

توابع Lambda (توابع ناشناس) در پایتون

۱. تعریف توابع Lambda

توابع Lambda در پایتون توابعی ناشناس و کوتاه هستند که بدون نیاز به تعریف یک تابع کامل با `def`، می‌توانند عملیات ساده‌ای را انجام دهند. این توابع معمولاً برای استفاده‌های سریع و موقت در کد استفاده می‌شوند.

ساختار کلی تابع Lambda:

```
lambda arguments: expression
```

- `arguments`: آرگومان‌هایی که تابع دریافت می‌کند.
- `expression`: بدنه تابع که باید مقدار خروجی را برگرداند.

مثال ساده:

یک تابع Lambda که مجموع دو عدد را برمی‌گرداند:

```
sum_lambda = lambda x, y: x + y  
print(sum_lambda(3, 5)) # خروجی: 8
```

در این مثال، `sum_lambda` یک تابع ناشناس است که دو عدد را دریافت کرده و مجموع آن‌ها را برمی‌گرداند.

۲. استفاده از توابع Lambda در متدهای استاندارد

توابع Lambda در پایتون معمولاً در ترکیب با توابع داخلی (Built-in Functions) مانند `map()`، `filter()`، `sorted()` استفاده می‌شوند.

۱. استفاده از `map()`

تابع `map()` برای اعمال یک تابع روی تمام عناصر یک لیست یا مجموعه استفاده می‌شود.

مثال:

افزودن ۱۰ به تمام اعداد لیست:

```
numbers = [1, 2, 3, 4, 5]  
new_numbers = list(map(lambda x: x + 10, numbers))  
print(new_numbers) # خروجی: [11, 12, 13, 14, 15]
```

۲. استفاده از `filter()`

تابع `filter()` برای فیلتر کردن عناصر یک لیست بر اساس یک شرط خاص استفاده می‌شود.

مثال:

فیلتر کردن اعداد زوج از یک لیست:

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8]  
even_numbers = list(filter(lambda x: x % 2 == 0, numbers))  
print(even_numbers) # خروجی: [2, 4, 6, 8]
```

در اینجا، تابع Lambda شرط `x % 2 == 0` را بررسی کرده و فقط اعداد زوج را نگه می‌دارد.

۳. استفاده از `sorted()`

تابع `sorted()` برای مرتب‌سازی لیست‌ها استفاده می‌شود و می‌توان با `key` یک شرط خاص برای مرتب‌سازی تعیین کرد.

مثال:

مرتب‌سازی لیستی از دیکشنری‌ها بر اساس مقدار کلید `age`:

```
people = [{'name': 'Ali', 'age': 25}, {'name': 'Sara', 'age': 20}, {'name': 'Reza', 'age': 30}]
sorted_people = sorted(people, key=lambda person: person['age'])
print(sorted_people)
```

خروجی:

```
[{'name': 'Sara', 'age': 20}, {'name': 'Ali', 'age': 25}, {'name': 'Reza', 'age': 30}]
```

در اینجا، تابع `Lambda` مقدار `age` را برای مقایسه و مرتب‌سازی لیست دیکشنری‌ها استفاده کرده است.

نتیجه‌گیری

توابع `Lambda` راهی ساده برای نوشتن توابع کوتاه و یک‌خطی در پایتون هستند. این توابع زمانی مفید هستند که نیازی به تعریف یک تابع جداگانه با `def` نداریم، مخصوصاً در ترکیب با متدهایی مانند `filter()`، `map()` و `sorted()`.