PYTHON

From Simple to Complex

With

Examples

AYESHA NOREEN

Bachelor's in Software Engineering,

Master's in Computer Science

from COMSATS University, Islamabad

NOTE!!!

In these notes Screenshots of practice examples and coding are added. The code files are also available in code folder that contain .ipynb files that are created on Jupyter notebook.

Chapter9
List comprehension

List comprehension is used for making lengthy codes smaller. e.g.

If we want to store square of numbers from 1 to 10 than,

- By simple method

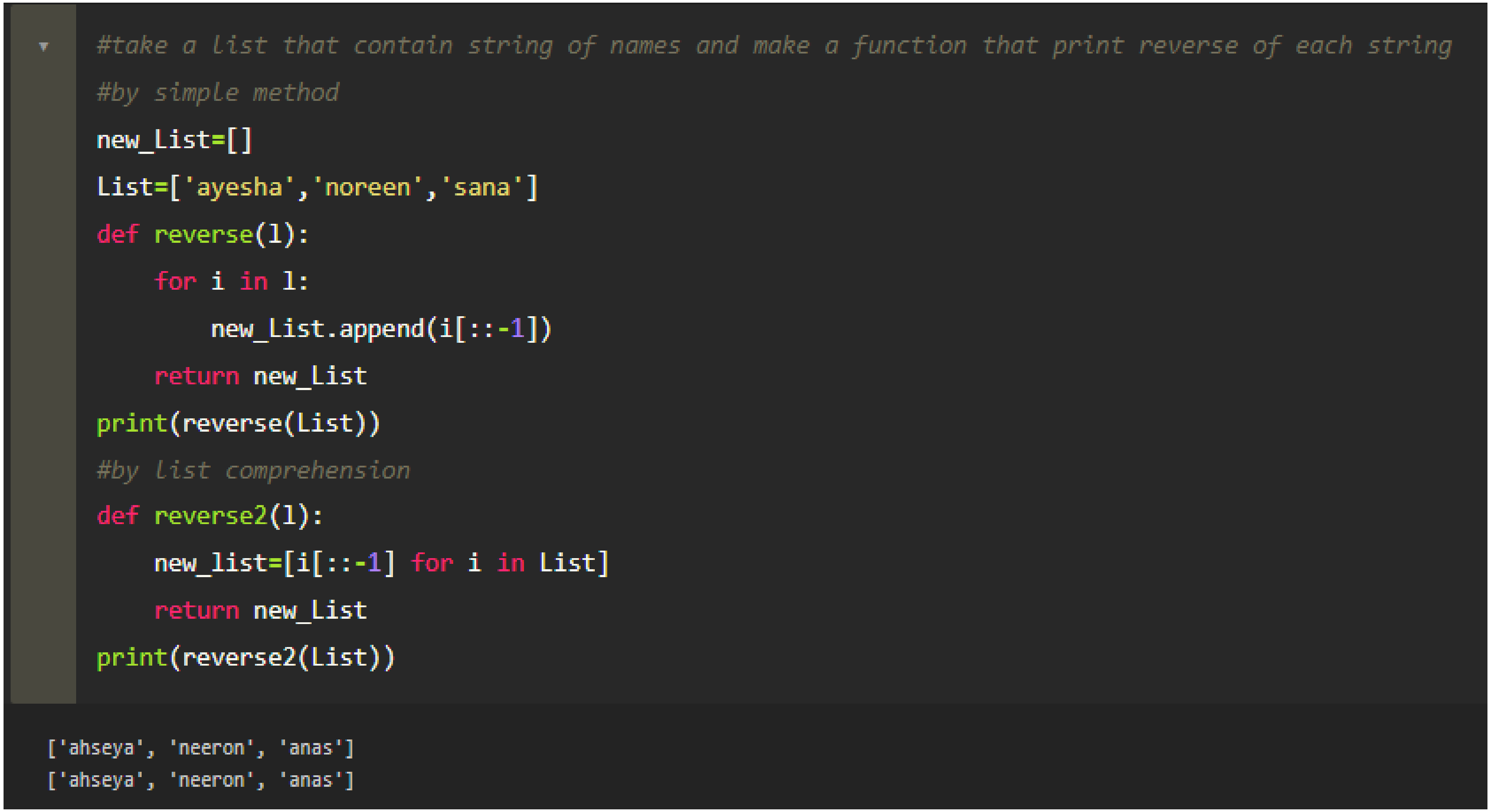
```
square=[]
for i in range(1,11):
    square.append(i**2)
print(square)
```

- By list comprehension

```
square2=[i**2 for i in range(1,11)]
print(square2)
```

- ## TODO Task

Take a list that contain string of names and make a function that print reverse of each string.

