

Особливості мови програмування Python, порівняно з розглянутими Pascal та VBA (LibreOffice VB) полягає в тому, що Python практично не передбачає визначеної структури для дрібних програм. Python - об'єктно орієнтована мова програмування, в якій практично кожен компонент є об'єктом і має набір властивостей, притаманних об'єктам відповідного типу.

Структурно, програма мовою Python, представляє собою набір інструкцій мови, розташованих одна під одною. Інструкції виконуються послідовно, якщо інше не передбачено використанням умовних операторів, операторів циклу, тощо.

Блоків опису змінних в програмі мовою Python не передбачено. Змінна у різних частинах програми може мати різний тип.

Для невеликих програм розрахункового плану важливими є два основні положення:

- Змінна ініціалізується значенням. Її тип відповідає типу присвоєного значення. Використання змінної, якій не надано значення не допускається. Таким чином, усі змінні, які використовуються в програмі, вперше повинні з'явитись в якості лівої частини оператора присвоювання:

```
my_var1=2
my_var2=my_var1**2
```

- Якщо в програмі використовуються функції зовнішнього модуля (наприклад модуль math для математичних розрахунків), то відповідний модуль необхідно підключити (імпортувати) до першого застосування описаних у ньому функцій:

```
import math
```

Принциповою відмінністю Python від розглянутих мов програмування є те, що Python регістрозалежний - розрізняє великі та малі літери. Тому My_Var, my_var, my_Var - три окремі змінні.

Ввод - вивід даних

Інструкцією вводу даних в Python є input, однак повний набір засобів вводу залежить від версії Python. У версії 2.7.X існують інструкції **input()** та **raw_input()**.

У версії 3.X інструкція **raw_input()** замінена інструкцією **input()**, а інструкція **input()** може бути відтворена через **eval(input())**.

Інструкція **raw_input()** призначена для введення рядка символів, тому при виконанні коду програми:

```
a=raw_input(" <= a ")      >>>
b=raw_input(" <= b ")      <= a 123
print(a+b)                 <= b 254
                             123254
                             >>>
```

Заміна інструкції вводу на `input()` покаже інший результат:

```
a=input(" <=  a ")          >>>
b=input(" <=  b ")          <=  a 123
print(a+b)                  <=  b 254
                             377
                             >>>
```

В дужках, що слідує за словом інструкції, вказують рядкову константу, що служить підказкою. Ознакою того, що константа рядкова є розміщення її в одинарних (' ') чи подвійних (" ") лапках. Текст константи програмою не обробляється і виводиться без змін при очікуванні вводу значення змінної.

Операція виводу здійснюється за допомогою інструкції **print**, яка в різних версіях Python також має свої особливості. Універсальний варіант інструкції (працює в обох версіях) має вигляд:

```
print('String for output'),
```

де замість 'String for output' знаходиться простий чи складений рядковий об'єкт:

```
print('A= ' + str(a)+' B= '+str(b))
print("No real roots")
print("a= "+str(a)+"\n"+"b= "+str(b)+"\n"+"a+b="+str(a+b))
```

При виводі окремих значень функції перетворення до рядкового типу (**str()**) можна опустити.

Математичні обчислення

Математичні операції доступні в Python:

+ -додавання

- -віднімання

* -множення

/ -ділення

% - залишок від цілочисельного ділення

** - степінь

// - ділення з округленням результату до меншого цілого

Стандартне середовище виконання програми Python містить дуже мало математичних функцій. Серед них можна вказати:

Функція	Тип аргумента	Тип результату	Значення
<code>abs(x)</code>	Число	Число	Модуль числа x
<code>divmod(x, y)</code>	Два цілих	Кортеж (tuple)	Частка та залишок від цілочисельного ділення x на y .
<code>pow(x, y [,z])</code>	Числа	Число	Степінь y числа x . Якщо задано параметр z повертає залишок від ділення результату на z
<code>sum(iterable [, start])</code>	Перечислювальні об'єкти (списки, кортежі)	Сума об'єктів	Повертає суму об'єктів, вказанх в дужках. Якщо задано параметр start - виступає як база.

