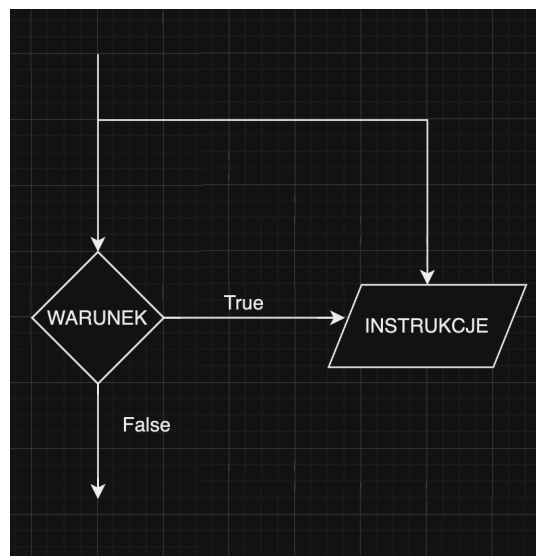


## Instrukcja while

Pętla **while** działa w następujący sposób, wykonuje instrukcję, dopóki jest spełniony zadany warunek. Tworząc pętlę **while** musimy zadbać o to by liczba jej wywołań była skończona. Oczywiście jest możliwość wykonania instrukcji iteracyjnej nieskończenie długo wstawiając w warunek True.

### Pętla while



### Przykład pętli while:

```
while warunek:  
    Instrukcja  
    Instrukcja  
    Instrukcja  
    ...
```

### Przykład:

Napisz program, który wyświetli liczby z zakresu 0..9.

```
while.py x  
1 i = 1  
2 while i<=10:  
3     print(i)  
4     i = i + 1  
5
```

## Przykład

Napisz program wyznaczający sumę cyfr podanej liczby naturalnej.

```
while.py x
1 suma = 0
2 liczba = 123
3 while liczba > 0:
4     suma = suma + liczba % 10
5     liczba = liczba // 10
6     print(suma)
7
```

## Nieskończona pętla while w Pythonie

Jeśli warunek w pętli ma zawsze wartość True, pętla będzie działać nieskończoną ilość razy (aż do zapełnienia pamięci). Na przykład:

```
while.py x
1 age = 32
2
3 # ten warunek jest zawsze prawdziwy,
4 # zmienna age nie jest modyfikowana w tym kodzie
5 while age > 18:
6     print('Możesz iść na dyskotekę.')
7
```

W powyższym przykładzie warunek zwraca zawsze wartość True. W związku z tym treść pętli będzie działać nieskończenie wiele razy.

## Pętla while z alternatywą

W języku Python podczas działania pętli **while** możesz dodać opcję **else**. Instrukcje po **else** wykonane będą, kiedy warunek nie jest prawdziwy (zwróci False).

```
while.py x
1   counter = 0
2
3   while counter < 3:
4       print('counter:', counter)
5       counter = counter + 1
6   else:
7       print('Warunek w pętli while nie został spełniony.')
8
```

### Uwaga:

Blok **else** nie zostanie wykonany, jeśli pętla **while** zostanie zakończona instrukcją **break**.

```
while.py x
1   counter = 0
2
3   while counter < 3:
4       print('counter:', counter)
5       counter = counter + 1
6       if counter == 1:
7           break
8   else:
9       print('Warunek w pętli while nie został spełniony.')
10
```