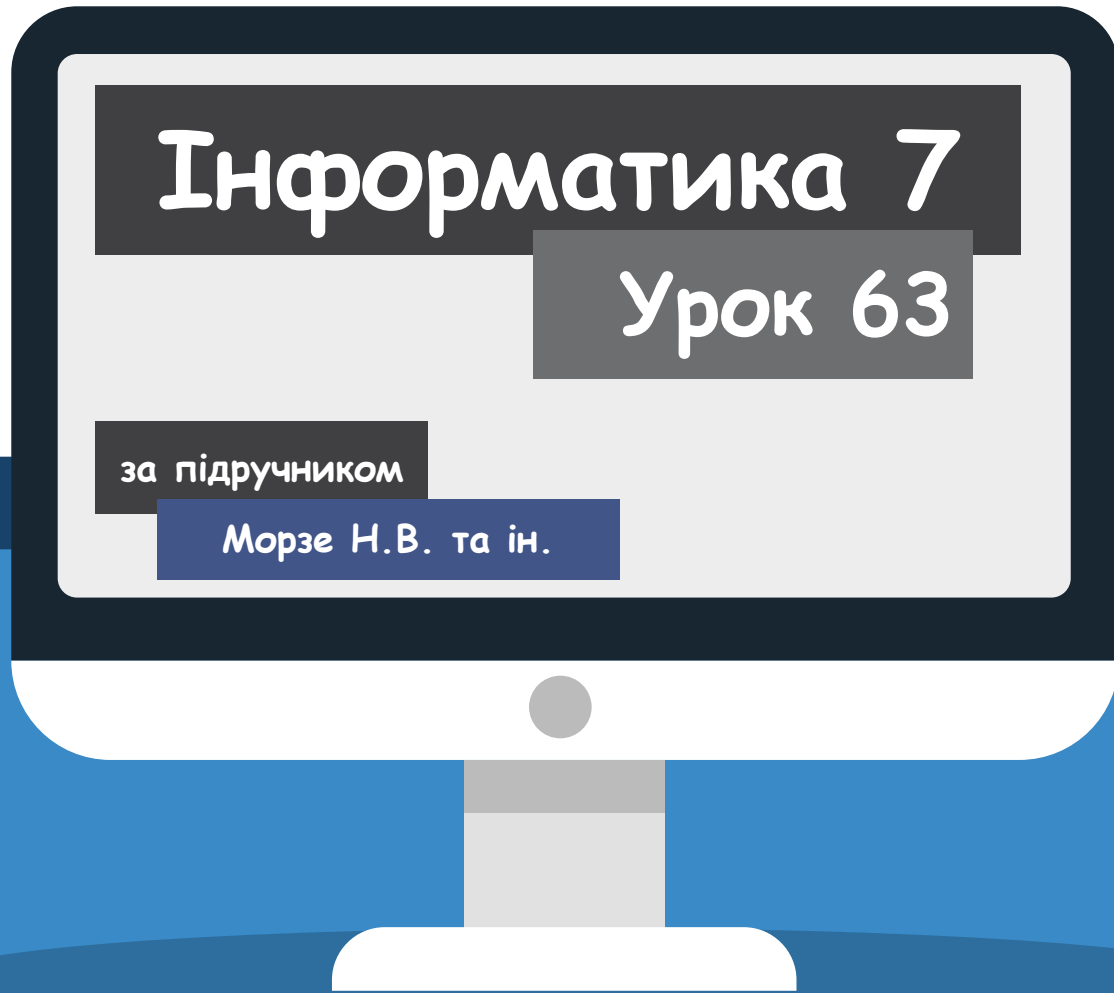


Порівняння значень величин



Нова українська школа



Ключове питання

- Чи можете ви запрограмувати вибір дій із деякого набору?

Завдання

- Спробуємо опанувати командами реалізації розгалужень, будемо продовжувати наповнювати наш сайт віртуальної школи програмування та створимо чат-бот у середовищі програмування.

Як порівнюють значення величин у програмах?

Ми вже розглядали програми, які реалізують **лінійні алгоритми**. Для опису умови в алгоритмах із **розгалуженням** використовують **логічні вирази** з використанням операцій порівняння.

