Введення даних

Введення значення одної змінної (текстової, числової)

Для введення тексту використовується оператор **input()** Результат зберігається у змінній. Приклад:

```
a = input()
b = input()
```

Часто в програмах вводяться текстові дані з непотрібними пробілами на початку або (і) в кінці. Для видалення пробільних символів на початку і в кінці рядка можна скористатися функцією strip() таким чином:

```
a = input().strip()
b = input().strip()
```

Якщо перед введенням даних необхідно вивести користувачу повідомлення, то синтаксис такий:

```
a = input("Введіть значення a ")
b = input("Введіть значення b ")
```

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

```
x = int(input())
```

Для введення дійсного числа, дані треба конвертувати з тексту в дійсне число за допомогою функції float() Приклад:

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

Введення значень декількох текстових змінних через пропуск або інший розділювач

Якщо необхідно дві частини тексту, що розділені пропуском ввести у відповідні змінні, то синтаксис наступний:

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

При введенні двох частин тексту через кому:

```
a, b  = input().split(',')
```

При необхідності введення значної кількості текстових даних, часто зручніше скористатися списком:

```
lst  = list(input().split())
```

Введення значень декількох числових змінних через пропуск або інший розділювач

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())
```

Введення даних за допомогою list comprehension

При необхідності введення значної кількості даних зручно скористатися «list comprehension», тобто конструкцією типу [i * 2 for i in my_list if i > 0]

Текстові дані, що розділяються пропусками, можна ввести в список таким чином:

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

Цілі числа, що розділяються пропусками, можна ввести в список таким чином:

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

Для дійсних чисел:

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

Виведення даних

Введення значень змінних

Стандартний запис:

```
print(a, b, sep = ' ', end= '\n')
```

де
sep = ' ' (розділювач аргументів, по замовчуванню — пропуск)
end = '\n' (кінець виводу, по замовчуванню — перевод рядка)

Виведення значення однієї змінної

```
print(a)
```

Виведення значення декількох змінних через пропуск:

```
print(a, b)
```

Виведення значення декількох змінних без пропуска між ними

```
print(a, b, sep = '')
```

Виведення значення декількох змінних з пропуском між ними і без переходу на новий рядок

```
print(a, b, sep = ' ', end = '')
```

Виведення списку за допомогою циклу:

Приклад, що ілюструє використання параметра «кінець виводу». В прикладі зліва простий вивід. Справа, завдяки вказанню параметра «кінець виводу», дані виводяться без переведення рядка.

lst = [1,2,5,12,44]
for i in lst:
  print (i)
 	
Результат
1
2
5
12
44
	lst = [1,2,5,12,44]
	for i in lst:
	  print (i, end = '')
	
	Результат
	1251244

Виведення списку за допомогою метода join:

Зручне виведення списку можна сформувати з використанням метода рядка join:

Виведення даних списку через пропуск якщо елементами списку є текстові дані:
```
print(' '.join(lst))
```

Виведення даних списку через пропуск якщо елементами списку є числа:
```
print(' '.join(map(str, lst)))
```

Варто звернути увагу, що метод join є інтелектуальним, після виведення останнього значення пропуск не ставиться.

Форматування виводу за допомогою оператора %

Приклад: Треба вивести дійсне число з двома знаками після коми:

```
a = 123.4567
print('%.2f' % a)
```

Результат:
123.46

Приклад: Вивести це саме число як десяткове, тобто обрізав все, що після коми:

```
a = 123.4567
print('%d' % a)
```

Результат:
123