

Присвоєння

Загальний вигляд команди присвоювання такий:

<ім'я величини> <знак присвоювання> <значення або вираз>

Приклади:

```
a = 5
name = "Ukraine"
маса = 92.45
vyraz = 3*y-4
Мій_зріст = 152
t1 = -50
a = 2*45+12
```

Введення даних

Щоб змінній присвоїти значення, яке вводить користувач з клавіатури під час виконання програми використовують команду **input**. Запис команди input в тексті програми:

k = input()

для рядка символів: a = input()	введення з клавіатури рядка символів і присвоєння змінній k
для дробових(дійсних) чисел: a = float(input('Введіть x='))	введення з клавіатури цифр і символів, перетворення їх в дробове число (float) і присвоєння змінній x
для цілих чисел: a = int(input('Введіть y='))	введення з клавіатури цифр, перетворення їх в число (int) і присвоєння змінній y

Під час виконання проєкту при виконанні команди input() → блиматиме курсор і очікується введення з клавіатури символів: літер або чисел

Приклад:

```
input('Введіть своє ім'я: ')
```

ПОЯСНЕННЯ: виводиться текст-коментар **Введіть своє ім'я:** після якого блиматиме курсор і потрібно ввести символи. Для завершення введення натискають клавішу **Enter**

Виведення даних

```
print(a)
print('Значення a=',a)
print('Сума покупки',a,'грн.')
```

Приклад:

```
y = 5
print(y) → виведення: 5
print('Значення a=',a) → виведення: Значення a=5
print('Сума покупки',a,'грн')→ виведення: Сума покупки 5 грн
```

Приклад:

 Треба вивести **ДІЙСНЕ(ДРОБОВЕ)** число з двома знаками після коми:

```
a = 123.4567
print('%.2f' % a)
Результат: 123.46
```

Приклад:

 Вивести це саме число як десяткове, тобто 'обрізати' все, що після коми:

```
a = 123.4567
print('%d' % a)
Результат: 123
```

Виведення дійсних чисел

Для підвищення наочності даних, що виводяться програмою, можна задавати кількість десяткових знаків, які будуть записуватися після десяткової крапки, за допомогою методу **format()**:

```
"{:<загальна кількість знаків>.<кількість десяткових знаків>f}".format(number)
```

Приклад:

Проаналізуємо вигляд числа що виводиться при різних значеннях параметра <кількість десяткових знаків>. Загальну кількість знаків, що відводиться на виведення числа, можна не вказувати, тоді слід у фігурних дужках поставити крапку одразу після двокрапки.

```
number = 23.8589578
print("{:.3f}".format(number)) # 23.859
print("{:.4f}".format(number)) # 23.8590
```

Зміна типу даних (пояснення)

Часто виникає необхідність перейти від одного типу до іншого. Наприклад, значення, отримане від команди **input**, Python сприймає як рядок (тип **String**), слід передбачити перетворення уведеного значення до потрібного типу.

Для перетворення типів призначені функції:

- **float()** — створює нове дробове число з рядка або цілого числа.
- **int()** — створює нове ціле число з рядка або дробового числа.
- **str()** — створює новий рядок з числа.

Проаналізуємо приклади застосування функцій перетворення типів для різних аргументів:

Іноді виникає потреба перетворити дробове число на ціле.

Вираз	Результат
<code>int ('25')</code>	25
<code>int (4.9)</code>	4
<code>int ('4abc56')</code>	П о м и л к а
<code>str (25)</code>	'25'
<code>float (25)</code>	25.0

Перетворення цілого в десяткове

Візьмемо ціле число і за допомогою функції `float ()` створимо на його основі число з плаваючою точкою:

```
a = 12
```

```
b = float(a)
```

Результат: 12.0

Зверніть увагу на десяткову точку і завершальний 0 при виведенні змінної `b`. Саме за цими ознаками ми бачимо, що перед нами не ціле, а десяткове число.

Перетворення десяткового в ціле

Тепер спробуємо зробити зворотний перехід - візьмемо десяткове число і за допомогою функції `int ()` отримаємо з нього ціле:

```
c = 36.6
```

```
d = int(c)
```

$k = \text{round}(c)$

Результат: 36

Перетворення рядків у числа з плаваючою точкою

Отримати з рядка число можна наступним чином:

```
a = '36.6'
```

```
b = float(a)
```

Результат: змінна `b` має числове значення 36.6

Зверніть увагу, що змінна `a` виводиться в лапках. Так інтерпретатор Python показує, що це рядок. А для змінної `b` виводиться значення з плаваючою точкою.

