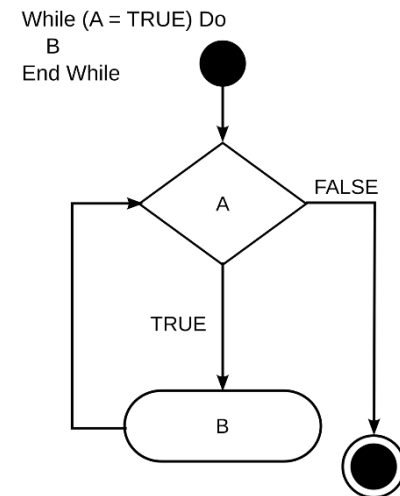




Pętla WHILE w Pythonie

M@я3k Pυđ€£kØ

Programowanie w Pythonie

Spis treści

- Pętla WHILE
- Pętla WHILE w Pythonie
- Cechy pętli WHILE
- Pętle zagnieżdżone
- Pętla nieskończona
- Instrukcja `break`.
- Instrukcja `continue` .

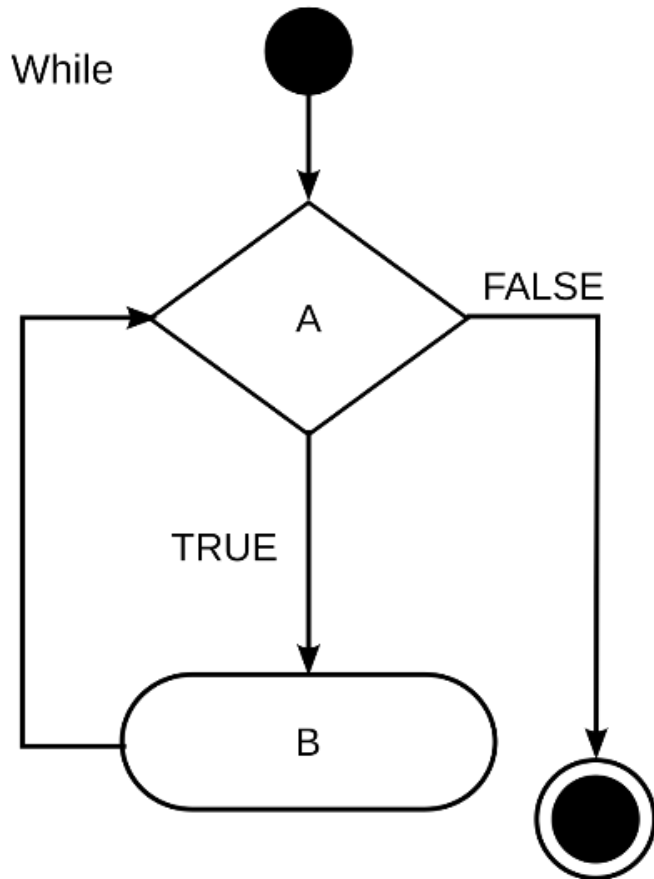
Pętla WHILE

- Pętla WHILE to pętla, która powtarza blok kodu tak długo, jak długo określony warunek jest prawdziwy.
 - Najpierw sprawdza warunek. Gdy jest spełniony (True), wykonuje instrukcje i ponownie sprawdza warunek.
 - Jeśli warunek jest fałszywy (FALSE) pętla przerywa pracę.
- Pętla jest użyteczna, gdy liczba powtórzeń nie jest znana z góry.

While (A = TRUE) Do

B

End While



Pętla WHILE w Pythonie

- Pętla WHILE w Pythonie pozwala na wykonanie pętli w zależności od warunku na początku pętli.
- Wykonuje go tak długo, jak długo dany warunek jest prawdziwy (`True`).
 - Po każdej iteracji warunek jest sprawdzany ponownie. Jeśli nadal jest prawdziwy, pętla kontynuuje pracę.
 - Jeśli stanie się fałszywy (`False`), pętla się kończy.

```
while warunek:  
    instrukcja_1  
    instrukcja_2  
    ...  
zmiana warunku pętli
```

Cechy pętli WHILE

- Sprawdzenie warunku:
 - Warunek jest sprawdzany przed wykonaniem bloku instrukcji w każdej iteracji. Jeśli warunek jest początkowo fałszywy, blok kodu w ogóle się nie wykona.
- Nieznana liczba iteracji:
 - Jest to główny przypadek użycia pętli while. Stosuje się ją, gdy nie wie wiadomo z góry, ile razy kod ma się powtórzyć (np. wczytywanie danych od użytkownika do momentu wprowadzenia poprawnej wartości).
- Wcięcie:
 - Blok kodu pętli jest definiowany za pomocą wcięcia, podobnie jak w instrukcjach `if`.
- Instrukcje `break` i `continue`:
 - Można ich użyć do przerywania pętli while (`break`) lub pominięcia pozostałej części bieżącej iteracji i przejścia do następnej (`continue`).
- Pętla nieskończona:
 - Jeśli warunek nigdy nie przestanie być spełniony (np. `while True`), pętla będzie wykonywana w nieskończoność, dopóki program nie zostanie zatrzymany.
- Opcjonalny blok `else`:
 - Pętla while może posiadać blok `else`, który wykonuje się tylko wtedy, gdy pętla zakończyła działanie naturalnie (przez spełnienie warunku), a nie została przerwana instrukcją `break`.
- Dwukropek:
 - Na końcu linijki z `while` należy postawić dwukropek :

