)
```
```
2
```

In [7]:
```
#integer Division operator
print(5//3)
```
```
1
```

In [8]:
```
#Composite Expression
print (3 + 2 + 1 - 5 + 4 % 2 - 1 / 4 + 6)
```
```
6.75
```

In [9]:
```
#Brackets  two operands called binary operators
(5 + 3) * (3 - 4)
```
Out[9]:
```
-8
```

In [10]:
```
# Negation
- (5 + 3)
```
Out[10]:
```
-8
```

In [16]:
```
#Floating
5.0*3
```
Out[16]:
```
15.0
```

In [17]:
```
#Floating Point Exponential
5.0 ** 3
```
Out[17]:
```
125.0
```

In [18]:
```
#Floating Point Exponential
36 ** 0.5
```
Out[18]:
```
6.0
```

In [21]:
```
# To Provide the Python Type
type(5)
```
Out[21]:
```
int
```

In [22]:
```
#type is inbuilt function
type(5.0)
```

Out[22]: float

In [23]:
```
type(5.0**3)
```

Out[23]: float

In [24]:
```
float(5)
```

Out[24]: 5.0

In [28]:
```
a=float(5)
print(a)
print(type(float(5)))
```

```
5.0
<class 'float'>
```

In [29]:
```
int(5.9)
```

Out[29]: 5

In [30]:
```
type(int(5.9))
```

Out[30]: int

In [31]:
```
round(5.9)
```

Out[31]: 6

In [32]:
```
round(5.98765, 2)
```

Out[32]: 5.99

In [33]:
```
#useful function is abs, which takes one numerical argument and returns its absolute value.
abs(-5)
```

Out[33]: 5

In [35]:
```
print(abs(5))
print(abs(5.0))
```

```
5
5.0
```

In [36]:
```
#Scientfic Notation
5e1
```

Out[36]: 50.0

In [39]:
```
5e-1
```

Out[39]: 0.5

In [41]:
```
5e2
```

Out[41]: 500.0

In [1]:
```
# Variable to hold value
month_name='January'
print(month_name)
```

```
January
```

In [2]:
```
month_name[0]
```

Out[2]: 'J'

In [3]:
```
month_name[0]='F'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-3-7091bab75be1> in <module>
----> 1 month_name[0]='F'

TypeError: 'str' object does not support item assignment
```

In [4]:
```
print('January',2000)
```

```
January 2000
```

In [6]:
```
# Concatenating two strings
print('January',',First month of Year')
```

```
January ,First month of Year
```

In [11]:
```
# Concatenating a variable and string
stock_name='Aditya Birala'
print(stock_name + " is the stock name.")
#String Concatenation will be integer,Literals,Strings and Variables
```

```
Aditya Birala is the stock name.
```

In [47]:
```
#f-strings or formatted string literal.
# Use Variable in curly brackets{}
print(f'The Stock Name',stock_name)
print(f'{stock_name} is the stock name invested by Dinesh Sekaran')
```

```
The Stock Name Aditya Birala
Aditya Birala is the stock name invested by Dinesh Sekaran
```

In [51]:
```
# %-formatting strings
price=38.1
print("%s is currently trading at the price %.2f" %(stock_name,price))

# %s is used for specifying a string literal and %f is used to specify float literal.
#We use %.2f to limit two digits after the decimal point.
```

```
Aditya Birala is currently trading at the price 38.10
```

In [59]:
```
#format() function used for printing and constructing string for output
stock_picker='Kotak'
price=50.25
print('i am intersted in {x},which is trading at {y}'.format(x=stock_picker,y=price))
print('i am intersted in {},which is trading at {}'.format(stock_picker,price))
print('i am intersted in {0},which is trading at {1}'.format(stock_picker,price))
print('The stock trading at {1},which is {0}'.format(stock_picker,price))
```

```
i am intersted in Kotak,which is trading at 50.25
i am intersted in Kotak,which is trading at 50.25
i am intersted in Kotak,which is trading at 50.25
The stock trading at 50.25,which is Kotak
```

In [64]:
```
# Escape Sequence
print("That's  Pretty easy task")
print('That\'s  Pretty easy task')    #\ (backslash character
# character to indicate that it is a part of the string
```

```
That's  Pretty easy task
That's  Pretty easy task
```

In [67]:
```
# \n To Break into multiple lines
print('This is Pretty Easy Task.\n Yes, it\'s easy compartively to other task')
```

```
This is Pretty Easy Task.
 Yes, it's easy compartively to other task
```

In [69]:
```
# \t tab escape sequence
print('Dinesh\tSekaran')
```

```
Dinesh  Sekaran
```

In [74]:
```
# single \ at the end of the line, it indicates that the string is continued in the next line
print('AAPL is the ticker for Apple Inc.\
...: It is a technology company.')

#More Escape Sequence Which is found.
```

```
AAPL is the ticker for Apple Inc....: It is a technology company.
```

In [ ]:
```
#Indentation
Whitespaces are important in Python. Whitespace at the start of a line is called indentation.
We either use four spaces or tab space for indentation.

```