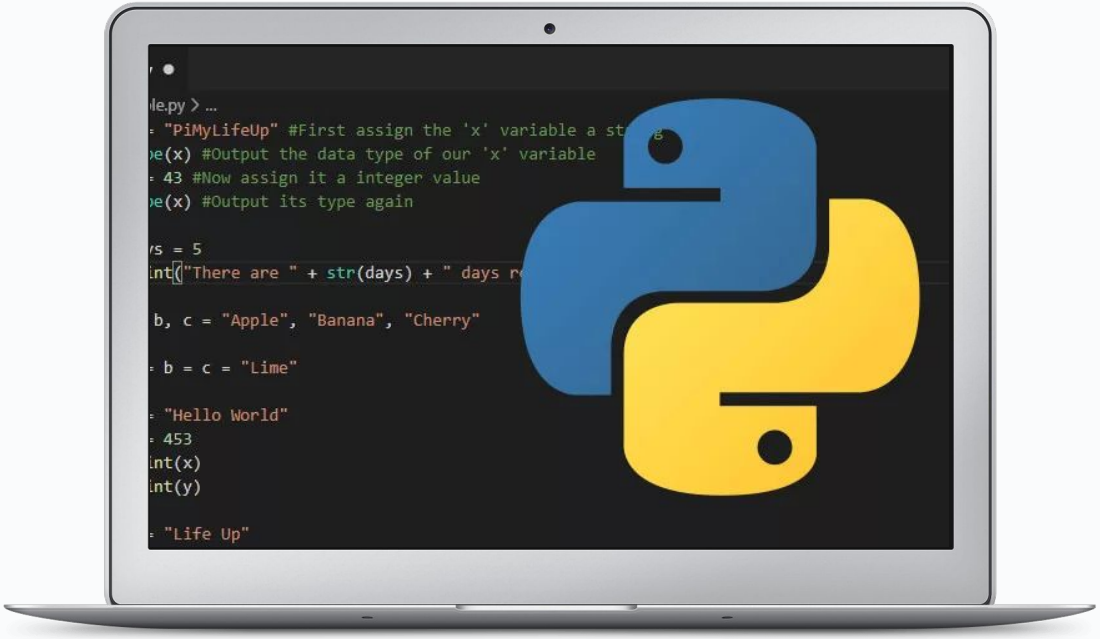
PROMETHEUS



Типи даних

Бутенко Сергій

План лекції

- **Тип даних як характеристика змінних**
- **Основні типи даних**
- **Катастрофічні помилки, пов’язані з типами даних**

Тип даних як характеристика змінних

Тип даних - це характеристика змінної, яка визначає:

- яких значень може набувати змінна;
- операції, які можна робити над цими даними;
- який обсяг пам’яті виділяється для збереження цієї змінної.

Основні типи даних та їх допустимі значення

Цілі – числа без десяткової крапки.

Приклад: -5, -21, 100500, 0, 259

Дійсні - числа, які мають десяткову крапку в своєму записі. Приклад: 9.8, 3.14, -99.99, 5.0, 0.0

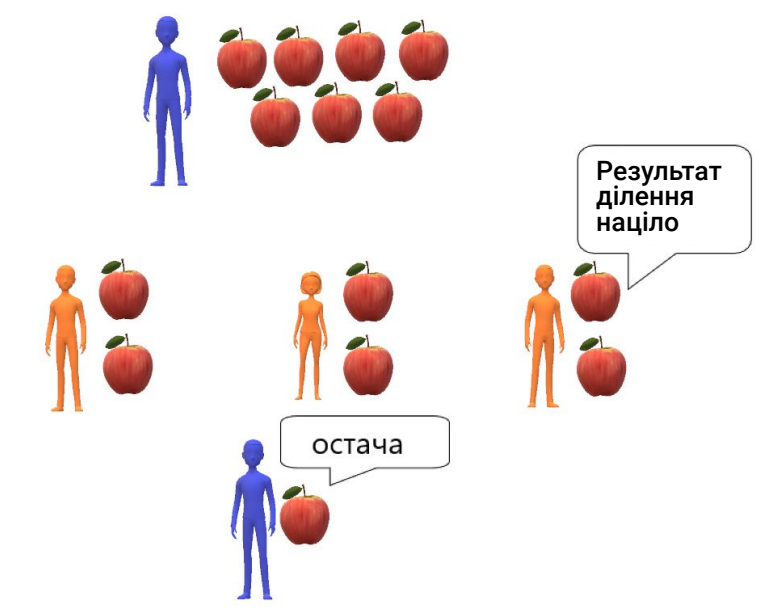
Символьний тип - будь-які символи або текст.

