



PYTHON

TAJENDAR ARORA

In and not in operator

- ▶ **in** Evaluates to true if it finds a variable in the specified sequence and false otherwise.
- ▶ **not in** Evaluates to true if it does not finds a variable in the specified sequence and false otherwise.

In and not in operator

```
a = 10
```

```
b = 20
```

```
list = [1, 2, 3, 4, 5 ];
```

```
if ( a in list ):
```

```
    print ("Line 1 - a is available in the given list" )
```

```
else:
```

```
    print ("Line 1 - a is not available in the given list" )
```

```
if ( b not in list ):
```

```
    print ("Line 2 - b is not available in the given list" )
```

```
else:
```

```
    print ("Line 2 - b is available in the given list" )
```



FUNCTION OF LIST

- ▶ `len(list)`
- ▶ `len([1, 2, 3])`
- ▶ `[1, 2, 3] + [4, 5, 6]` `[1, 2, 3, 4, 5, 6]`
- ▶ `['Hi!'] * 4` `['Hi!', 'Hi!', 'Hi!', 'Hi!']`
- ▶ `max(list)` Returns item from the list with max value.
- ▶ `min(list)` Returns item from the list with min value.
- ▶ `list.append(obj)` Appends object obj to list
- ▶ `list.count(obj)` Returns count of how many times obj occurs in list



FUNCTION OF LIST

- ▶ `list.index(obj)` Returns the lowest index in list that obj appears
 - ▶ `list.insert(index, obj)` Inserts object obj into list at offset index
 - ▶ `list.pop(obj=list[-1])` Removes and returns last object or obj from list
 - ▶ `list.remove(obj)` Removes object obj from list
 - ▶ `list.reverse()` Reverses objects of list in place
-
- ▶ Tuple have same function as same

Math function IN PYTHON

Main Function for Maths are

`abs(x)` The absolute value of x: the (positive) distance between x and zero.

`cmp(x, y)` -1 if $x < y$, 0 if $x == y$, or 1 if $x > y$

`max(x1, x2,...)` The largest of its arguments: the value closest to positive infinity

`min(x1, x2,...)` The smallest of its arguments: the value closest to negative infinity

`pow(x, y)` The value of $x^{**}y$.

STRING IN PYTHON

String is similar as we used in JAVA, C, C++ etc. We have many functions used with string in the python.

To access substrings, use the square brackets for slicing along with the index or indices to obtain your substring. For example:

```
var1 = 'Hello World!'
```

```
var2 = "Python Programming"
```

```
print ("var1[0]: ", var1[0] )
```

```
print ("var2[1:5]: ", var2[1:5] )
```

STRING IN PYTHON

We have many functions used with string in the python.

capitalize()	center()	count()
startswith()	endswith()	expandtabs(), str.expandtabs(16);
isalpha()	isdigit	max(str)
min(str),len(str)	find(str, beg=0 end=len(string))	replace()
title()	swapcase()	Islower(),isupper()
Lower(),upper()	str.center(width[, fillchar])	str.index(str, beg=0 end=len(string))

USE OF STRING FUNCTIONS

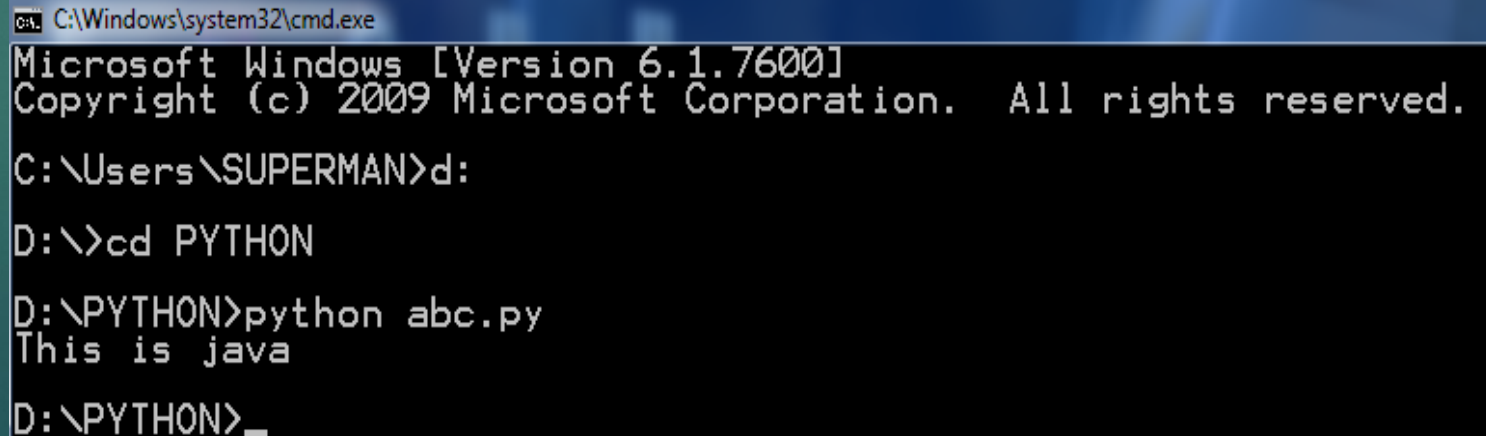
capitalize()