Операція	Операція мовою Python	Приклад A=10	Результат
Більше	>	A>100	False
Менше	<	A<100	True
Не більше	<=	A<=10	True
Не менше	>=	A>=10	True
Дорівнює	==	A==100	False
Не дорівнює	!=	A!=100	True

Як порівнюють значення величин у програмах?

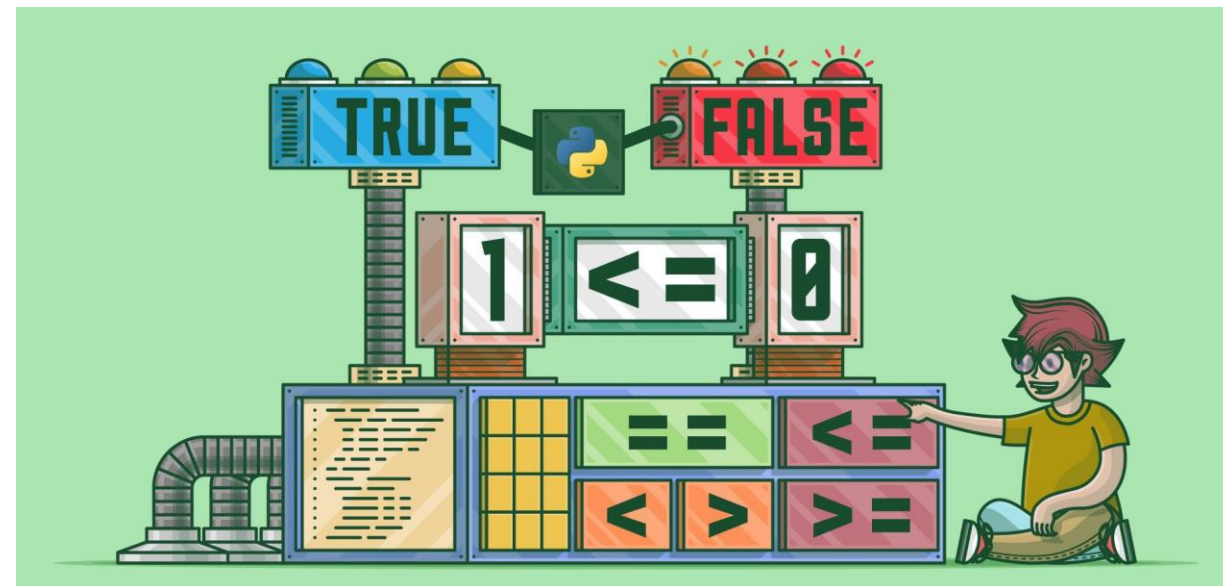
Результатом виконання операції порівняння значень двох величин є величина **логічного типу**, яка може набувати одного із двох значень:

