Pętla WHILE

Komenda ***WHILE*** rozpoczyna blok zadanych instrukcji, które będą wykonywać się tak długo, jeżeli zadany warunek zostanie spełniony.

Pętla ***WHILE*** zawiera:

- słowo kluczowe ***WHILE***
- warunek
- blok kodu, który będzie wykonywał się w pętli

Warunek po każdym wykonaniu jest ponownie sprawdzany:

- Jeżeli warunek jest prawdziwy (True) to kod w bloku jest wykonywany.
- Jeżeli warunek jest fałszywy –(False), blok kodu wewnątrz pętli nie zostanie wykonany.

Przykład:

```
a = 3
while a > 2:
     a = int(input("Podaj liczbę większą od 2: "))
     print("Podano liczbę a =", a)
print("Twoja liczba nie była większa od 2")
```

Jeżeli warunek byłby cały czas spełniony to pętla WHILE wykonywałaby się bez końca. Instrukcja while True w Pythonie to pętla nieskończona.

Przykład:

```
while True:
     print("Wyświetlaj do woli!")
```

Podobnie jak w pętli ***FOR***, możemy korzystać z intrukcji ***BREAK*** i ***CONTINUE***:

- break – kończy działanie pętli. Interpreter przechodzi do dalszej części instrukcji – następujących po bloku pętli.
- continue – kończy iterację bieżącej pętli. Interpreter wraca do początku pętli, wyrażenie warunkowe jest ponownie sprawdzane, aby określić, czy pętla zostanie wykonana ponownie, czy zakończyć i przejść dalej.

Przykład z instrukcją break:

```
a = 10
while a < 30:
     a += 1
     if a == 20:
         break
     print("Aktualny numer to", a)
print("Przerwanie, Jesteś poza pętlą")
```

Efekt:

```
Aktualny numer to 11
Aktualny numer to 12
Aktualny numer to 13
Aktualny numer to 14
Aktualny numer to 15
Aktualny numer to 16
Aktualny numer to 17
Aktualny numer to 18
Aktualny numer to 19
Przerwanie, Jesteś poza pętlą
>>>
```

Przykład z instrukcją continue:

```
a = 10
while a < 30:
     a += 1
     if a == 20:
         continue
     print("Aktualny numer to", a)
print("Pominąłeś liczbę 20 i zakończyłeś pętlę")
```

Efekt:

```
Aktualny numer to 11
Aktualny numer to 12
Aktualny numer to 13
Aktualny numer to 14
Aktualny numer to 15
Aktualny numer to 16
Aktualny numer to 17
Aktualny numer to 18
Aktualny numer to 19
Aktualny numer to 21
Aktualny numer to 22
Aktualny numer to 23
Aktualny numer to 24
Aktualny numer to 25
Aktualny numer to 26
Aktualny numer to 27
Aktualny numer to 28
Aktualny numer to 29
Aktualny numer to 30
Pominąłeś liczbę 20 i zakończyłeś pętlę
>>>
```

Zadanie 1

W pierwszej kolejności poczytaj o module random (import random). W drugiej kolejności napisz skrypt na grę zgadywankę. Komputer będzie losował wartość z przedziału liczb całkowitych 1-20. Poproś użytkownika aby zgadnął liczbę wylosowaną przez komputer. Program pyta użytkownika o podanie liczby tak długo, dopóki gracz nie zgadnie. Jeżeli użytkownik nie zgadnie liczby to gra musi go poinformować czy wybrana liczba jest większa lub mniejsza od tej, którą wylosował komputer.

Zadanie 2

Napisz program z wykorzystaniem pętli while, który dla 10 kolejnych liczb naturalnych wyświetli sumę poprzedników.
Oczekiwany wynik: 1, 3, 6, 10, 15, 21, 28, 36, 45, 55

Zadanie 3

Napisz program, który będzie liczył silnię. Rozwiąż zadanie za pomocą pętli for oraz pętli while.
Dane wejściowe: Podaj dowolną liczbę całkowitą do 10
Dane wyjściowe: Silnia to, np.: 4! = 24

Dla chętnych na ocenę 5

Zadanie 4

Wykorzystaj moduł random do napisania skryptu do zgadywanki. Komputer losuje jedno słowo z zadanego zakresu (z 6 zadeklarowanych wyrazów). W następnej kolejności litery tego słowa są mieszane, a użytkownik musi zgadnąć co to za słowo. Użytkownik zgaduje do skutku, dopiero zgadnięcie przerywa grę, a jeżeli użytkownik nie chce już grać to wciskając „q” lub „Q” kończy działanie programu.