8 – Lambda Functions

Le **lambda functions** sono funzioni anonime e compatte in Python, utili quando serve una funzione **veloce e usa-e-getta** senza doverla definire con `def`. Sono particolarmente comode come **callback** o come chiavi per ordinamenti e trasformazioni di dati.

Sintassi

La sintassi base è:

```
lambda arg1, arg2, ... : espressione
```

- Non serve il nome della funzione (anonima).
- Restituisce automaticamente il risultato dell’espressione.
- Può avere qualsiasi numero di argomenti, ma **una sola espressione**.

Esempio semplice

```
doppio = lambda x: x * 2
print(doppio(5))  # 10
```

È equivalente a:

```
def doppio(x):
    return x * 2
```

Lambda con più argomenti

```
somma = lambda a, b: a + b
print(somma(3, 7))  # 10
```

Lambda come argomento di altre funzioni

Molto utile con funzioni come `sorted()`, `map()`, `filter()`:

```
numeri = [5, 2, 9, 1]

# ordinamento per valore assoluto decrescente
ordinati = sorted(numeri, key=lambda x: -x)
print(ordinati)  # [9, 5, 2, 1]
```

```
# raddoppiare tutti i numeri con map
raddoppio = list(map(lambda x: x*2, numeri))
print(raddoppio)  # [10, 4, 18, 2]
```

```
# filtrare numeri maggiori di 4
filtrati = list(filter(lambda x: x > 4, numeri))
print(filtrati)  # [5, 9]
```

Vantaggi

- Sintassi compatta e veloce da scrivere.
- Ideale per funzioni temporanee o callback.
- Si integra perfettamente con `map()`, `filter()`, `sorted()`, `reduce()`.

Limitazioni

- Solo **un'espressione**, niente istruzioni multiple.
- Può diventare difficile da leggere se troppo complessa.

Vuoi procedere con il **paragrafo 9 – Map Function**?