True

або

False

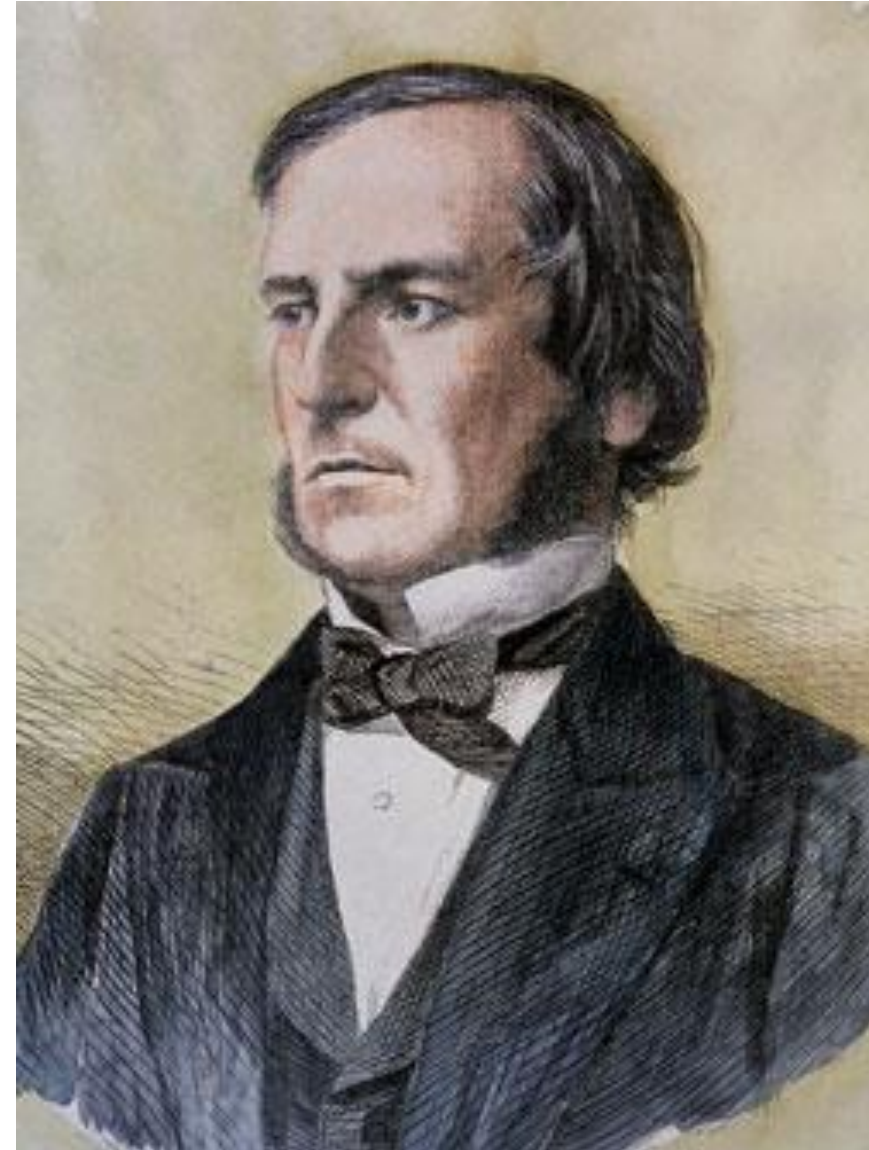
Для опису логічних величин у мові програмування Python використовують службове слово **bool**.



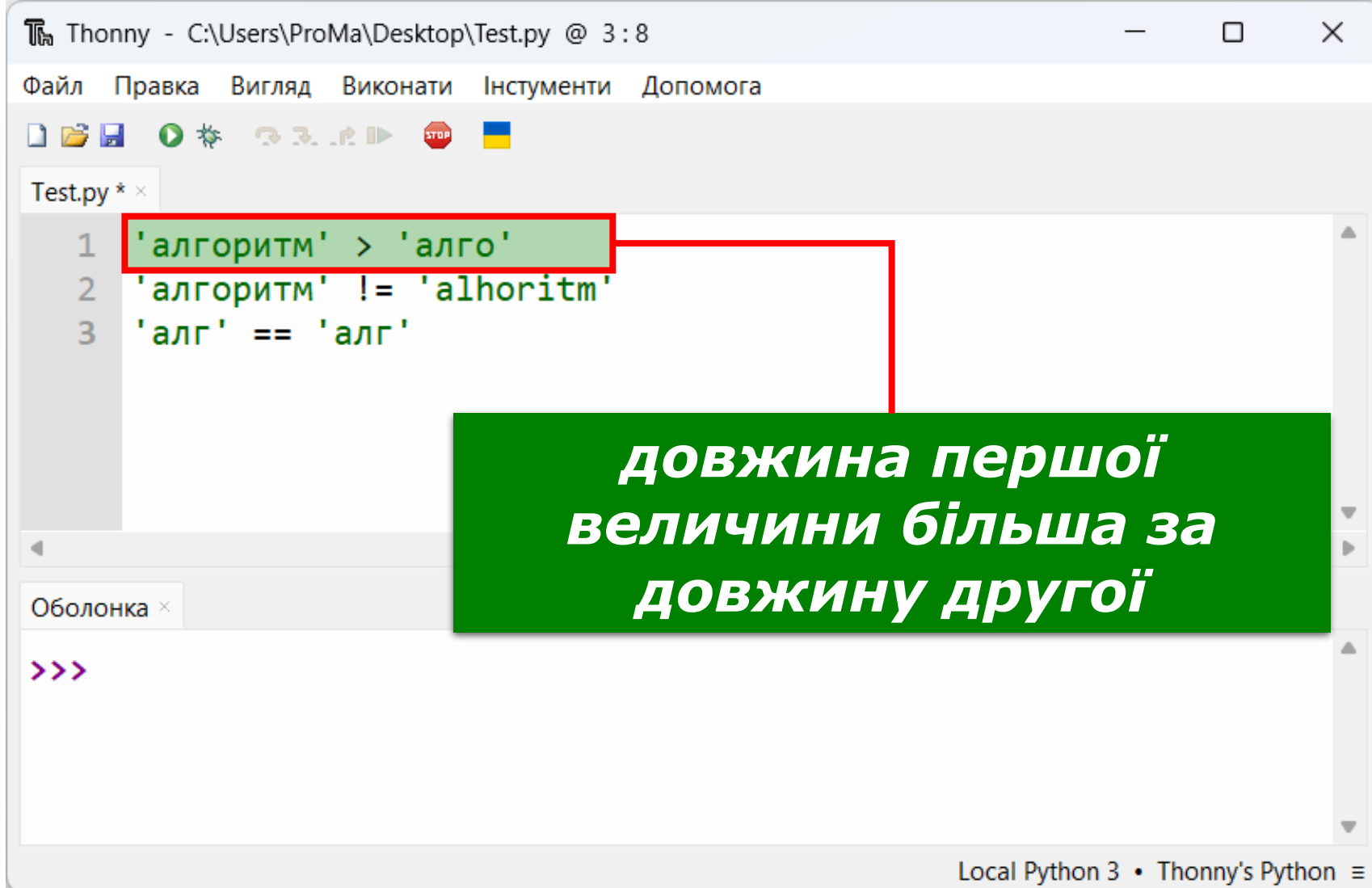
Як порівнюють значення величин у програмах?