Обчислювальні можливості Python збільшують через використання зовнішнього модуля `math`:

```
import math
```

Тоді доступ до функцій, описаних в модулі здійснюється за синтаксисом:

```
module.function
```

Список функцій модуля `math`:

Функція	Тип аргумента	Тип результату	Значення
<code>math.ceil(x)</code>	float	integer	Найменше ціле, що більше або рівне x
<code>math.copysign(x, y)</code>			Надає x знак y
<code>math.fabs(x)</code>	numerical	numerical	Абсолютне значення (модуль) числа
<code>math.factorial(x)</code>	Позитивне ціле	ціле	Факторіал числа
<code>math.floor(x)</code>	float	integer	Найбільше ціле, що менше або рівне x
<code>math.fmod(x, y)</code>	Два числа	дійсне	Залишок від ділення x на y . Має підвищену точність при роботі з дійсними числами
<code>math.frexp(x)</code>	numerical	Дійсне і ціле	Мантиса і степінь для представлення числа у вигляді x=m*2**e
<code>math.fsum(iterlist)</code>			Аналог функції sum , але з вищою точністю
<code>math.isinf(x)</code>	numerical	boolean	Перевіряє, чи є число +/- нескінченністю
<code>math.isnan(x)</code>		boolean	Перевіряє, чи аргумент NaN (not a number)
<code>math.ldexp(x, i)</code>	Дійсне і ціле	дійсне	Повертає: z=x*(2**i) . Обернено до math.frexp
<code>math.modf(x)</code>	Дійсне	Два дійсних	Повертає кортеж з двох дійсних чисел, що є дрободією та цілою частиною x
<code>math.trunc(x)</code>	дійсне	дійсне	Повертає число, обрізане до цілого
<code>math.exp(x)</code>	дійсне	дійсне	Експонента числа x
<code>math.exp(x)</code>	дійсне	дійсне	exp(x)-1 для збереження точності при малих x
<code>math.log(x [,base])</code>	дійсне	дійсне	Натуральний логарифм або логарифм за основою base , якщо вказана
<code>math.log1p(x)</code>	дійсне	дійсне	Натуральний логарифм від (1+x) для збереження точності при малих x
<code>math.log10(x)</code>	дійсне	дійсне	Логарифм від x за основою 10
<code>math.pow(x, y)</code>	дійсне	дійсне	Степінь числа x**y з підвищеною точністю
<code>math.sqrt(x)</code>	Дійсне, позитивне	дійсне	Квадратний корінь
<code>math.sin(x)</code>	дійсне	дійсне	Синус аргумента в радіанах
<code>math.cos(x)</code>	дійсне	дійсне	Косинус аргумента в радіанах
<code>math.tan(x)</code>	дійсне	дійсне	Тангенс аргумента в радіанах
<code>math.hypot(x, y)</code>	Два дійсних	дійсне	Евклідова норма. sqrt(x*x+y*y)
<code>math.asin(x)</code>	дійсне	дійсне	Арксинус аргумента
<code>math.acos(x)</code>	дійсне	дійсне	Арккосинус аргумента
<code>math.atan(x)</code>	дійсне	дійсне	Арктангенс аргумента
<code>math.atan2(x, y)</code>	Два дійсних	дійсне	Кут, визначений двома катетами прямокутного трикутника
<code>math.degrees(x)</code>	дійсне	дійсне	Перетворення радіанів у градуси
<code>math.radians(x)</code>	дійсне	дійсне	Перетворення градусів у радіани

<i>math.sinh(x)</i>	дійсне	дійсне	Гіперболічний синус
<i>math.cosh(x)</i>	дійсне	дійсне	Гіперболічний косинус
<i>math.tanh(x)</i>	дійсне	дійсне	Гіперболічний тангенс
<i>math.asinh(x)</i>	дійсне	дійсне	Обернений гіперболічний синус
<i>math.acosh(x)</i>	дійсне	дійсне	Обернений гіперболічний косинус
<i>math.atanh(x)</i>	дійсне	дійсне	Обернений гіперболічний тангенс
<i>math.pi</i>		дійсне	Число пі 3.14159265....
<i>math.e</i>		дійсне	Основа натурального логарифма 2.7182818...