**lambda :**

در پایتون ، یک تابع ناشناس ممکن است تابعی باشد که بدون شهرت تعریف شده باشد. هنگام تعریف توابع عادی، ما از کلمه کلیدی def در پایتون استفاده می کنیم، اما هنگام تعریف توابع ناشناس از کلمه کلیدی lambda استفاده می کنیم. لامبدا روشی سریع برای ایجاد تابع است. عبارات لامبدا باید به ساده ترین شکل ممکن و به دور از هرگونه پیچیدگی نوشته شوند

## ****# چطور از lambda استفاده کنیم؟****

ساختار نگارش کلی عبارات لامبدا به شکل زیر است:

lambda (arguments): manipulation(arguments)

ابتدا با استفاده از کلمه کلیدی lambda مشخص میکنید که قراره یک لامبدا داشته باشید. بعد از اون آرگومان هایی که قرار است عبارت لامبدا داشته باشه رو مینویسید. و در آخر کاری قراره روی آرگومان ها انجام بشه رو مینویسید. دقت کنید که عبارت lambda هر تعدادی که بخواهید میتواند آرگومان داشته باشد اما فقط یک manipulation خواهد داشت.

### ****+ یک مثال از عبارت لامبدا****

به مثال زیر دقت کنید:

add = lambda x, y: x + y

print( add(4, 6) )

در مثال بالا یک عبارت لامبدا ساختیم که دو مقدار x و y رو میگیره و اونها رو با هم جمع میکنه. و در آخر لامبدا رو داخل یک متغیر ذخیره میکنیم تا بتونیم بعدا ازش استفاده کنیم. در خط آخر هم لامبدا رو صدا زدیم و دو عدد رو بهش فرستادیم. اگه کد بالا رو اجرا کنید مقدار ۱۰ رو به شما برمیگردونه. دقت کنید که لامبدا به صورت اتوماتیک مقدار نهایی رو return میکنه.

**+ چرا از عبارت لامبدا استفاده کنیم؟**

همانطور که در قسمت قبل مشاهده کردید، پایتون با لامبدا به شکل یکسان با فانکشن‌های معمولی برخورد میکند. به نوعی، lambda شکلی جمع و جور برای نوشتن توابع ارائه می دهد که یک عبارت را باز می گرداند. با این حال، شما باید بدانید که استفاده از لامبدا چه زمانی مناسب است و چه زمانی از آنها اجتناب کنید.

یکی از مهمترین موارد استفاده رایج برای lambda، برنامه نویسی فانکشنال است زیرا پایتون از پارادایم (یا سبک) برنامه نویسی که به عنوان برنامه نویسی فانکشنال شناخته می شود پشتیبانی می کند.

لامبدا به شما این امکان را می دهد که یک تابع را به عنوان پارامتر به یک تابع دیگر (به عنوان مثال، در map، filter و غیره) ارائه دهید. در چنین مواردی، استفاده از lambda به لطف ایجاد یک تابع یک بارمصرف، مفید است.

در محیط production، هرگز نباید توابع lambda پیچیده را بنویسید، زیرا رمزگشایی برای کد نویسان که کد شما را حفظ می کنند بسیار دشوار است. اگر خودتان را در حال ساختن عبارات پیچیده تک خطی کشف کردید، بهتر است با def یک فانکشن کامل بنویسید.

## ****# استفاده از lambda با filter****

تابع filter برای فیلتر کردن برخی عناصر خاص از یک دنباله استفاده می شود. دنباله مورد استفاده در این تابع یک [آبجکت ‌iterable](https://www.mongard.ir/one_part/53/creating-iterable-objects-python/) مانند لیست ها ، مجموعه ها ، چندتایی ها و غیره است.

sequences = [10,2,8,7,5,4,3,11,0, 1]

filtered\_answer = filter (lambda x: x > 6, sequences)

print(list(filtered\_answer))

# OUTPUT

# [10, 8, 7, 11]

**# استفاده از lambda با map**

تابع map برای استفاده از یک عملیات خاص برای هر عنصر در یک دنباله استفاده می شود.

sequences = [10,2,8,7,5,4,11]

squared\_result = map (lambda x: x\*x, sequences)

print(list(squared\_result))

# OUTPUT

# [100, 4, 64, 49, 25, 16, 121]

=================================================