Логічний тип отримав свою назву на честь англійського математика та логіка середини XIX століття, одного із засновників математичної логіки **Джорджа Буля.**

**Будинок
Джорджа
Буля у
Лінкольні**



Порівняння значень величин різних типів має особливості. Числові величини порівнюються за правилами математики. Порівняння текстових величин мовою *Python*, в результаті якого отримано логічне значення *True*.



```
Thonny - C:\Users\ProMa\Desktop\Test.py @ 3:8
Файл  Правка  Вигляд  Виконати  Інструменти  Допомога
Test.py * x
1 'алгоритм' > 'алго'
2 'алгоритм' != 'alhoritm'
3 'алг' == 'алг'
Оболонка x
>>>
```

довжина першої величини більша за довжину другої

Local Python 3 • Thonny's Python

Як порівнюють значення величин у програмах?

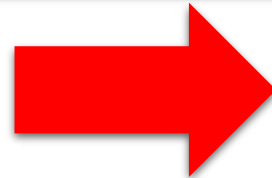
Для логічних величин:

a = True

b = False

Результатом виконання операції:

a > b



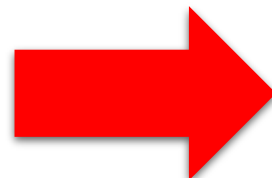
*Буде значення **True***

Оскільки:

