# Day Objectives

## Introduction To Juypter Notebook

**BOLD**

*ITALIAN*

*TEXT


1.ordered list 1

2.ordered list 2

3.ordered list 3

# Python Basics

## Arthematic operations

In [1]:
```
##1

'''divija
divija1
divijag2''' #Using For Multi-line Comments
```

Out[1]:
```
'divija\ndivija1\ndivijag2'
```

In [2]:
```
##2

##variable assignment
n=369
```

Out[2]:
```
369
```

In [3]:
```
##3

n1=n
```

Out[3]:
```
369
```

In [8]:
```
##4

n2=n1
```

Out[8]:
```
369
```

In [10]:
```
##5

##multi variables holding single value
n1=n2=n3=n
```

Out[10]:
```
369
```

In [15]:
```
##6

n=12
n
n1=10
```

Out[15]:
```
10
```

In [14]:
```
##7
```

Out[14]:
```
12
```

In [16]:
```
##8

n=12
print(n)
n1=10
```

```
12
```

Out[16]:
```
10
```

In [17]:
```
##9

n=12
n1=10
n1
```

```
12
```

In [18]:
```
##10

n=1
n=2
n=3
print(n)
n1=2
```

```
3
2
```

In [34]:
```
##11

#type conversion: converting one datatype into another
x=10
print(x)#default type int
type(x)
y=str(x)
print(type(y))
print(float(x))
```

```
10
<class 'str'>
10.0
```

In [35]:
```
##12

#type conversion: converting one datatype into another
x=12
type(x)
float(x)
```

Out[35]:
```
int
```

In [36]:
```
##13

#type conversion: converting one datatype into another
x=12
type(x)
float(x)
```

Out[36]:
```
12.0
```

In [37]:
```
#basic operations
##add
x=1
y=2
print
print(x+y) ##directly also we can get output by giving values, but the correct wa
##sub
k=-3
j=-4
print(k-j)
##mult
c=((x+y)*(k+j))
##division
print(c/2)#single modular division operator gives float value
print(c//2)#double single modular division operator gives int value
##modular
print(c%2)
c
```

```
3
1
-10.5
-11
1
```

Out[37]:
```
-21
```

In [38]:
```
print(2+3)
```

```
5
```

```
4946774523215560916
```

In [39]:
```
a='k'
```

Out[39]:
```
str
```

In [40]:
```
a='k'
int(str(a))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-40-2ece08e9b1d0> in <module>()
      1 a='k'
----> 2 int(str(a))
      3 type(a)

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

In [41]:
```
u="1"
print(u)
y=int(u)
```

```
1
1
```

# Indentation In Python

## Indentation is nothing but space or tab

### Conditional statements

#### if,if else,if elif else,nested if

In [43]:
```
## If Condition
## Write a program to know how to use if condition
n=3
if(n%2==0):
    n=n+1
    print("n is even no.")
else:
```

```
n is not even no.
```

In [44]:
```
a=int(input())
b=int(input())
if a > b:
    print("a is big")
else:
```

```
10
30
b is big
```

## String Slicing

In [45]:
```
s="python"
```

Out[45]: 'python'

In [46]:

Out[46]: 6

In [47]:

Out[47]: 'n'

In [48]:

```
IndexError                                Traceback (most recent call last)
<ipython-input-48-665bb6993e1f> in <module>()
----> 1 s[11]

IndexError: string index out of range
```

In [49]:
```
print(s[2])
```

```
t
```

Out[49]: 'p'

In [56]:

Out[56]: 'n'

In [54]:
```
s1="divija sindhu"
```

Out[54]: 'divija sindhu'

In [55]:

Out[55]: 'u'

In [59]:
```
k="near"
print(k[-1])
print(k[-2])
```

```
r
a
a
```

In [60]:
```
s="python is powerful language"
print(s[0:6])
print(s[50:])
```

```
python
```

In [61]:

In [65]:

```
languag
```

In [66]:

```
language
```

In [67]:

```
yhni oefllnug
```

In [69]:

```
pto spwru agae
```

In [71]:

```
print(s[-1::-1])#method 1 to reverse a no.
```

```
egaugnal lufrewop si nohtyp
egaugnal lufrewop si nohtyp
```

In [76]:

```
print(s[-1::-2])#method 1 to get even no.s from reverse order
```

```
eaga urwps otp
eaga urwps otp
```

In [78]:

```
pto spwru agae
```

# String Methods

In [80]:

Out[80]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rmod__',
 '__rmul__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'capitalize',
 'casefold',
 'center',
 'count',
 'encode',
 'endswith',
 'expandtabs',
 'find',
 'format',
 'format_map',
 'index',
 'isalnum',
 'isalpha',
 'isdecimal',
 'isdigit',
 'isidentifier',
 'islower',
 'isnumeric',
 'isprintable',
 'isspace',
 'istitle',
 'isupper',
```

```
 'join',
 'ljust',
 'lower',
 'lstrip',
 'maketrans',
 'partition',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [88]:
```
a="PYThon welcome"
print(a)
a=a.capitalize()
print(a)
print(a.title())##for methods we need to put paranthesis at ending
print(a.upper())
print(a)#no change in a value,value which is in s be printed
```

