Цикл while, for

За допомогою циклів можна організувати повторення частини кода. Така необхідність виникає дуже часто. Наприклад, користувач вводить послідовно числа, а їх поочередно потрібно додавати до загальної суми.

Усього в Python є два типа циклів: while та for.

Цикл While

“While” перекладається як «поки», а якщо бути конкретнішим, то «поки виконується це, виконуємо це». Власне, це і є сутність цього типа циклу.

Цикл while має заголовок та тіло. У заголовку перевіряється умова та, якщо вона повертає значеняя True, відбувається код тіла. Після завершення останньої команди тіла циклу, інтерпретатор повертається до заголовку та робить перевірку заново. Таке коло буде відбуватися до тих пір, поки з заголовка не повернеться значення False.

Дуже важливо *надавати можливість змінній, по якій йде перевірка, змінювати своє значення*. Якщо заголовок буде весь час повертати істину, то цикл не закінчиться ніколи.

Приклад циклу:

```
a = int(input('Число '))
while a < 100:
    a += 10
    print(a)
```

Та його виконання:

```
Число 33
43
53
63
73
83
93
103
```

Цикл for

Цикл for у Python призначений для перебора елементів структур даних (наприклад, списків). *Перебор елементів структури даних*– це послідовне вийняття кожного елементу структури та дії з ними.

У заголовку цикла for знаходиться масив на назва локальної змінної, яка використовується для зберігання кожного елемента, який береться у масиві. Після завершення циклу ця змінна зчезає.

```
spis = [7, 3.42, -14, 0]
for i in spis:
```

Тіло цикла for – це набір команд, які виконуються з кожним елементом массива послідовно.

Приклад циклу for:

```
spis = [7, 3.42, -14, 0]
for i in spis:
    i += 2
    print(i)
```

Результат виконання циклу:

```
9
5.42
-12
2
```

При цьому у самому списку зміни не зберігаються:

```
>>> print(spis)
[7, 3.42, -14, 0]
```

Буквою «i» ми позначаємо елемент списка. Кожне наступне коло циклу буде брати елемент з індексом на 1 більший, за попередній. Замість «i» може бути будь-яка літера або сполучення літер (нова змінна).

Можемо замінити цей цикл більш складною конструкцією, яка містить while:

Цикл:

```
spis = [7, 3.42, -14, 0]
i = 0
while i < len(spis):
    print(spis[i]+2)
    i += 1
```

Результат виконання циклу:

```
9
5.42
-12
2
```

Функція *len()* визначає кількість елементів у структурі даних.

Цей метод також не змінює початковий список і є рівносильним данному циклу for.

Внесення змін у сам масив розглядатиметься пізніше. Він можливий як для циклу for, так і для while.

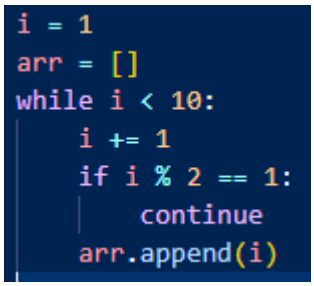
Break, continue

Для виходу з циклу використовується інструкція *break*. Зачасту вона знаходиться в тілі оператора if або else.

```
i = 1
while(True):
    if i > 10:
        break
    else:
        i += 1
```

В прикладі після збільшення лічильника i до 11 цикл завершиться – спрацює інструкція break.

Для пропуску ітерації використовується інструкція *continue*. Зазвичай вона також знаходиться в тілі оператора if або else.



```
i = 1
arr = []
while i < 10:
    i += 1
    if i % 2 == 1:
        continue
    arr.append(i)
```

У цьому прикладі всі непарні значення і будуть активувати пропуск ітерації і, відповідно, не будуть додані до списку.

Контрольні питання:

1. Для чого використовують цикли?
2. Чим відрізняється цикл while від for?
3. При якій умові буде виконуватись тіло циклу while?
4. Пояснити інструкції break і continue.
5. Як утворити нескінченний цикл?