*Істинне значення **True**
інтерпретується як **1***

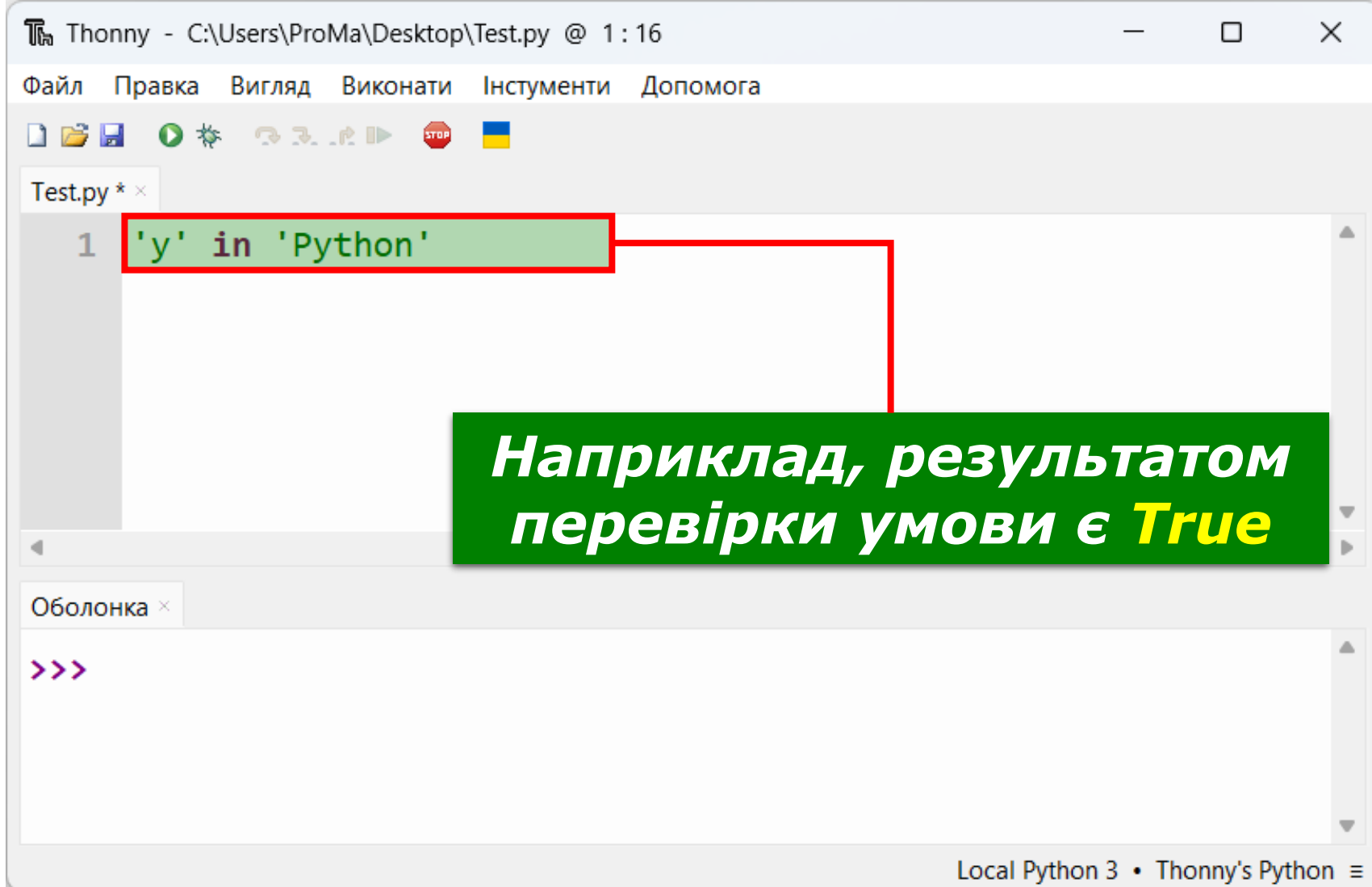
*Хибне значення **False**
інтерпретується як **0***

Зрозуміло, що



1 > 0

*Для того, щоб перевірити чи входить деякий фрагмент текстової величини в іншу текстову величину використовують оператор входження **in**.*



```
Thonny - C:\Users\ProMa\Desktop\Test.py @ 1:16
Файл  Правка  Вигляд  Виконати  Інструменти  Допомога
Test.py * x
1 'y' in 'Python'
Оболонка x
>>>
```

Наприклад, результатом перевірки умови є **True**

Local Python 3 • Thonny's Python

Які операції виконують над логічними величинами?

Окрім операцій порівняння над логічними величинами в усіх мовах програмування для запису складених висловлювань використовують логічні операції:

not (не)

заперечення;

and (і)

кон'юнкція;

or (або)

диз'юнкція (нестрога);

xor (виключне або)

диз'юнкція (строга).

Операцію **or** ще називають **логічним додаванням**, а **and** — **логічним множенням**. Назви кон'юнкція та диз'юнкція походять від англ. *conjunction* — об'єднання та *disjunction* — роз'єднання.