Приклад: “?”, “Prometheus”, “s”, “8”

Логічний тип. Він має лише два значення - True (Істина) і False (Хибність).

Наприклад, цих значень можуть набувати операції порівняння: при x=5 вираз x>0 набуває значення True, а при x=-3 вираз x>0 набуває значення False.

Допустимі операції над цілими числами



додавання + 1 + 3 = 4

віднімання - 1 - 7 = -6

множення * 5 * 3 = 15

ділення / 6 / 2 = 3

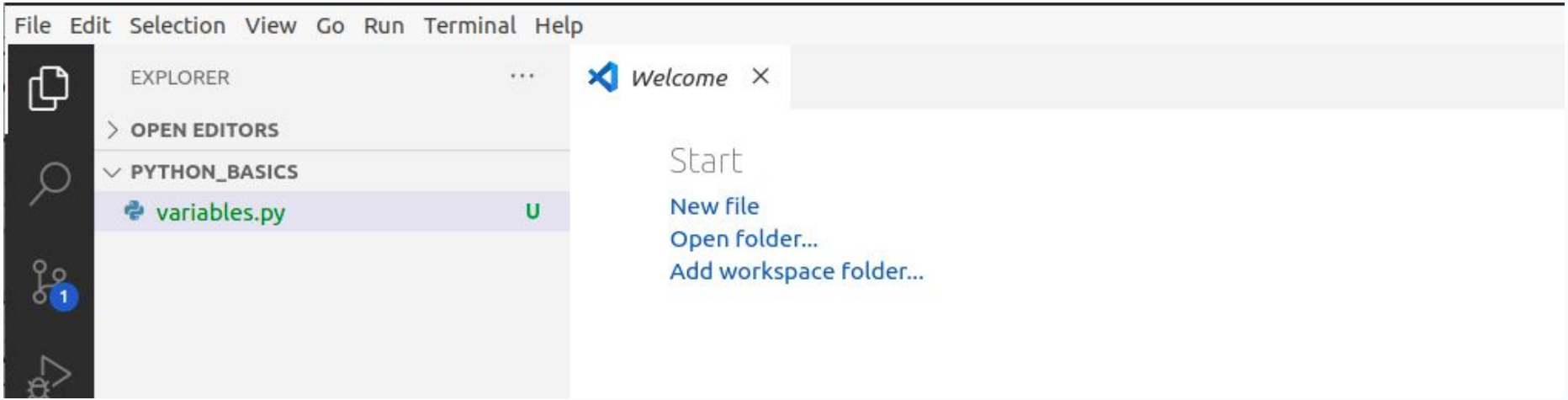
остача від ділення націло %

7 % 3 = 1

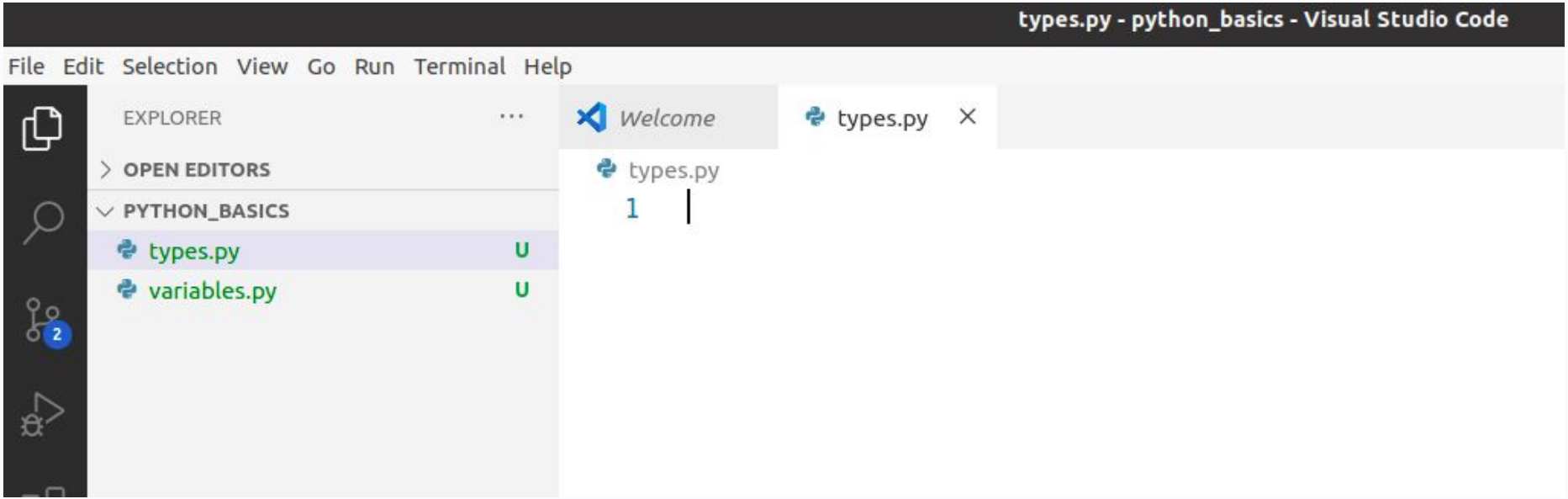
цілочисельне ділення //

7 // 3 = 2

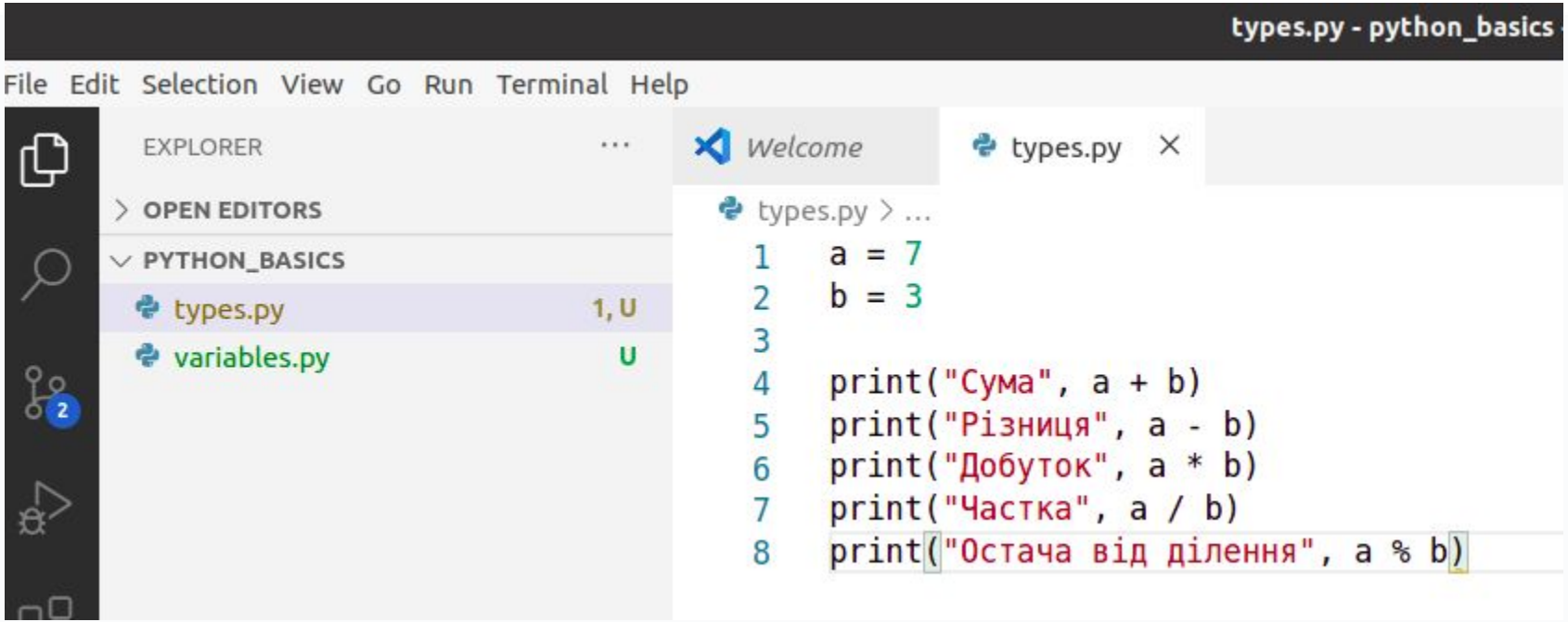
◎ Практика


File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
PYTHON_BASICS
variables.py
U
Welcome
Start
New file
Open folder...
Add workspace folder...

