List Comprehensions

علاوه بر عملیات توالی و روش های لیست ، پایتون عملیات پیشرفته تری را با نام List Comprehensions شامل می شود.

List Comprehensions به ما اجازه می دهد لیست ها را با استفاده از یک نماد متفاوت بسازیم. شما می توانید آن را به عنوان یک حلقه `for` در یک خط در داخل براکت ها در نظر بگیرید. برای یک مثال ساده:

مثال 1

In [1]:

```
cubic_list = []

for i in range(1,11):
    result = i ** 3
    cubic_list.append(result)

cubic_list
```

Out[1]:

```
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
```

In [2]:

```

l = [x for x in range(1, 11)]

l
```

Out[2]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [3]:

```
l = [x**3 for x in range(1, 11)]

l
```

Out[3]:

```
[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]
```

In [4]:

```
# Grab every letter in string

lst = [x for x in 'Hello world!']
```

In [5]:

```
# Check
lst
```

Out[5]:

```
['H', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd', '!']
```

این ایده اصلی یک List Comprehensions است. اگر با نمادهای ریاضی آشنا باشید ، این فرمت باید برای شما آشنا باشد ، به عنوان مثال: `{x^2 : x در 0،1،2 ... 10}`

بیایید چند مثال دیگر از فهم لیست در پایتون را ببینیم:

مثال 2

In [6]:

```
# Square numbers in range and turn into list
lst = [x**2 for x in range(0,11)]
```

In [7]:

```
lst
```

Out[7]:

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

مثال 3

حالا اگر نیاز به دستور `if` داشتیم چگونه عمل کنیم؟:

In [8]:

```
# Check for even numbers in a range
lst = [x**2 for x in range(0,11) if x % 2 == 0]
```

In [9]:

```
lst
```

Out[9]:

```
[0, 4, 16, 36, 64, 100]
```

مثال 4

همچنین می توانیم محاسبات پیچیده تری نیز داشته باشیم

In [10]:

```
# Convert Celsius to Fahrenheit
celsius = [0, 10, 20.1, 34.6]

fahrneheit = [ (((9 / 5) * temp) + 32)  for temp in celsius]

fahrneheit
```

Out[10]:

```
[32.0, 50.0, 68.18, 94.28]
```

مثال 5

علاوه بر مثال های بالا، ما می توانیم عملیاتی تودرتو از List Comprehensions را داشته باشیم:

In [11]:

```
lst = [x ** 2 for x in [i ** 2 for i in range(11)]]

lst
```

Out[11]:

```
[0, 1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000]
```