У математиці та логіці для запису логічних операцій прийняті спеціальні позначення.

Операцію **and**

позначають \wedge

Операцію **or**

позначають \vee

Які операції виконують над логічними величинами?

Результати виконання цих операцій над змінними **A** і **B** логічного типу, які набувають значень **True (1)** і **False (0)** наведені в **таблиці істинності**.

A	B	Not A	A and B	A or B	A xor B
1	1	0	1	1	0
1	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	0	0

Які операції виконують над логічними величинами?

З **таблиці істинності** видно, що за допомогою логічної операції **not** змінюють значення логічної величини на протилежне.

Результат операції **and** збігається з результатом множення значень **1 і 0** — є істинним лише за умови, що обидві величини **A і B** набувають значення **True** — істина.



Які операції виконують над логічними величинами?

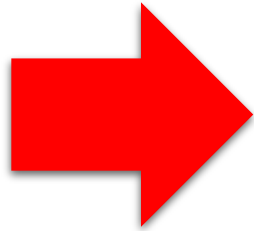
Результат операції **or** буде хибним лише за умови, що величини A і B набувають значення **False**, та істинним в усіх інших випадках. Наприклад, якщо $A = 5$, а $B = 7$, то значенням логічного виразу $(A < B)$ and $(B = 7)$ є **True**, оскільки логічні вирази $5 < 7$ та $B = 7$ набувають значення **True**, а тому, за таблицею істинності, результатом операції **and** є значення **True**.



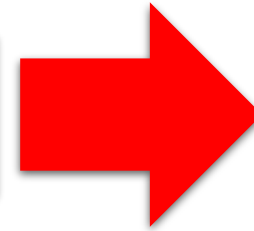
Які операції виконують над логічними величинами?

Щоб визначити значення складеного логічного виразу, який містить кілька логічних операцій, використовують таблицю істинності та враховують пріоритет виконання операцій: у першу чергу виконується операція