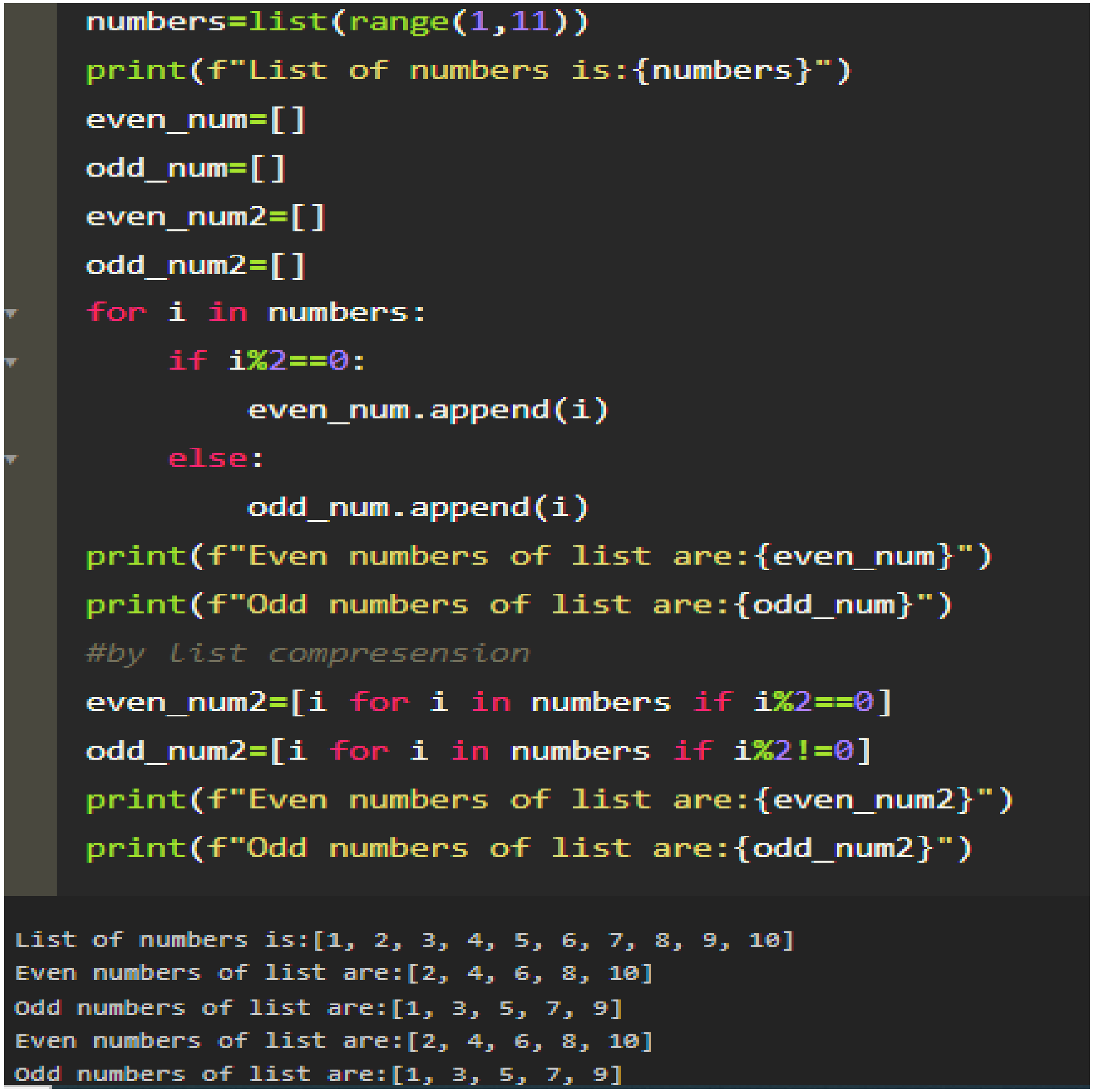


```
#take a list that contain string of names and make a function that print reverse of each string
#by simple method
new_List=[]
List=['ayesha','noreen','sana']
def reverse(l):
    for i in l:
        new_List.append(i[::-1])
    return new_List
print(reverse(List))
#by list comprehension
def reverse2(l):
    new_list=[i[::-1] for i in List]
    return new_List
print(reverse2(List))
```

```
['ahseya', 'neeron', 'anas']
['ahseya', 'neeron', 'anas']
```

