

Урок 2. Змінні. Присвоювання

Вивчення нового матеріалу

Слайд
№ 1

Скласти прості програми легко, ми в цьому переконалися.
Однак, потрібно навчитися вирішувати й складніші завдання.

Більшість програм взаємодіють із користувачами,
тобто користувачі вводять до них дані і ці дані
потрібно десь зберігати.

Їх зберігають у **змінних**.



Згадай, що таке змінна в математиці.

Слайд
№ 2

Будь-яка **змінна величина** має ім'я та значення.



Як ти думаєш, де в реальному житті використовуються змінні?

Далі

Слайд
№ 3

Змінну можна порівняти з чашкою, в яку наливають каву, чай, сік та інше, тобто наша змінна буде мати ім'я **чашка**, а значення залежатиме від її вмісту.



Одна та сама змінна може набувати різних значень.

Слайд
№ 4

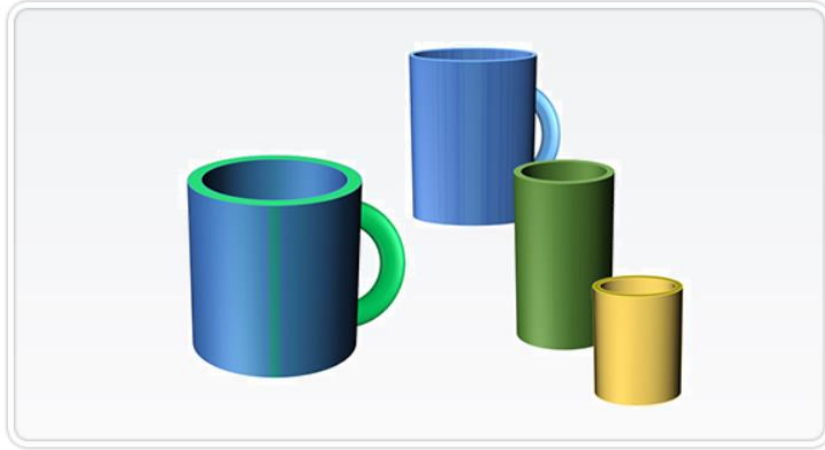
Сам процес надання змінній значення називається **присвоюванням**.



Присвоювання можна порівняти з наливанням в чашку (змінну) рідини (значення).

Часто для розв'язання задачі замало однієї змінної.

Тоді використовують змінні з різними іменами і присвоюють їм різні значення.



Як ти думаєш, скільки змінних може бути в програмі?

У Python присвоєння позначається символом `=`.

ім'я змінної = значення

Розглянемо приклади.

```
x = 5      # змінній x присвоїли значення 5
y = x      # змінній y присвоїли значення змінної x
x = y+1    # змінній x присвоїли значення, що на 1 більше,
           # ніж те, яке було в змінній y
```



Змінній можна присвоювати число, значення іншої змінної тощо.

Слайд
№ 7

У результаті виконання двох верхніх фрагментів коду значення змінних стануть однаковими, а в нижньому фрагменті змінні обмінюються значеннями.

$x = y$
 $y = x$

	x	y
спочатку	1	2
$x = y$	2	2
$y = x$	2	2

$y = x$
 $x = y$

	x	y
спочатку	1	2
$y = x$	1	1
$x = y$	1	1

$z = x$
 $x = y$
 $y = z$

	x	y	z
спочатку	1	2	
$z = x$	1	2	1
$x = y$	2	2	1
$y = z$	2	1	



Переглянь уважно таблиці, у яких на прикладі $x=1$, $y=2$ показано, як значення змінюються після виконання кожної команди.

Слайд
№ 8

Як ми вже бачили, змінній можна присвоїти значення **виразу**, що може містити числа, а також імена цієї або інших змінних, з'єднані знаками арифметичних операцій.

У такий спосіб можна збільшувати або зменшувати значення змінних!
Наприклад, так ми збільшуємо значення змінної **x** на **1**:

$x = x + 1$

Нове значення змінної

Старе значення змінної



Як ти думаєш, яку операцію використовують, щоб збільшити значення змінної у кілька разів?

Слайд
№ 9

Тепер розв'яжемо практичну задачу з використанням змінної!