not



and



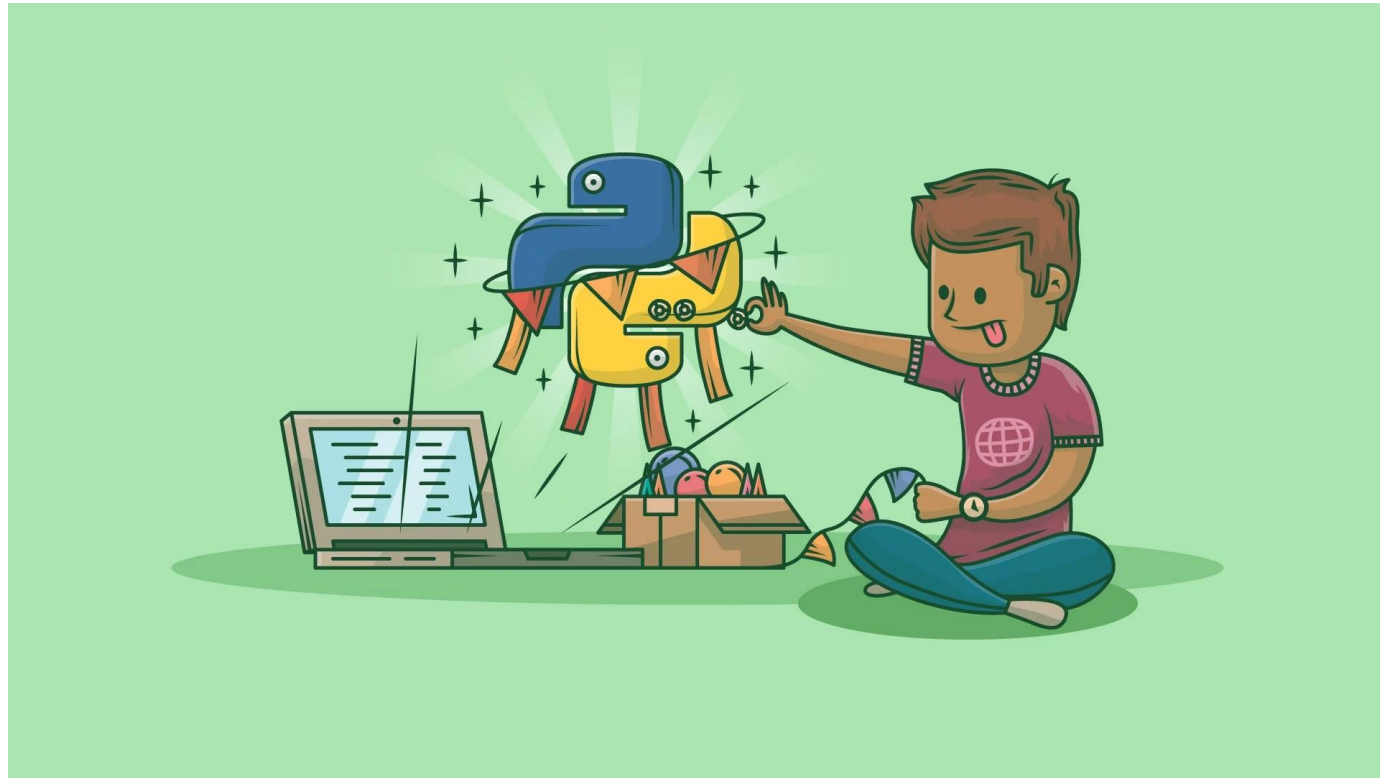
or

Як і для числових виразів, для зміни порядку виконання логічних операцій використовують дужки.

Як перетворити логічні дані в дані іншого типу?

У мові програмування **Python** передбачено, що логічні значення **True** та **False** можна перетворювати на значення інших типів.

Наприклад, при перетворенні їх у рядкові величини отримаємо слова **'True'** та **'False'** відповідно.



Як перетворити логічні дані в дані іншого типу?

Для перетворення логічних значень у значення числових типів використовують функції перетворення **int**, **float**. У цьому разі, як для цілих, так і для дійсних чисел:

True відповідає

1

False відповідає

0

`int(True) == 1`

`float(False) == 0.0`

`int(False) == 0`

`str(True) == 'True'`

`float(True) == 1.0`

`str(False) == 'False'`

Як перетворити логічні дані в дані іншого типу?

У зворотному порядку перетворення працює трохи по-іншому, але запам'ятати дуже просто:

*будь-які «непорожні»
значення
конвертуються в*

True

*будь-які «нульові»
значення
конвертуються в*

False

Як перетворити логічні дані в дані іншого типу?

*Для примусового приведення значення до логічного типу використовується вбудована функція **bool**.
Наприклад:*

```
bool(None) == False
```

```
bool(1) == True
```

```
bool(-1.1) == True
```

