

Прості лінійні алгоритми

Лінійний алгоритм означає, що обчислювальний процес буде виконуватися послідовно, без пропуску яких-небудь частин.

Оператори

Лінійний обчислювальний процес на мові Python передбачає оператори, які виконуються послідовно:

1. Оператор присвоювання =.

Відбувається присвоювання значення змінній. Спочатку виконується операція, що знаходиться справа від оператора присвоювання, а результат виконання записується у змінну, що стоїть зліва.

2. Оператор складання або конкатенації +.

Оператор “+” має два значення, залежно від даних, до яких він використовується. Якщо “+” застосовується до чисел, то відбудеться складання цих чисел; а якщо до строк – конкатенація (об'єднання строк).

*Конкатенація — операція склеювання об'єктів лінійної структури.
Наприклад, конкатенація слів «мікро» і «скоп» дасть слово «мікроскоп».*

3. Оператор множення *.

Оператор множення має також два значення. Він може використовуватися як для двох чисел (множення), так і для числа та строки (повторення строки n кількість разів).

4. Арифметичні оператори -, /, **.

Окрім операторів складання та множення Python має також інші арифметичні оператори, які можуть бути використані тільки стосовно чисел. Це – “-” (віднімання), “/” (ділення) та “**” (возведення у степінь).

5. Логічні оператори ~, >>, <<, &, |, ^.

Логічні оператори виконують дію побітово з числом у двійковій системі зчислення.

“~” – інверсія, значення кожного біту числа змінюється на протилежне.

“>>” та “<<” – оператори зсуву вправо або вліво на вказану кількість біт.

“&” – бітовий оператор І (AND)

“|” – бітовий оператор АБО (OR)

“^” – бітовий оператор ВИКЛЮЧНЕ АБО (XOR)

Типи даних

У мові Python існує декілька типів даних: цілі числа (*int*), дробові числа (*float*), строки (*str*), логічні вирази (*bool*) та інші, більш складні типи.

Цілі числа можуть взаємодіяти з дробовими та отримувати в результаті дробове число ($3+7.2=10.2$). Строки можуть бути сполучені тільки з строками ('при'+ 'віт'='привіт'). Спробу поєднати число зі строкою інтерпретатор виявить як помилку *ValueError*. Але можливе перетворення одного типу даних в інший. Так, будь-яке число (у тому числі і дробове) можна перетворити у строку. Для цього потрібно виконати наступне:

```
a = 4          #type(a) = int
b = 4.2        #type(b) = float
a = str(a)     #type(a) = str
b = str(b)     #type(b) = str
```

Також можливе перетворення з дробового числа у ціле та навпаки. У першому випадку відбувається «відкидання» дробової частини:

```
b = 4.99      #type(b) = float
b = int(b)    #type(b) = int, b=4
```

У другому випадку число відображається з одним нулем після крапки.

Функції

Окрім операторів дії можуть виконувати команди мови програмування. Для переводу в інші типи даних використовують саме функції.

Функція `type()` повертає тип даних змінної, що записана в дужках.

Функція `print()` дозволяє вивести дані на екран.

Функція `input()` дозволяє користувачеві ввести дані. Повертає строку.

При вводі даних користувачем у функцію `input()`, дані автоматично стають строкового типу. Тому іноді виникає необхідність переводити їх у інші типи. Процедура переводу приведена вище, але для економії об'єму програми можна використовувати наступний вираз:

```
a = int(input('Введіть ціле число'))
```

Таким чином, ми одразу перетворюємо введені дані у ціле число.

Щоб округлити результат обчислень використовують функцію `round()`. В цій команді в дужках пишеться змінна або число, яке потрібно округлити та, через кому, кількість знаків після крапки. Наприклад:

```
a = 7.27494675
a = round(a, 2)  #результат округляється до 2 знаків після коми
```

Контрольні питання:

1. Перечислити прості типи даних у Python.
2. Як працює оператор додавання для двох строк?
3. Як вивести інформацію на екран?
4. Що відбудеться, якщо перевести дробове (float) число в ціле (int)?