

Введення числових даних

Для введення одного цілого числа, треба конвертувати дані з тексту за допомогою функції `int()` Приклад:

```
a=int(input())
```

Для введення одного дійсного числа в змінну, треба конвертувати дані з тексту за допомогою функції `float()` Приклад:

```
a=float(input())
```

Введення двох цілих чисел, записаних через пропуск у відповідні змінні:

```
a, b = map(int, input().split())
```

Введення двох дійсних чисел, записаних через пропуск у відповідні змінні:

```
a, b = map(float, input().split())
```

Цілі числа, що розділяються пропусками, можна ввести в список за допомогою list comprehension:

```
lst = [int(x) for x in input().split()]
```

Дійсні числа, що розділяються пропусками, можна ввести в список за допомогою list comprehension:

```
lst = [float(x) for x in input().split()]
```

Арифметичні операції:

$x + y$	Додавання
$x - y$	Віднімання
$x * y$	Множення
x / y	Ділення
$x // y$	Отримання цілої частки від ділення
$x \% y$	Залишок від ділення
$-x$	Зміна знаку числа
<code>abs(x)</code>	Модуль числа
<code>divmod(x, y)</code>	Пара ($x // y$, $x \% y$)
$x ** y$	Піднесення до степеня

Деякі функції роботи з числами знаходяться в окремих модулях. Для використання цих функцій, необхідно підключити модуль. Для роботи з числами частіше за все використовується модуль `math`

Прикладом такої функції є функція кореня `sqrt()` Програма, що обраховує $\sqrt{16}$:

або так:

```
import math  
print(math.sqrt(16))
```

або так:

```
from math import sqrt  
print(sqrt(16))
```

Результат: 4.0

Тут треба пояснити важливу деталь.

Можна написати і так:

```
from math import *  
print(sqrt(16))
```

Але тоді всі функції модуля `math` ми підключаємо до своєї програми, а це неправильно. Професіонали використовують лише ті функції, що їм потрібні. Тому краще уникати такого підходу.

Іноді в математичному модулі ховаються на перший погляд дивні речі, як то число π . Для того, щоб в тому розбиратися, варто набиратися досвіду в програмуванні і вчитися орієнтуватися в англomовному Python Help, що викликається натисканням F1.

Приклади:

```
import math  
print(math.pi)
```

Як ілюстрацію використання модулів, можна показати як в Python реалізований генератор випадкових чисел. В прикладі реалізовано генерацію випадкового числа в межах від нуля до одиниці включно:

```
import random  
print(random.random())
```

В Python є функція генерації цілого числа з проміжку:

```
import random  
n = random.randint(1,10)
```

В даному прикладі результат може приймати крайні значення, тобто $1 \leq n \leq 10$.

Можна при підключенні функції її перейменувати:

```
from random import randint as r  
n = r(1,10)
```

Конвертація числа в текст

```
a = 5  
b = str(a)
```

Конвертація тексту в число

```
a = '5'  
b = int(a)
```

Конвертація між десятковим і двійковим представленням числа

```
a = 5  
b = bin(a)  
print(b)
```

Результат:
'0b101'

```
a = '0b101'  
b = int(a,2)  
print(b)
```

Результат:
5

Робота з числами. Програма "Зоопарк"

Задача: В зоопарку живуть певна кількість кроликів і качок. Порахуйте, скільки у всіх тварин зоопарку лап.
Вхідні дані: в першому рядку вказується кількість кроликів, у другому – кількість качок. Всі числа – натуральні.

Приклад вхідних даних: 5 7	Приклад вихідних даних: 34
---	--------------------------------------

Варіант програми:

```
krolyky = int(input())  
kachky = int(input())  
print(krolyky * 4 + kachky * 2)
```