The capitalize() function is used to change the first letter of the string from small to capital letter.

```
str='this is java'  
print (str.capitalize())
```



OUTPUT



```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\SUPERMAN>d:  
  
D:\>cd PYTHON  
  
D:\PYTHON>python abc.py  
This is java  
  
D:\PYTHON>_
```

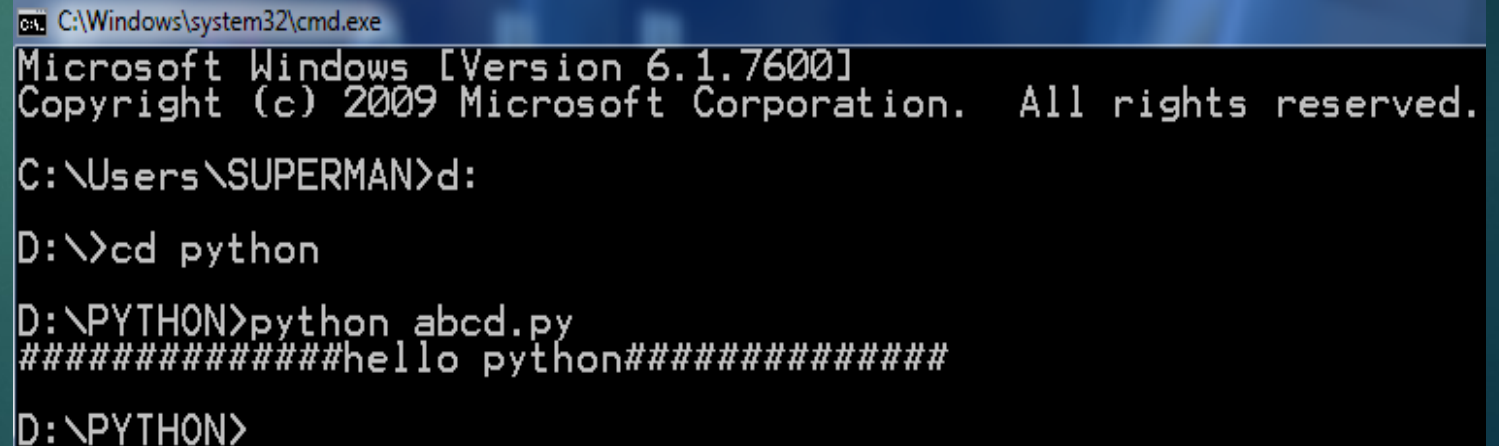
center()

This function is used to placed the text into the center of the line.

```
str1='hello python'  
print (str1.center(40,'#'))
```



OUTPUT



```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 6.1.7600]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\SUPERMAN>d:  
  
D:\>cd python  
  
D:\PYTHON>python abcd.py  
#####hello python#####  
  
D:\PYTHON>
```

count()

The count() function is used to count the number of characters are in string.

