

Python

04

Функції агрегації

sum() - обчислює суму елементів в об'єкті, що ітерується (списку, кортежі і т.д.)

max() - повертає максимальний елемент в об'єкті, що ітерується

min() - повертає мінімальний елемент в об'єкті, що ітерується

len() - повертає кількість елементів в об'єкті, що ітерується

any() - повертає **True**, якщо хоча б один елемент об'єкта, що ітерується, відповідені істинні

all() - повертає **True**, якщо всі елементи об'єкта, що ітерується, відповідні істинні

sorted() - сортує елементи об'єкта, що ітерується

Цикли

- a. While - поки умова виконується
- b. For - перебір
- c. break - оператор переривання циклу
- d. continue - оператор зупинки поточної ітерації та переходу на нову
- e. else - і тут без нього не обійшлося)

Цикли

Цикл while

```
while boolean_expression:  
    while_suite  
else:  
    else_suite
```

Цикл for ... in

```
for expression in iterable:  
    for_suite  
else:  
    else_suite
```

Функція виводу print()

print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)

- objects - об'єкт, який потрібно вивести * означає, що об'єктів може бути кілька;
- sep - поділяє об'єкти. Значення за замовчуванням: ' ';
- end - ставиться після всіх об'єктів;
- file - очікується об'єкт із методом write (string). Якщо значення не встановлено, для виведення об'єктів використовується файл sys.stdout;
- flush - якщо встановлено значення True, потік примусово скидається у файл. Значення за замовчуванням: False.

Функція введення input()

input(object)

- object - необов'язковий строковий аргумент.
Функція завжди повертає відповідь у строковому вигляді.

Цикли

Звичайний цикл

```
for i in [1, 2, 3]:  
    print(i)
```

```
1  
2  
3
```

Цикл з break

```
for i in [1, 2, 3]:  
    if i == 2:  
        break  
    print(i)
```

```
1
```

Цикл з continue

```
for i in [1, 2, 3]:  
    if i == 2:  
        continue  
    print(i)
```

```
1  
3
```

Генератори списків

У мові програмування Python є спеціальна синтаксична конструкція, яка дозволяє за певними правилами створювати заповнені списки. Такі конструкції називають генераторами списків. Їх зручність полягає у більш короткому запису програмного коду, ніж якби створювався список звичайним способом.

Конструкція **[i for i in range(1,15)]** є генератором списку. Вся конструкція полягає у квадратних дужках, що ніби каже, що буде створено список. Усередині квадратних дужок можна виділити три частини:
1) що робимо з елементом (в даному випадку нічого не робимо, просто додаємо до списку);
2) що беремо (в даному випадку елемент i);
3) звідки беремо (тут з об'єкта range) . Частини відокремлені одна від одної ключовими словами for та in.