

Тема 3

Введення та виведення даних

Знак рівності

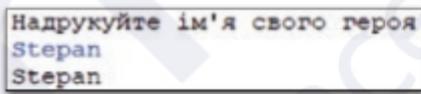
Знак рівності (=) в Python використовується не так, як у математиці. **SwimScore=70** означає, що змінній **SwimScore** присвоюється число 70. Знак рівності можна також використовувати для обчислення з правого боку від знаку рівності, а потім присвоїти результат змінній із лівого боку.

input()

Щоб отримати будь-яку інформацію від користувача, скористайтеся функцією **input()**.

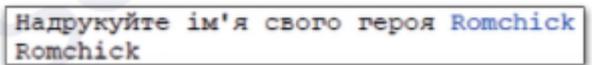
Наприклад, можна попросити користувача гри назвати ім'я свого спортсмена. У наведеній нижче програмі після її запуску система чекає, поки користувач введе деякий текст і натисне **Enter ↵**. Потім функція **input()** приймає введені дані та присвоює їх змінній **HeroName**.

```
print ("Надрукуйте ім'я свого героя ")
HeroName=input()
print(HeroName)
```



Функція **input()** також може використовувати рядок як повідомлення, що допоможе користувачеві зрозуміти, які саме дані потрібно ввести. Для нашого прикладу можна використати одну команду.

```
HeroName=input("Надрукуйте ім'я свого героя ")
print(HeroName)
```



Якщо ви хочете, щоб користувач вводив числа замість тексту, просто використовуйте команди **int(input())** для цілих чисел і **float(input())** для чисел із рухомою крапкою (дійсне число).

print()

Найпростіша команда, яку можна дати комп'ютеру, – вивести дані на екран. Наприклад, змінній **SwimScore** присвоєно число **70**.

```
SwimScore=70
print(SwimScore)
```

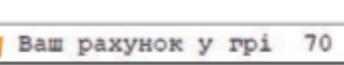


Запустіть програму. У такий спосіб можна вивести на екран будь-які повідомлення або дані, які вам потрібні. **print()** у Python – це функція, що виводить вказаний об'єкт на екран.

Якщо потрібно зробити інформацію, яку виводять на екран, зрозумілішою та зручнішою для використання, змініть код.

Ви бачите різницю?

```
SwimScore=70
print("Ваш рахунок у грі ", SwimScore)
```



Обчислення

Python також можна використовувати для виконання будь-яких обчислень: додавання, віднімання, множення, ділення тощо. Правила, які ви вивчили в Microsoft Excel для використання круглих дужок, застосовуються й тут. Припустимо, що ви хочете обчислити середній бал.

Арифметичні операції

Арифметичні оператори використовуються для виконання математичних операцій, як-от додавання, віднімання, множення тощо.

//	Цілочисельне ділення
%	Залишок від ділення
**	Піднесення до степеня

Оператори присвоювання

Для надання значень змінним використовуємо **оператори присвоювання**.

=	Присвоювання	a=5
+=	Додавання з присвоюванням	b += 7 → b = b + 7
-=	Віднімання з присвоюванням	b -= 1 → b = b - 1
*=	Множення з присвоюванням	c *= 2 → c = c * 2
/=	Ділення з присвоюванням	c /= 4 → c = c / 4
%=	Залишок від ділення з присвоюванням	d %= 5 → d = d % 5
**=	Піднесення до степеня з присвоюванням	d **= 3 → d = d ** 3

***** та **/** обчислюються перед **+** та **-**. Це означає, що $4 + 2 * 5$ дає в результаті **14**, а не **30**. Використовуйте **()**, щоб розставити пріоритети в послідовності обчислень.



Підключіть модуль `math` та використовуйте такі функції:

<code>sqrt()</code>	Квадратний корінь
<code>ceil()</code>	Найближче ціле (більше)
<code>floor()</code>	Найближче ціле (менше)
<code>factorial()</code>	Факторіал
<code>fmod(a,n)</code>	Остача від ділення числа a на число n
<code>max()</code>	Повертає найбільший елемент із двох чи більше аргументів
<code>min()</code>	Повертає найменший елемент із двох чи більше аргументів
<code>len()</code>	Повертає кількість символів рядка або числове значення
<code>abs()</code>	Повертає абсолютне значення числа
<code>round()</code>	Округлює число до вказаної кількості цифр
<code>pow(x,y)</code>	Повертає x у степені y

Робота з рядками

За допомогою `+` можна об'єднати два чи більше рядки, або створити більший рядок. Для цього використайте рядкові змінні або безпосередньо текст. Наприклад,

```
Sport="SprintRace"
Events="100 метрів"+Sport
```

У вас є дані:

```
Distance="100 метрів"
Sport="SprintRace"
```

Додайте наступні рядки:

```
Events=Distance+Sport
print(Events)
```

На екрані з'явиться:

```
100 метрівSprintRace
```

Але більш грамотний спосіб поєднання рядків буде такий:

```
Events=Distance+" "+Sport
```

```
100 метрів SprintRace
```

Не забувайте додавати `Space`, коли об'єднуєте текстові змінні, щоб вони були зрозумілими.

Коментарі

Використовуйте коментарі, щоб пам'ятати, що робить ваш код. Вони знадобляться й у майбутньому, якщо захочете внести зміни у вашу програму. Щоб додати коментар, використовуйте `#` на початку фрази. Все, що записане в рядку коментаря, комп'ютер проігнорує.

```
# Рахунок у грі
SwimScore=70
print("Ваш рахунок у грі ", SwimScore)
```

```
import math
r = 10
'''Це коментар
Він буде ігноруватися при виконанні програми.
Тут можна написати все що завгодно'''
long = 2 * math.pi * r
print(long)
```

```
62.83185307179586
```

Щоб додати багаторядковий коментар, створіть новий рядок і візьміть свій коментар у потрібні лапки. Такі лапки дозволяють об'єднати між собою групу текстових рядків.

Введені коментарі в редакторі коду виділяються іншим кольором. Це зроблено, щоб полегшити вам читання коду.

ПІДКАЗКА

Завжди використовуйте назви змінних, які легко запам'ятати. Наприклад, для підрахунку балів, набраних гравцем у стрибках у довжину, використовуйте таку змінну, як **LongJumpScore**.

Практичне завдання

Завантажте файл-заготовку зі сторінки інтернет-підтримки та виконайте запропоновані завдання.