• If statement in list comprehension

If we want to separate even and odd numbers of a list than by simple method

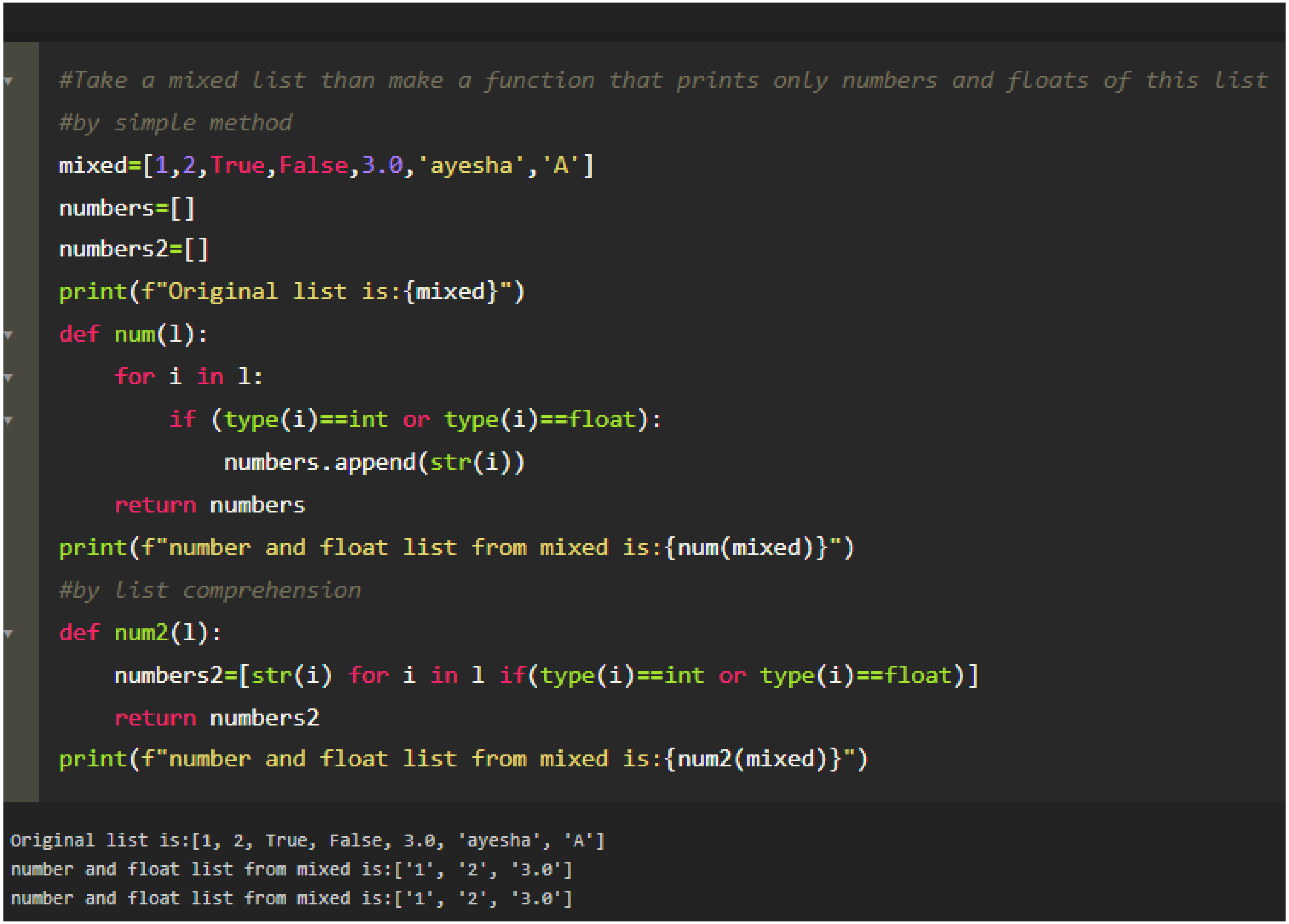


```python
numbers=list(range(1,11))
print(f"List of numbers is:{numbers}")
even_num=[]
odd_num=[]
even_num2=[]
odd_num2=[]
for i in numbers:
    if i%2==0:
        even_num.append(i)
    else:
        odd_num.append(i)
print(f"Even numbers of list are:{even_num}")
print(f"Odd numbers of list are:{odd_num}")
#by list compresension
even_num2=[i for i in numbers if i%2==0]
odd_num2=[i for i in numbers if i%2!=0]
print(f"Even numbers of list are:{even_num2}")
print(f"Odd numbers of list are:{odd_num2}")
```

```
List of numbers is:[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Even numbers of list are:[2, 4, 6, 8, 10]
Odd numbers of list are:[1, 3, 5, 7, 9]
Even numbers of list are:[2, 4, 6, 8, 10]
Odd numbers of list are:[1, 3, 5, 7, 9]
```