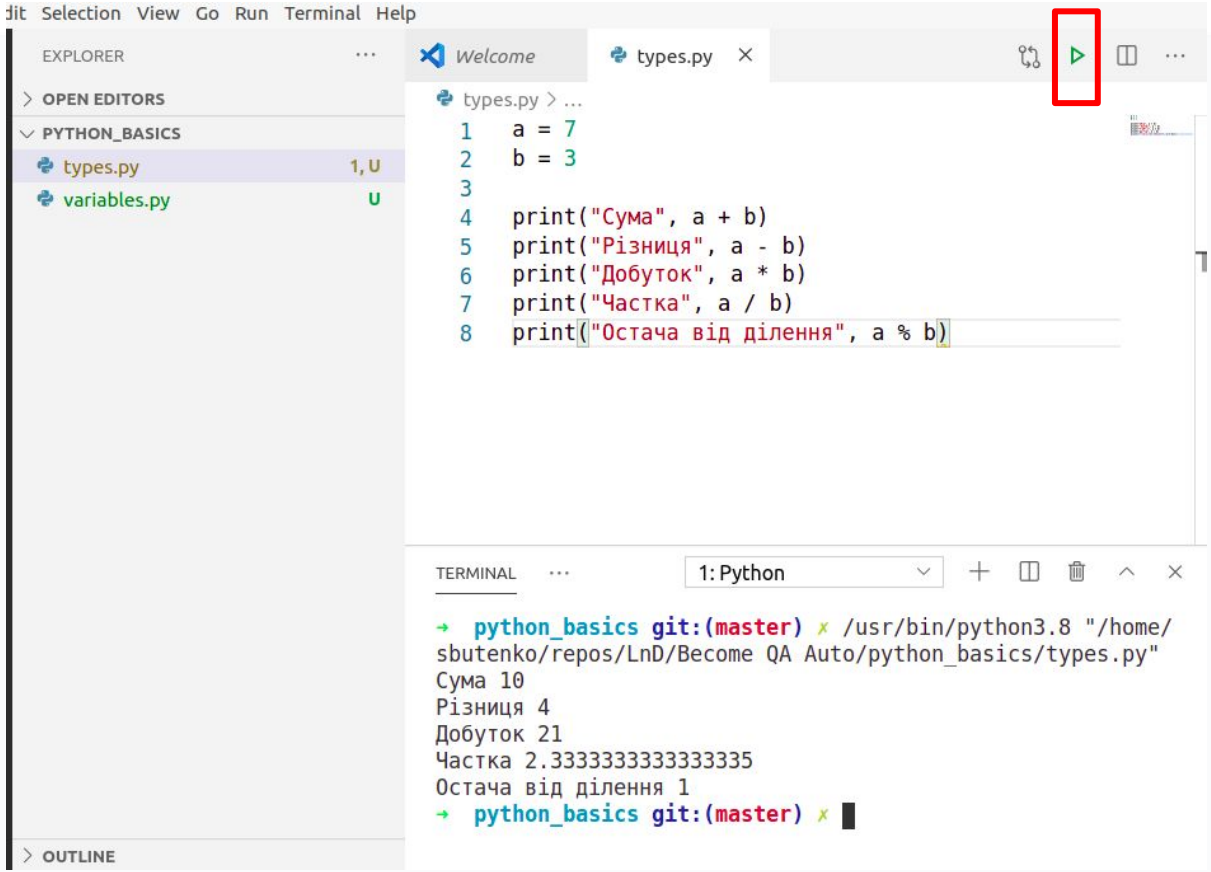
◎ Практика


types.py - python_basics - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
PYTHON_BASICS
types.py U
variables.py U
Welcome
types.py
types.py
1