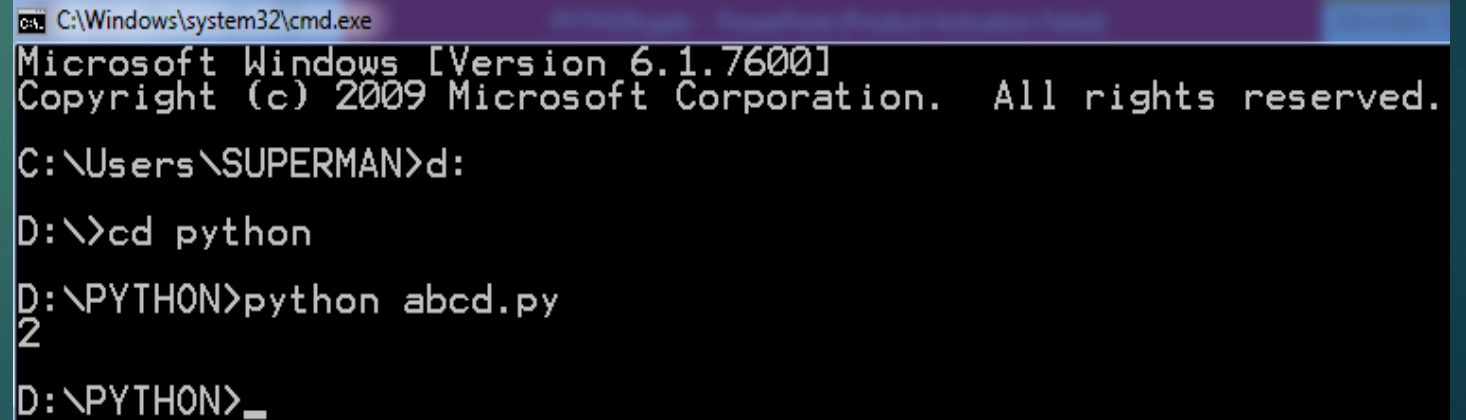
```
str2='I AM RAHUL'
```

```
ch="A"
```

```
print (str2.count(ch))
```



OUTPUT



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\SUPERMAN>d:
D:\>cd python
D:\PYTHON>python abcd.py
2
D:\PYTHON>_
```

DATE AND TIME

Module Time is use to get time information

Time

time(): This function is used to find the time in seconds. Example:

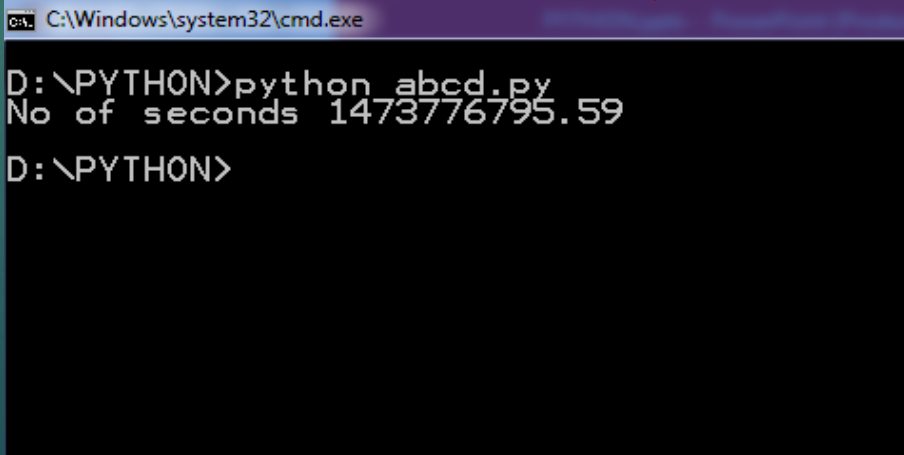
```
import time;
```

```
ticks=time.time();
```

```
print ("No of seconds",ticks )
```



OUTPUT



```
C:\Windows\system32\cmd.exe  
D:\PYTHON>python abcd.py  
No of seconds 1473776795.59  
D:\PYTHON>
```

DATE AND TIME

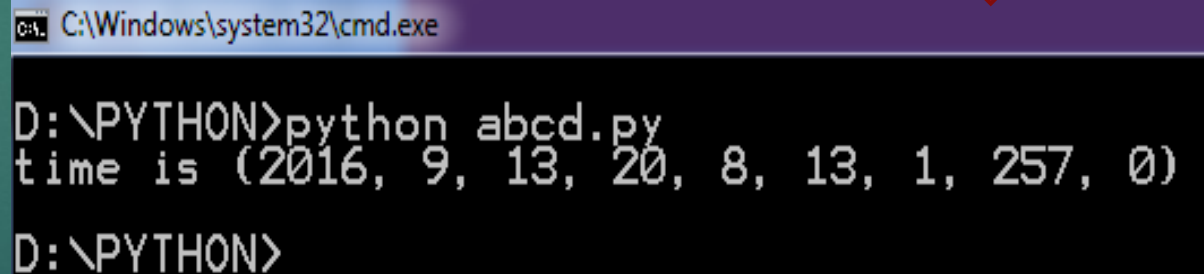
localtime(): This function is used to show the local time.

Example:

```
import time;  
t=time.localtime(time.time())  
print ("time is",t_)
```



OUTPUT



```
C:\Windows\system32\cmd.exe  
  
D:\PYTHON>python abcd.py  
time is (2016, 9, 13, 20, 8, 13, 1, 257, 0)  
  