- # TODO Task

Take a mixed list than make a function that prints only numbers and floats of this list

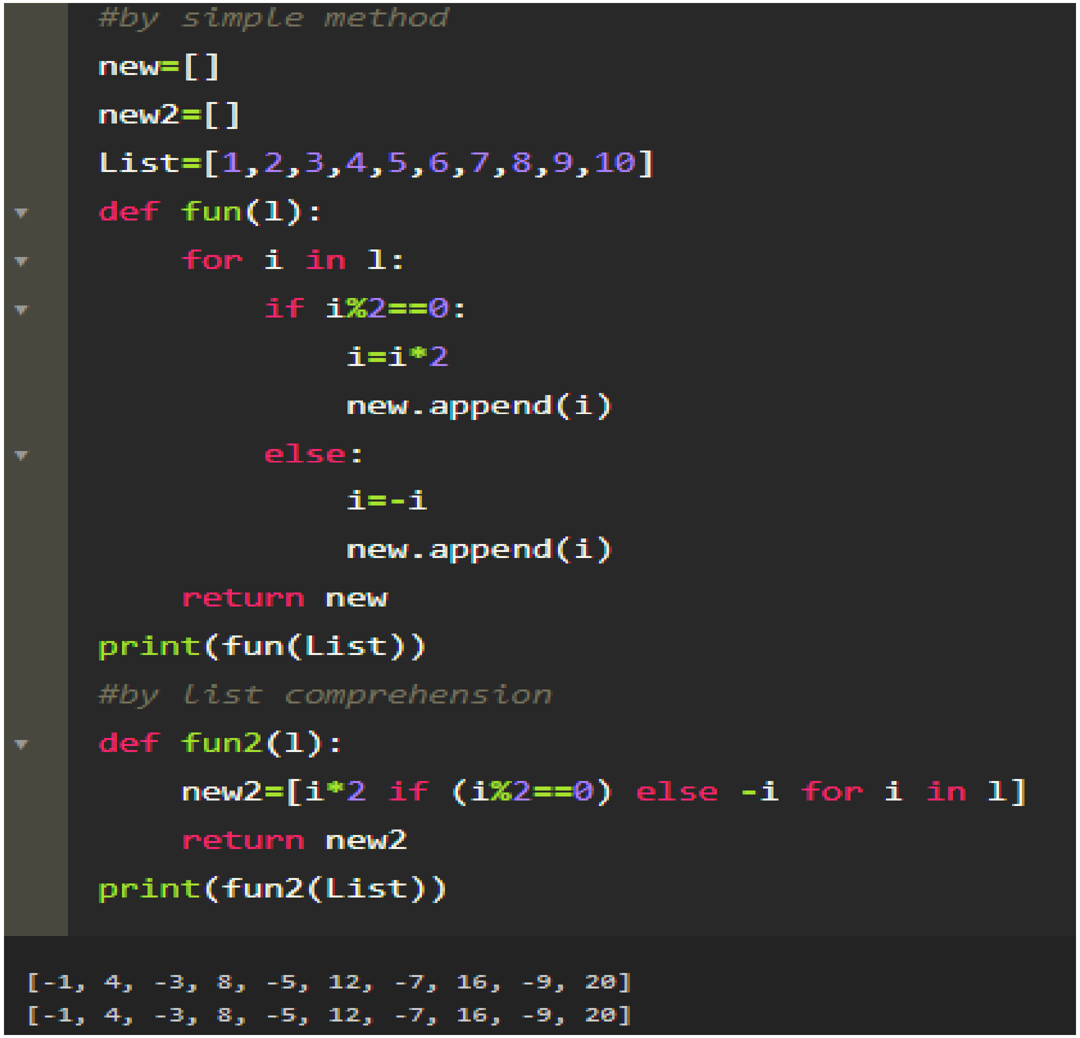


```python
#Take a mixed list than make a function that prints only numbers and floats of this list
#by simple method
mixed=[1,2,True,False,3.0,'ayesha','A']
numbers=[]
numbers2=[]
print(f"Original list is:{mixed}")
def num(l):
    for i in l:
        if (type(i)==int or type(i)==float):
            numbers.append(str(i))
    return numbers
print(f"number and float list from mixed is:{num(mixed)}")
#by list comprehension
def num2(l):
    numbers2=[str(i) for i in l if(type(i)==int or type(i)==float)]
    return numbers2
print(f"number and float list from mixed is:{num2(mixed)}")
```

```
Original list is:[1, 2, True, False, 3.0, 'ayesha', 'A']
number and float list from mixed is:['1', '2', '3.0']
number and float list from mixed is:['1', '2', '3.0']
```

