

Python - elementy funkcyjne

Michał Gałka

Elementy funkcyjne w języku Python

Wyrażenia lambda

- Lambda jest funkcją anonimową.
- Tworzona jest przy użyciu słowa kluczowego `lambda`.
- Nie ma instrukcji `return`.
- Zawsze posiada wyrażenie, którego wynik zwraca.

```
lambda x: x*x
```

- `map(f, seq)`
 - Wykonuje funkcję `f` dla każdego elementu z sekwencji `seq`.
 - Wynikiem działania jest nowa lista.

- `filter(f, seq)`
- Wywołuje funkcję `f` dla każdego elementu z sekwencji `seq`.
- Do listy wynikowej trafiają elementy z `seq` dla których funkcja `f` zwróciła `True`.

- `reduce(f, seq)`
 - W pierwszym przebiegu Wywołuje funkcję `f` dla pierwszych dwóch elementów z sekwencji `seq`
 - W kolejnym przebiegu funkcja wywoływana jest dla wyniku poprzedniego działania oraz kolejnego elementu z sekwencji `seq` itd.
 - Wynikiem jest wartość osiągnięta z powyższej operacji, bo osiągnięciu końca sekwencji.

List comprehension i wyrażenia generujące

List comprehension

- List comprehension to wyrażenia pozwalające na stworzenie listy na podstawie opisu.
- Przypominają nieco matematyczną definicję zbiorów.

List comprehension - przykład

```
square_values = [x**2 for x in range(10)]
```

```
import string
text = 'Lorem ipsum dolor sit amet.'
list_of_uppercase_letters = [letter.upper() for
                             letter in text if letter in string.ascii_letters]
```

- Wyrażenia generujące (ang. generator expressions) definiuje się podobnie do list comprehension.
- Zamiast listy zwracają one generator tworzący wartości wg. ustalonych reguł.

Wyrażenia generujące - przykład

```
import string
text = 'Lorem ipsum dolor sit amet.'
gen = (letter.upper() for letter in text if letter
       in string.ascii_letters)

for a in gen:
    print(a)
```

- W sposób podobny do przedstawionych powyżej można generować także słowniki.
- Wyrażenie ujęte jest wtedy w nawiazy klamrowe.

Dict comprehension - przykład

```
import string
ascii_codes = {letter: ord(letter) for letter in
               string.ascii_letters}
```