Допустимі операції над цілими числами



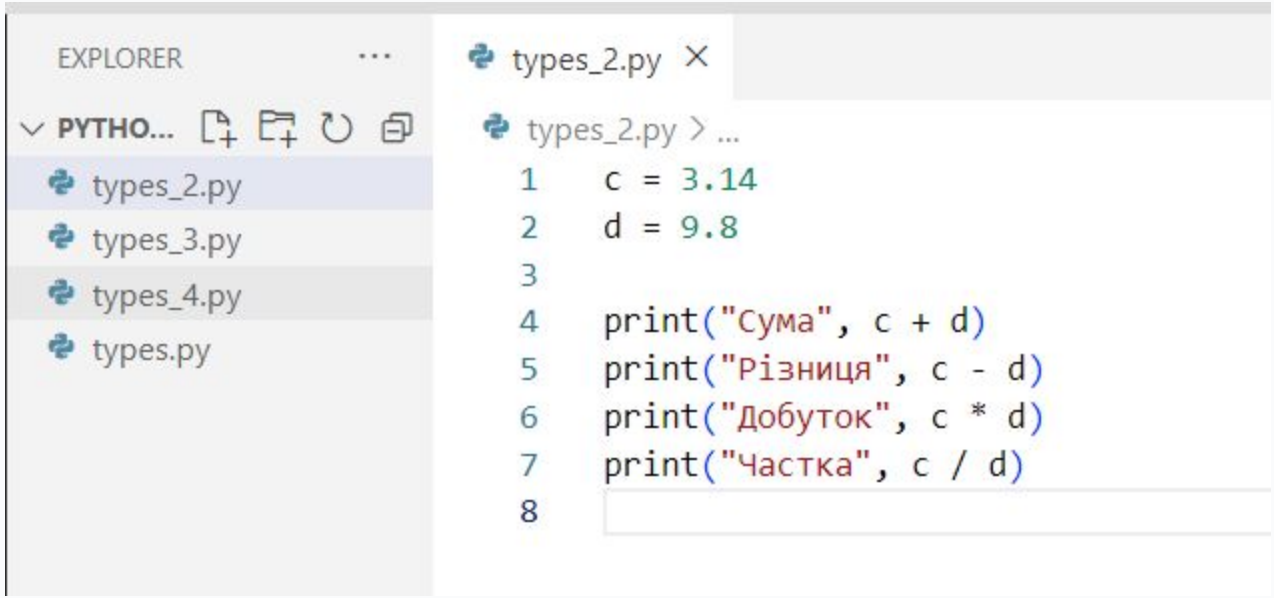
Допустимі операції над цілими числами



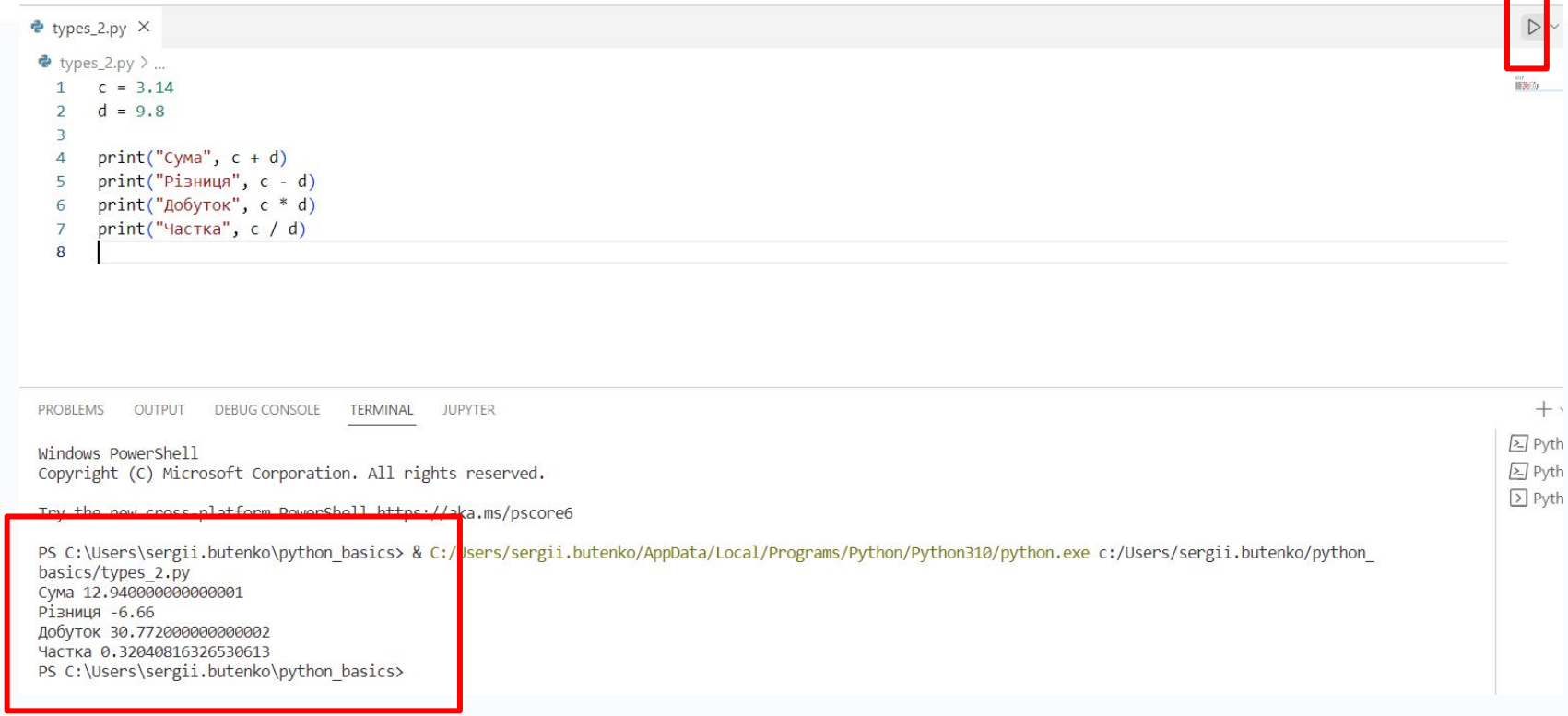
Допустимі операції над дійсними числами

додавання +	1 + 3.2 = 4.2
віднімання -	7 - 1,5 = 5,5
множення *	1.5 * 3 = 4.5
ділення /	6 / 1.5 = 4.0

