

List Comprehension

List comprehensions provides a concise way to create lists.

It consists of brackets containing an expression followed by a for clause, then zero or more for or if clauses. The expression can be anything, meaning you can put in all kinds of objects in lists.

```
In [2]: lst1=[]  
def lst_square(lst):  
    for i in lst:  
        lst1.append(i*i)  
    return lst1
```

```
In [3]: lst_square([1,2,3,4,5,6,7])
```

```
Out[3]: [1, 4, 9, 16, 25, 36, 49]
```

```
In [4]: ## list comprehension
```

```
In [5]: lst=([1,2,3,4,5,6,7])
```

```
In [6]: [i*i for i in lst]
```

```
Out[6]: [1, 4, 9, 16, 25, 36, 49]
```

```
In [7]: lst1=[i*i for i in lst]
```

```
In [8]: print(lst1)
```

```
[1, 4, 9, 16, 25, 36, 49]
```

```
In [9]: lst1=[i*i for i in lst if i%2==0] ## for odd ! and for even !
```

```
In [10]: print(lst1)
```

```
[4, 16, 36]
```

```
In [11]: lst1=[i*i for i in lst if i%2!=0]
```

```
In [12]: print(lst1)
```

```
[1, 9, 25, 49]
```

In []:

In []:

String Formatting in Python

In [13]:

```
print('hello everyone')
```

hello everyone

In [16]:

```
def greeting(name):  
    return "hello {}.Welcome to the community".format(name)
```

In [18]:

```
greeting("Desh Deepak Verma")
```

Out[18]: 'hello Desh Deepak Verma.Welcome to the community'

In [19]:

```
def welcome_email(name,age):  
    return "welcome {}.your age is {}".format(name,age)
```

In [20]:

```
welcome_email('desh deepak verma',29)
```

Out[20]: 'welcome desh deepak verma.your age is 29'