• If/else statement in list comprehension

Take a number list and than define a function that print -ve of number if number is odd if even than multiply it by 2

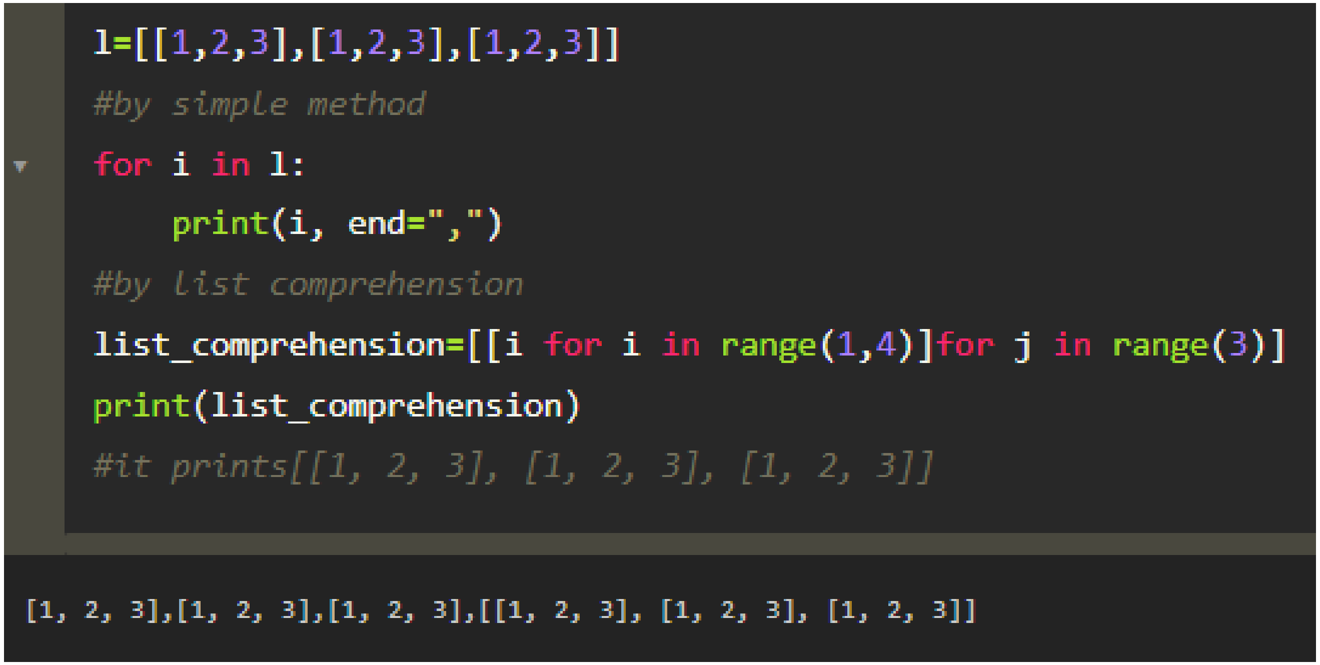


```
#by simple method
new=[]
new2=[]
List=[1,2,3,4,5,6,7,8,9,10]
def fun(l):
    for i in l:
        if i%2==0:
            i=i*2
            new.append(i)
        else:
            i=-i
            new.append(i)
    return new
print(fun(List))
#by list comprehension
def fun2(l):
    new2=[i*2 if (i%2==0) else -i for i in l]
    return new2
print(fun2(List))
```

```
[-1, 4, -3, 8, -5, 12, -7, 16, -9, 20]
[-1, 4, -3, 8, -5, 12, -7, 16, -9, 20]
```

• Nested list comprehension

If we have list inside list than how to print by list comprehension.



```
l=[[1,2,3],[1,2,3],[1,2,3]]
#by simple method
for i in l:
    print(i, end=",")
#by list comprehension
list_comprehension=[[i for i in range(1,4)]for j in range(3)]
print(list_comprehension)
#it prints[[1, 2, 3], [1, 2, 3], [1, 2, 3]]
```

```
[1, 2, 3],[1, 2, 3],[1, 2, 3],[[1, 2, 3], [1, 2, 3], [1, 2, 3]]
```