D:\PYTHON>
```

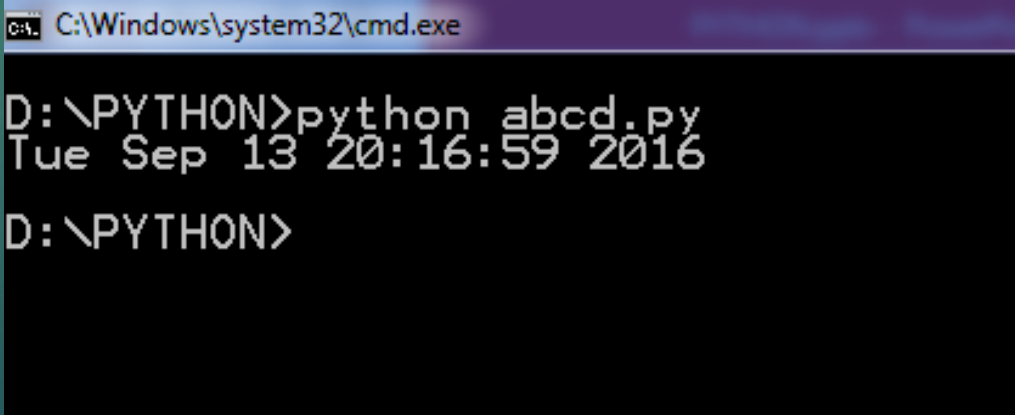

DATE AND TIME

asctime(): This function is used to show the formatted time.

```
import time;  
ts=time.asctime(time.localtime(time.time()))  
print (ts)
```



OUTPUT



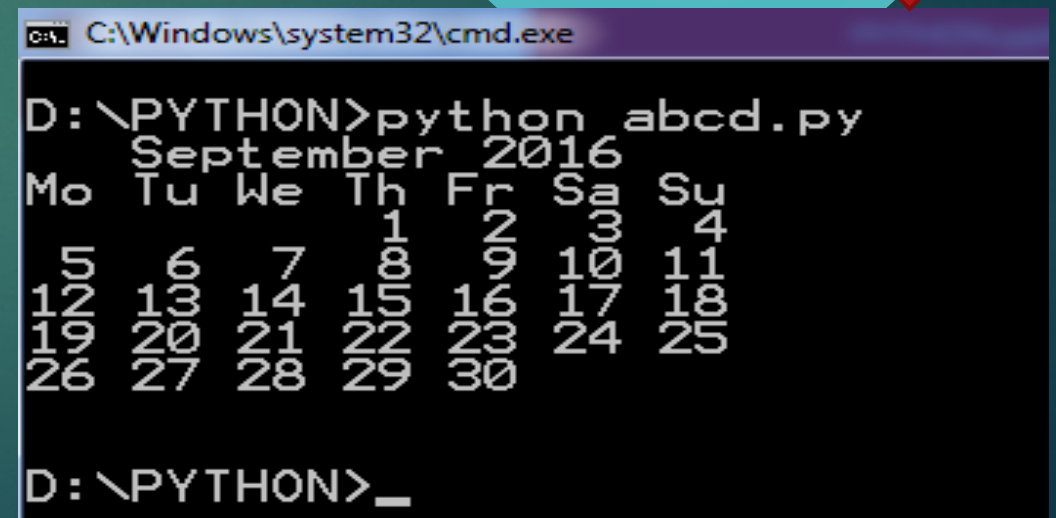
```
C:\Windows\system32\cmd.exe  
D:\PYTHON>python abcd.py  
Tue Sep 13 20:16:59 2016  
D:\PYTHON>
```

Calendar

month : This function is used to show the calendar by importing calendar package.

```
import calendar;  
cal=calendar.month(2016,9)  
print (cal)
```

OUTPUT



```
C:\Windows\system32\cmd.exe  
D:\PYTHON>python abcd.py  
September 2016  
Mo Tu We Th Fr Sa Su  
    1  2  3  4  
 5  6  7  8  9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30  
D:\PYTHON>_
```

FUNCTION OF DICTIONARY

`len(dict)`

Gives the total length of the dictionary. This would be equal to the number of items in the dictionary.

`str(dict)`

Produces a printable string representation of a dictionary

`dict.clear()`

Removes all elements of dictionary *dict*

`dict.get(key, default=None)`

For key *key*, returns value or default if key not in dictionary

`dict.keys()`

Returns list of dictionary *dict*'s keys

`dict.values()`

Returns list of dictionary *dict*'s values