```
PYThon welcome
Python welcome
Python Welcome
PYTHON WELCOME
Python welcome
python welcome
```

In [111]:
```
b="A"
print(b.islower())
print(b.isupper())
s="Divija123"
print(s.isalpha())
print(s.isnumeric())
print(s.isalnum())
print(s.isdigit())
g="1233 "
print(g.isnumeric())
```

```
False
True
False
False
True
False
False
False
```

In [99]:
```
v="p.sindhu"
```

```
p.sindhu
```

In [106]:
```
a=input()
if(a==a[::-1]):
    print("palindrome")
else:
```

```
lol
palindrome
```

In [112]:
```
a=input()
if(a==a[::-1]):
    print("palindrome")
else:
```

```
divi
not a palindrome
```

In [119]:
```
a="   python   "
print(a.lstrip())
print(a.rstrip())
b="8888python8888"
print(b.lstrip("8"))
print(b.rstrip("8"))
t="wwwi"
```

```
python
   python
python8888
8888python
```

Out[119]:
```
3
```

In [10]:
```
v=input()
d=input()
if(v==d):
    print("true")
else:
```

```
uli
vic
false
```

In [11]:
```
v=input()
d=input()
if(v==d):
    print("true")
else:
```

```
lol
lol
true
```

In [7]:
```
# Function
## functions in python defined by a keyword "def"
## write a program for finding leap year
def isLeapyear(year):
    if((year%400==0)or(year%4==0)and(year%100!=0)):
        print("leapyear")
    else:
        print("not leapyear")
y=int(input("enter year:"))
```

```
enter year:2019
not leapyear
```

In [9]:
```
def isLeapyear(year):
    if((year%400==0)or(year%4==0)and(year%100!=0)):
        print("leapyear")
    else:
        print("not leapyear")
y=int(input("enter year:"))
```

```
enter year:2020
leapyear
```

In [15]:
```
##divisible by 10 and not by 4
def check_divisibility(a):
    if(a%10==0 and a%4!=0):
        print("true")
    else:
        print("false")
b=int(input())
```

```
400
false
```

In [17]:
```
def check_divisibility(a):
    if(a%10==0 and a%4!=0):
        print("true")
    else:
        print("false")
b=int(input())
```

```
4
false
```

In [19]:
```
def check_divisibility(a):
    if(a%10==0 and a%4!=0):
        print("true")
    else:
        print("false")
b=int(input())
```

```
10
true
```

In [20]:
```
# Python program to find the largest number among the three input numbers

# change the values of num1, num2 and num3
# for a different result
num1 = 10
num2 = 14
num3 = 12

# uncomment following lines to take three numbers from user
#num1 = float(input("Enter first number: "))
#num2 = float(input("Enter second number: "))
#num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):
   largest = num1
elif (num2 >= num1) and (num2 >= num3):
   largest = num2
else:
   largest = num3
```

```
The largest number between 10 , 14 and 12 is 14
```

In [30]:
```
##biggest of a,b,c def maximum(a, b, c):
  # Python program to find the largest
# number among the three numbers

def maximum(a, b, c):

    if (a >= b) and (a >= c):
        largest = a

    elif (b >= c):
        largest = b
    else:
        largest = c

    return largest


# Driven code
a = int(input())
b = int(input())
c = int(input())
```

```
7
8
5
biggest no. is  8
```

In [31]:
```
##for(initialisation,condition,increment):
def nat_num(n):
    for i in range(1,n+1,3):
        print(i)
```

```
1
4
7
10
```

In [43]:
```
to print even no.s from 1 to 20
def even_num(n):
    for i in range(1,n+1):
        if(i%2==0):
            print(i)
```

```
2
4
6
8
10
12
14
16
18
20
```

In [52]:
```
##to print odd numbers by skipp every 2 no.s odd in between the numbers till 40
def odd_num(n):
    for i in range(1,n+1,3):
        if(i%2!=0):
            print(i)
```

```
1
7
13
19
25
31
37
```

In [56]:
```
#write a program to print factors of n
def factor(n):
    for i in range(1,7):
        if(n%i==0):
            print(i)
```

```
1
2
3
6
```

In [57]:
```
##write program to find factorial of number
def factorial(num):
    """This is a recursive function that calls
   itself to find the factorial of given number"""
    if num == 1:
        return num
    else:
        return num * factorial(num - 1)


# We will find the factorial of this number
num = int(input("Enter a Number: "))

# if input number is negative then return an error message
# elif the input number is 0 then display 1 as output
# else calculate the factorial by calling the user defined function
if num < 0:
    print("Factorial cannot be found for negative numbers")
elif num == 0:
    print("Factorial of 0 is 1")
else:
```

```
Enter a Number: 5
Factorial of 5 is:  120
```

In [58]:
```
def factorial(num):
    if num == 1:
        return num
    else:
        return num * factorial(num - 1)
num = int(input("Enter a Number: "))
if num < 0:
    print("cannot be found")
elif num == 0:
    print("Factorial of 0 is 1")
else:
```

```
Enter a Number: 5
Factorial 5 is  120
```

In [ ]:
```
```