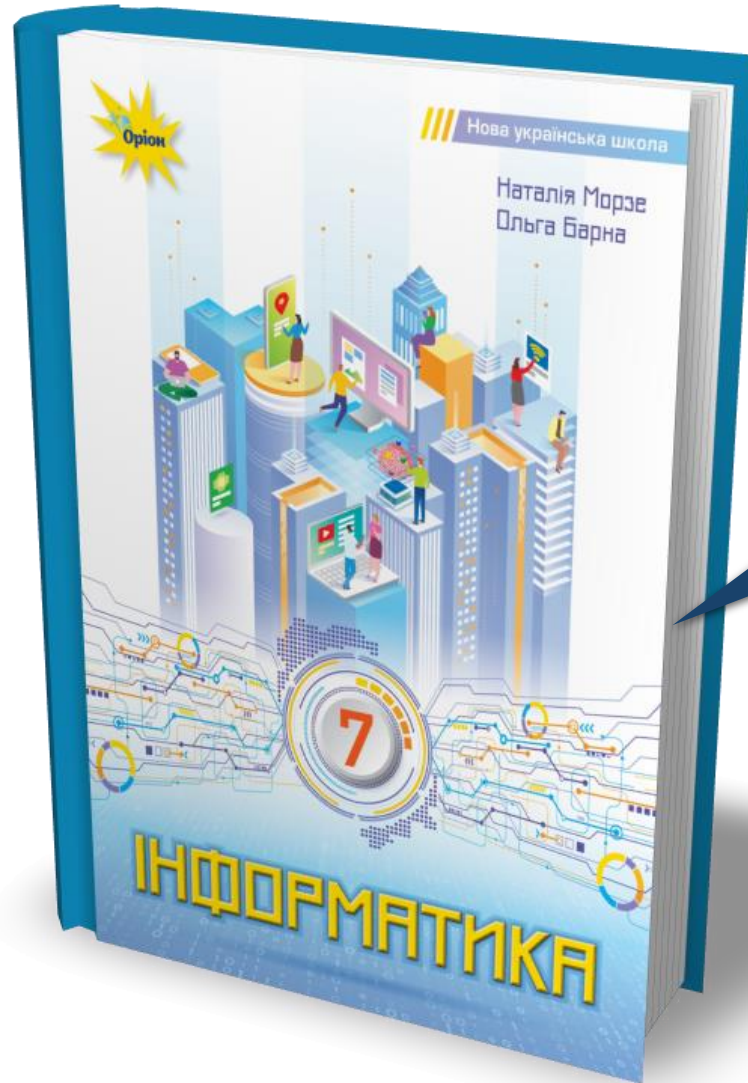
Розгадайте ребус



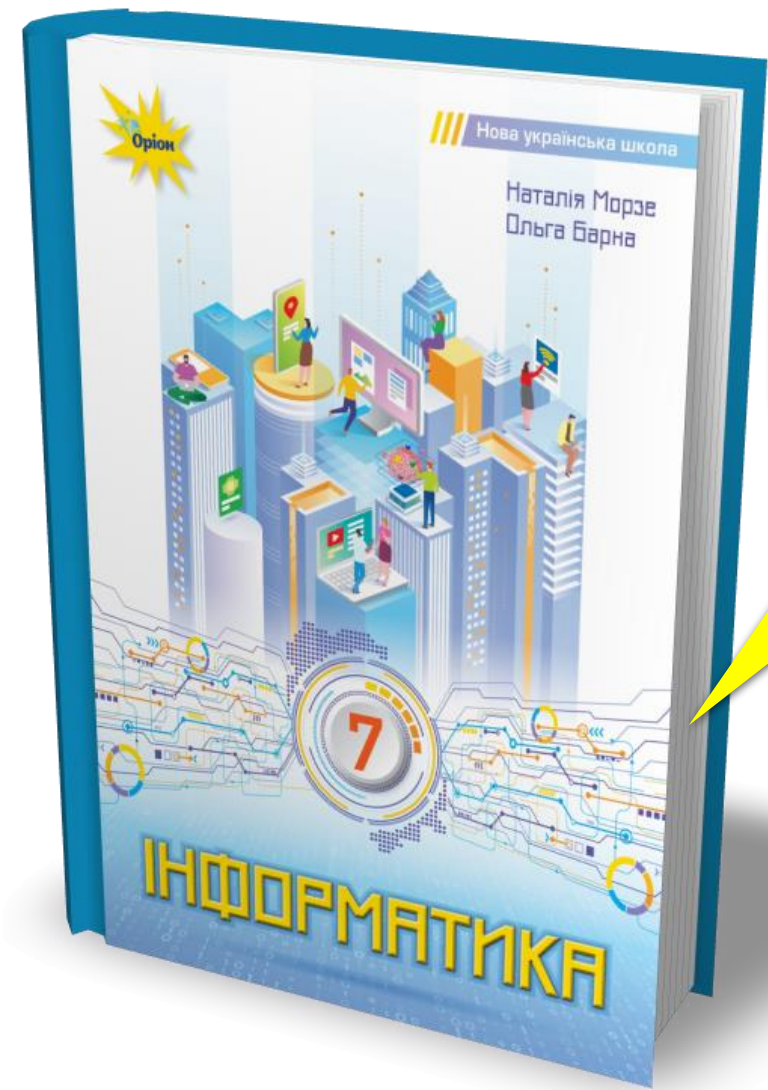
Порівняння



Домашнє завдання



***Проаналізувати
с. 233-236***



**Сторінка
234-236**



Інформатика 7

Урок 63

за підручником

Морзе Н.В. та ін.

Дякую за увагу!

Нова українська школа