Допустимі операції над дійсними числами



Допустимі операції над дійсними числами



Допустимі операції над символьними величинами

Конкатенація (склеювання) позначається +.

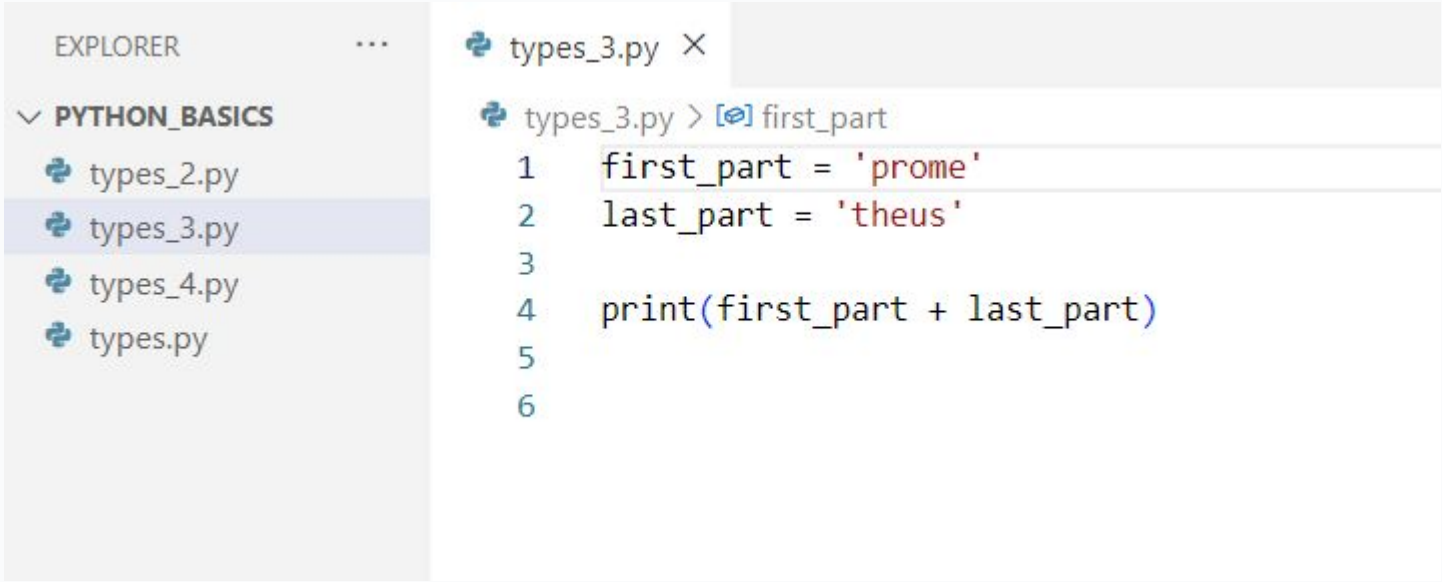
Результат: до кінця першого значення «доклеюється» початок другого

Приклади:

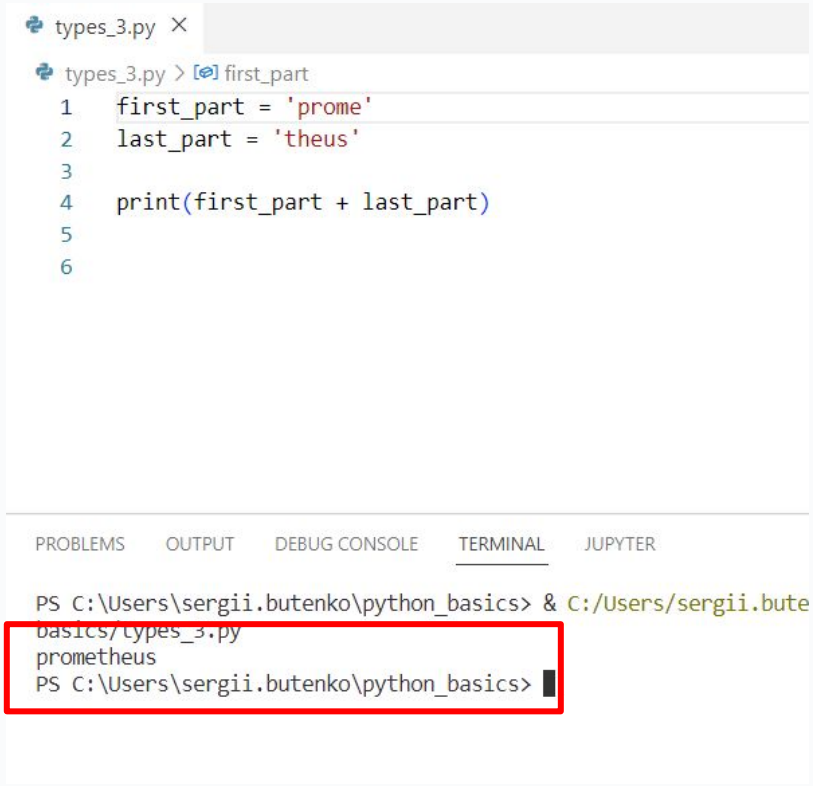
«шко» + «ла» = «школа»

«8» + «6» = «86»