Операції

Операція	Символ оператора	Приклад	Результат
Додавання	+	$a = 15 + 3$	$a = 18$
Віднімання	-	$b = 15 - 3$	$b = 12$
Множення	*	$c = 8 * 2$	$c = 16$
Ділення	/	$d = 7 / 2$	$d = 3.5$
Обчислення неповної частки від ділення	//	$e = 7 // 2$	$e = 3$
Обчислення остачі від ділення	%	$f = 7 \% 2$	$f = 1$
Піднесення до степеня	**	$j = 7 ** 2$	$j = 49$

Над цілими та дійсними числами, крім чотирьох основних операцій (додавання +, віднімання -, множення *, ділення /), можна виконувати такі операції:

- піднесення до степеня (**) # команда `print(27**4)` надрукує 6
- частка від ділення (//) # команда `print(27//4)` надрукує 6
- остача від ділення (%) # команда `print(27%4)` надрукує 3

$$\begin{array}{r|l}
 27 & 4 \\
 \hline
 24 & 6 \text{ — частка} \\
 \hline
 \text{остача} & 3
 \end{array}$$

Збільшення значення змінної

```

a=a+1
suma=suma+a

```

Умовний оператор (розгалуження)

Повне розгалуження:

```
if <логічний вираз>:  
    <послідовність команд 1>  
else:  
    <послідовність команд 2>
```

Неповне розгалуження:

```
if <логічний вираз>:  
    <послідовність команд>
```

Багатоальтернативне розгалуження:

```
if <логічний вираз>:  
    <послідовність команд 1>  
elif <логічний вираз 2>:  
    <послідовність команд 2>  
elif <логічний вираз 3>:  
    <послідовність команд 3>  
...  
else:  
    <послідовність команд N>
```

Важливо: Послідовності команд, виконання яких залежить від значень логічних виразів, записують з відступом вправо відносно першої літери слів if, elif та else.

Цикл з лічильником

Команда циклу

for *змінна* in *перелік значень*:

команди

лічильник *перелік значень*
for *chyslo* in [1,2,3,4,5]:

print(chyslo)

тіло циклу

for a in [1,2,3,4,5]:
 print(a, " ряд - ", a, " пакунки")

a=1
print(a, " ряд - ", a, " пакунки")
a=a+1
print(a, " ряд - ", a, " пакунки")
a=a+1
print(a, " ряд - ", a, " пакунки")
a=a+1
print(a, " ряд - ", a, " пакунки")
a=a+1
print(a, " ряд - ", a, " пакунки")
a=a+1
print(a, " ряд - ", a, " пакунки")

Шкала чисел (range)

range (n)

Шкала чисел від 0 до n-1

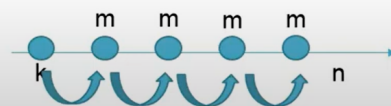
range (k, n)

Шкала чисел від k до n-1

range (k, n, m)

Шкала чисел від k до n-1 з кроком m (може бути від'ємним)

k – початкове число
n – кінцеве число
m – крок



Приклад:

range(1, 6) → згенерує 1, 1+1=2, 2+1=3 → 1,2,3,4,5
range(6) 0,1,2,3,4,5

range(3, 9, 2) → згенерує 3,5,7
range(1, 11, 1) → згенерує 1, 2,3,4,5,6,7,8,9,10
range(3, 9) 3,4,5,6,7,8

наступне число $7 + 2 = 9$ не увійде в набір чисел

```
for i in range(n):  
    <команди тіла циклу>
```

i - лічильник циклу який приймає значення 0, 1, 2, ...n-1
n - змінна відповідає за кількість повторень тіла циклу

Приклад:

```
for i in range(3, 9):
```

```
    s = s+i
```

```
    x = x+a
```

```
    d = d*g
```

Задача про пакунки

Початкова кількість = 0

Переглядаючи кожен ряд,

Додаємо до загальної кількості ті пакунки, які містяться у поточному ряді

