

filter()

تابع `filter(function, list)` راهی مناسب برای فیلتر کردن تمام عناصر یک مجموعه است، به این صورت که تابع برای آن عنصر مقدار `True` برگرداند.

تابع `filter(function, list)` به یک تابع به عنوان آرگومان اول نیاز دارد. این تابع باید یک مقدار بولین (با ارزش `True` یا `False`) برگرداند. این تابع بر روی هر عنصر قابل تکرار اعمال می‌شود. تنها اگر تابع `True` را برگرداند، عنصر مجموعه در نتیجه شامل خواهد شد.

مانند `map()` تابع `filter()` نیز یک آیتريٲور را برمی‌گرداند - به عبارتی، `filter()` نتیجه را به صورت یک مقدار در هر بار نیازمندی برمی‌گرداند. درباره آیتريٲورها و ژنراتورها در یک فصول آینده صحبت خواهیم کرد. در حال حاضر، زیرا مثال‌های ما بسیار کوچک هستند، ما `filter()` را به صورت یک لیست تبدیل می‌کنیم تا نتایج را به صورت فوری مشاهده کنیم.

بیایید چند مثال ببینیم:

In [1]:

```
1 #First Let's make a function
2 def even_check(num):
3     if num % 2 == 0:
4         return True
```

حالا بیایید یک لیست از اعداد را فیلتر کنیم. توجه: قرار دادن تابع در `filter()` بدون هیچ پرانتزی ممکن است عجیب به نظر برسد، اما به یاد داشته باشید که توابع نیز شی‌هایی هستند.

In [3]:

```
1 lst = range(20)
2
3 list(filter(even_check, lst))
```

Out[3]:

[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

`filter()` بیشتر با استفاده از توابع لامبدا مورد استفاده قرار می‌گیرد، زیرا معمولاً از `filter()` برای کار سریعی استفاده می‌کنیم که نیازی به نوشتن یک تابع کامل ندارد. بیایید مثال بالا را با استفاده از یک عبارت لامبدا تکرار کنیم:

In [4]:

```
1 list(filter(lambda x: x % 2 == 0, lst))
```

Out[4]:

[0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

عالی! حالا باید یک درک قوی از `filter()` و نحوه استفاده از آن در کد خود داشته باشید!

