

Iteratory i Generatory

Iterator to obiekt, który umożliwia iterowanie po kolekcji, takiej jak lista lub słownik. Iterator przechowuje stan iteracji i umożliwia pobieranie kolejnych elementów kolekcji, dopóki nie zostanie osiągnięty koniec kolekcji. Aby utworzyć iterator, musisz najpierw utworzyć obiekt kolekcji, a następnie użyć funkcji `iter()`.

Generator to rodzaj iteratora, który generuje wartości "na żądanie" zamiast przechowywać wszystkie wartości w pamięci. Aby utworzyć generator, używasz funkcji `yield` zamiast `return`. Generatory są bardziej wydajne w przetwarzaniu dużych zestawów danych, ponieważ nie muszą przechowywać wszystkich wartości w pamięci jednocześnie.

Warto zauważyć, że generator jest iteratorem, ale nie każdy iterator jest generatorem. Iterator to bardziej ogólny koncept, który może być używany do iterowania po różnych typach kolekcji, podczas gdy generator jest specjalnym rodzajem iteratora, który generuje wartości z kodu wywołującego.

```
# Utworzenie listy
my_list = [1, 2, 3, 4, 5]

# Utworzenie iteratora
my_iterator = iter(my_list)

# Pobranie kolejnych elementów z iteratora
print(next(my_iterator)) # 1
print(next(my_iterator)) # 2
print(next(my_iterator)) # 3

# Utworzenie generatora
def my_generator():
    yield 1
    yield 2
    yield 3
    yield 4
    yield 5

# Użycie generatora
for value in my_generator():
    print(value)
```