



Pętla WHILE, instrukcja BREAK i CONTINUE

Wprowadzenie do języka Python (VI)





Wcześniej była mowa o tworzeniu pętli z użyciem instrukcji FOR. Możemy też kontynuować pętlę tak długo, jak warunek jest prawdziwy z pętlą WHILE. Pętla WHILE jest używana, gdy nie wiesz ile razy będziesz musiał powtórzyć pętlę.

Tutaj wygenerujemy liczbę losową za pomocą modułu random i funkcji randrange(). Wtedy będziemy używać pętli WHILE, aby odgadnąć losową wartość i wyprowadzić ją.





KOD

```
# Możemy użyć modułu random do generowania liczb losowych
```

```
import random
```

```
# Wygeneruj losową liczbę całkowitą od 1 do 50
```

```
rand_num = random.randrange(1, 51)
```

```
# Wartość, którą zwiększamy w pętli while jest zdefiniowana przed pętlą
```

```
i = 1
```





```
# Zdefiniuj warunek, że dopóki True będziemy kontynuować pętlę
while (i != rand_num):
    # Musisz zwiększyć swój iterator wewnątrz pętli while
    i += 1
# Poza pętlą while, gdy przestajemy dodawać wartości do zmiennej
print("Losowa wartość to : ", rand_num)
```





BREAK i CONTINUE

Instrukcje `BREAK` i `CONTINUE` są bardzo przydatne. `CONTINUE` zatrzymuje wykonywanie kodu, który pozostaje w pętli i skacze z powrotem na górę (początek pętli). Podczas gdy instrukcja `BREAK` kończy wykonanie i przeskakuje bezpośrednio do kodu, który leży bezpośrednio poza pętlą.

Tutaj przejdziemy od 0 do 20 z pętlą `WHILE`. Jeśli liczba jest parzysta, należy pominąć jej wypisywanie. Jeśli liczba jest nieparzysta, to należy ją wypisać. Następnie zakończymy działanie pętli, jeśli wartość kiedykolwiek osiągnie 15.





```
i = 1
while i <= 20:
    # Jeśli numer jest parzysty, nie wypisuj go
    if (i % 2) == 0:
        i += 1
        continue
    # Jeśli i równa się 15, zatrzymaj pętlę
    if i == 15:
        break
    # Wydrukuj liczby nieparzyste
    print("Liczba nieparzysta: ", i)
    # Zwiększ i
    i += 1
```