Допустимі операції над символьними величинами



Допустимі операції над символьними величинами

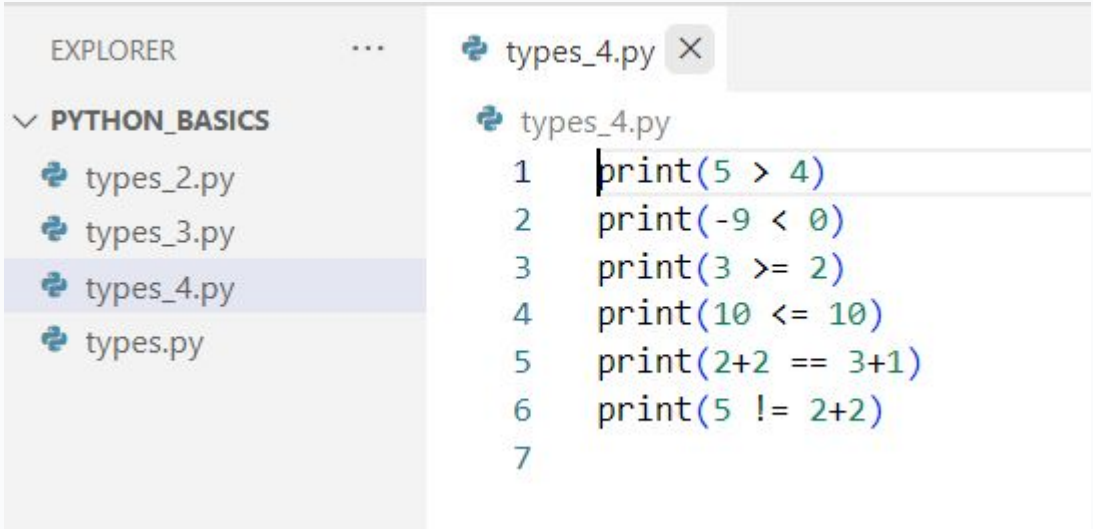


Логічний (булевий) тип

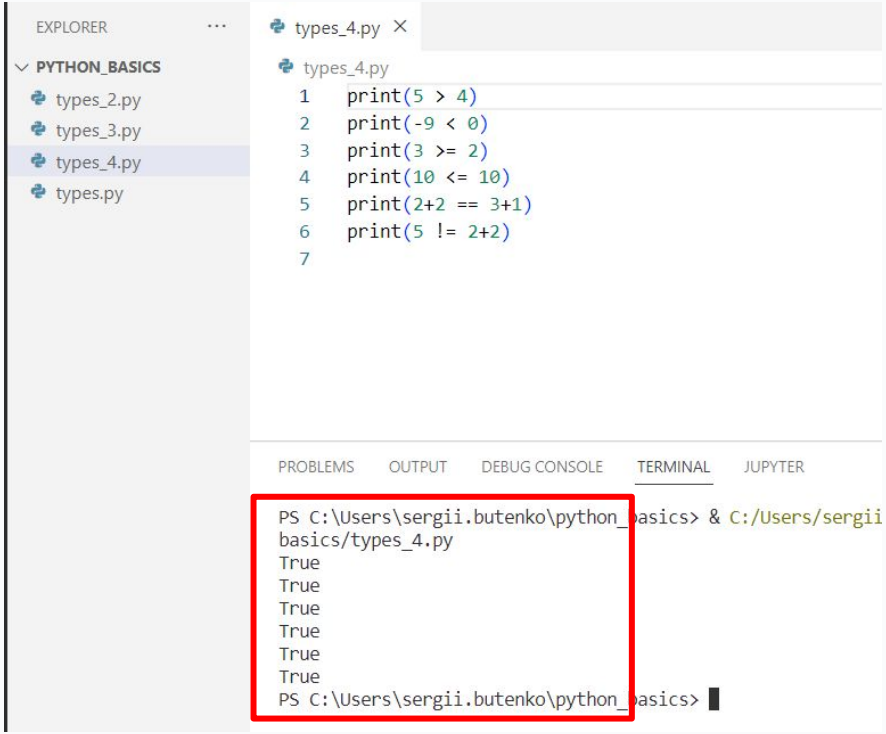
Операції порівняння:

більше	**>**	5 > 4
менше	**<**	-9 < 0
більше рівне	**>=**	3 >= 2
менше рівне	**<=**	10 <= 10
рівне	**==**	2+2 == 3+1
не рівне	**!=**	5 != 2+2

Логічний (булевий) тип



Логічний (булевий) тип



Катастрофа “Аріан-5”

- Помилка у програмі під час перетворення дійсного числа, яке зберігалося у 8 байтах пам’яті, у ціле, для збереження якого було виділено 2 байти пам’яті.
- Надто велике значення дійсного числа не вмістилося в 2 байтах, що спричинило переповнення.
- 4 червня 1996 року під час першого запуску на 34 секунді польоту ракета вибухнула

Документальні кадри катастрофи

https://youtu.be/gp_D8r-2hwk

Особливості типізації в Python

Явно-типізовані мови вимагають, щоб тип нових змінних, функцій і їх аргументів задавався явно.
Мови з **неявною** типізацією не передбачають вказання типів змінних, функцій та їх аргументів.
Наприклад:
int sum = 4 - явна типізація
count = 25.5 - неявна типізація

Сильна типізація - не дозволяє змішувати у виразах різні типи та не виконує автоматичні неявні перетворення. Мови зі **слабкою типізацією** виконують багато неявних перетворень автоматично.
Наприклад, сильна типізація забороняє таке:
a = “Число” + 2

Статична типізація визначається тим, що типи змінних і функцій встановлюються на етапі створення. В **динамічній типізації** усі типи даних з'ясовуються вже під час виконання програми.
Наприклад, динамічна типізація дозволяє таке:
count = 74
count = “Павло”

Мова Python - це мова, що підтримує динамічну, сильну і неявну типізацію:

- не вимагає вказання типів;
- типи даних визначаються присвоєним значенням;
- не можна змішувати у виразах різні типи.

Перетворення типів в Python

- **`int(str)`** – перетворити рядок на ціле число.
- **`float(str)`** – перетворити рядок на число з плаваючою комою.
- **`bool(val)`** – перетворити значення на логічне значення, **`true`** або **`false`** (якщо `val` **`-`** відсутнє, рівне 0 чи 0.0 то результат **`false,`** інакше **`- true)`**.
- **`str(val)`** – повертає рядкове представлення значення.

`a` = “Число” + `str`(2) - дозволяє сильна типізація. Результат: “Число2”

Використовуйте команду type(), щоб отримати тип значення змінної

Наприклад,

```
count = 74
type(count)          - поверне int
count = “Павло”
type(count)          - поверне str
```

Підсумки

Тип даних - це характеристика будь якої змінної, яка визначає:

- яких значень може набувати змінна;
- операції, які можна робити над цими даними;
- який обсяг пам’яті виділяється для збереження цієї величини.

Основними типами даних є:

- цілі числа,
- дійсні числа,
- символьний,
- булевий (логічний).