Ćwiczenie 1

1. Napisz program wypisujący liczby od **n** do **m**. Liczby **n** i **m** wczytaj z klawiatury. Użyj pętli WHILE.
2. Napisz program wypisujący kolejne potęgi liczby 2 od 1 do **n**. Liczbę **n** wczytaj z klawiatury.
3. Wczytaj liczbę **x** z klawiatury. Użyj operatorów `//` i `%`, aby obliczyć sumę jej cyfr.
4. Dany jest pierwszy wyraz $a_1 = 2$ i iloraz $q = 3$. Oblicz **n**-ty wyraz ciągu geometrycznego przy pomocy operatora potęgowania `**`.
5. Liczba mieszkańców miasta wynosi 1000. Co roku rośnie o 3 %. Oblicz, ile osób tam będzie mieszkało po **n** latach.
6. Zmienna **x** = 1. W każdej iteracji pętli zwiększ ją o 50 % (`x *= 1.5`) przez 10 kroków. Wypisz wartość **x** po każdej zmianie.
7. Pobierz liczbę całkowitą **n**. Oblicz **n!** (silnię).
8. Wczytaj liczbę **a** i procent **p**. Oblicz jak rośnie kwota **a** przy wykorzystaniu procentu składanego.
 - Procent składany polega na tym, że liczbę **a** zwiększamy za każdym razem o wartość procentu **p**.

Pętle zagnieżdżone

- W języku Python możemy umieszczać jedne pętle w drugich. Dotyczy to również pętli WHILE:

```
i = 1
while i < 11:
    j = 1
    while j < 11:
        k = i*j
        if j == 10:
            print(k)
        else:
            print(k, "\t", end = " ")
        j += 1
    i += 1
```

- Pętla zewnętrzna kontroluje liczbę pełnych wykonani pętli wewnętrznej.
- Pętla wewnętrzna jest wielokrotnie wykonywana w całości dla każdej iteracji pętli zewnętrznej. Musi zakończyć wszystkie swoje iteracje, zanim pętla zewnętrzna przejdzie do kolejnej iteracji.

Ćwiczenie 2

1. Napisz program wczytujący liczby n i m . Następnie narysuj prostokąt złożony z gwiazdek o wymiarach n na m .
2. Napisz program wczytujący liczby n i m . Następnie narysuj ramkę złożoną z gwiazdek o wymiarach n na m .
3. Napisz program wypisujący liczby w trójkącie. Wczytuje liczbę n wypisuje liczby według wzoru na rysunku.
4. Napisz program rysujący szachownicę. Wczytuje rozmiar wiersza i kolumny, oraz znaki jakie ma rysować naprzemian.

```
*****
*****
*****
```

```
*****
*   10   *
*   5    *
*       *
*****
```

```
1
12
123
1234
12345
```

```
1 2 3
4 5 6
7 8 9
```

```
@ _ @ _ @ _
 _ @ _ @ _ @
 @ _ @ _ @ _
 _ @ _ @ _ @
```

```
*
**
***
****
*****
```

5. Napisz program wypisujący liczby występujące na wszystkich kostkach domina.
 - (Przykład: 0-0, 0-1, ... 5-5, 5-6, 6-6)
6. Napisz program wczytujący liczbę n i rysujący kwadrat $n \times n$. W kolejnych rzędach mają być kolejne liczby od 1 do n
111
222
333
7. Napisz program wczytujący liczbę n i wypisujący liczby od 1 do n^2 . Rysuje je w kwadracie $n \times n$.
8. Napisz program wczytujący liczbę n i rysujący trójkąt z gwiazdek o wymiarach $n \times n$ według wzoru na rysunku.

Pętla nieskończona

- Kluczowym warunkiem jest wyrażenie logiczne, które jest oceniane przed każdą iteracją pętli.
- Jeśli warunek nigdy nie stanie się fałszywy, pętla będzie działać w nieskończoność.

```
x = 0
```

```
while x != 7:
```

```
    print (x)
```

- Należy zadbać o to, aby wewnątrz pętli znajdowała się instrukcja, która ostatecznie zmieni warunek na `False`.

```
i = 0
```

```
x = 0
```

```
while (x != 7) or (i <= 100):
```

```
    print (x)
```

```
    i += 1
```

Instrukcja break

- Instrukcja `break` pozwala natychmiast przerwać pętlę w której się znajduje, niezależnie od warunku pętli.
- Po wykonaniu `break`, program przerywa wykonywanie pętli i kontynuuje działanie od pierwszej instrukcji znajdującej się po niej.
- Często jest używana w połączeniu z instrukcją `if`, aby przerwać pętlę, gdy zostanie spełniony określony warunek.

```
n = 100
x = 10
while x >= -10:
    if x == 0:
        print ("Dzielenie przez zero!")
        break          #instrukcja przerywająca pracę pętli
    if (n % x == 0):
        print ("Dzielnikiem ", n, " jest ",x)
    x -= 1
                                #pierwsza instrukcja poza pętlą
print ("Przerwano pętlę WHILE")
```

Instrukcja continue

- Instrukcja `continue` w Pythonie służy do pominięcia bieżącej iteracji pętli i przejścia do następnej.
- W odróżnieniu od `break`, która przerywa całą pętlę, `continue` przerywa tylko obecny cykl i nie wykonuje pozostałego kodu wewnątrz pętli dla tej iteracji, po czym natychmiast rozpoczyna następną.

```
n = 100
x = 11
while x >= -10:
    x -= 1
    if x == 0:
        print ("Dzielenie przez zero!")
        #instrukcja przerywająca daną iterację
        continue
    if (n % x == 0):
        print ("Dzielnikiem ", n, " jest ",x)
```

Powtórzenie

1. Co robi pętla `While`?
2. Jak jest realizowana w języku python?
3. Jak zagnieżdżać pętle `While` w pythonie?
4. Jak uzyskać pętlę nieskończoną?
5. Jaką rolę pełni instrukcja `break`?
6. Jaką rolę pełni instrukcja `continue`?