Удосконалимо програму **Hello, world**, яку ми склали на попередньому уроці.

Припустимо, необхідно, щоб програма віталася не з усіма, а з конкретним користувачем, спитавши його ім'я, тобто виводила написи на кшталт

Привіт, Андрію!
Привіт, Оленко!

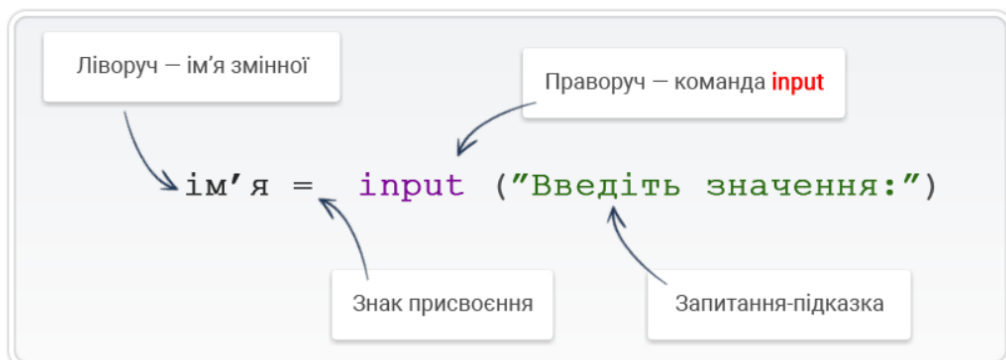


Поясни, яким чином, на твою думку, можна це реалізувати.

Слайд
№ 10

Щоб комп'ютер міг «поговорити» з користувачем, використовують команду **input**.

Виконуючи цю команду, програма відобразить текст-запрошення (наприклад, **Введіть значення:**), очікуватиме введення даних від користувача, а потім присвоїть їх змінній.



Команду **input** треба записати в правій частині присвоєння, а в лівій — ім'я змінної, якій присвоїться значення.

А за допомогою команди **print** можна вивести значення змінної:

```
x = 5  
print (x)
```

Відобразиться **5**

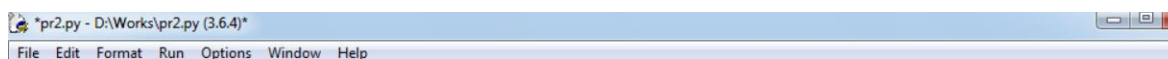
Або кілька значень:

```
x = 5  
y = 7  
print (x, y)
```

Відобразиться **57**



Команду **input** треба записати в правій частині присвоєння, а в лівій — ім'я змінної, якій присвоїться значення.



```
name=input ("Як тебе звати?")
```

Ім'я змінної — **name**

Запитання-підказка

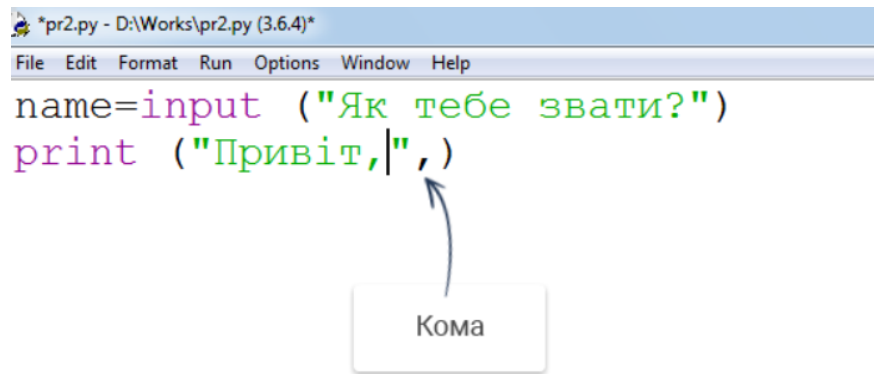
Продемонструємо
створення програми,
що вітається з
користувачем.

Ця команда відобразить рядок **Як тебе звати?**, зачекає введення даних від користувача і присвоїть їх змінній **name**.



Створюємо новий файл і набираємо в ньому команду введення даних і присвоєння їх змінній **name**.

Слайд
№ 13



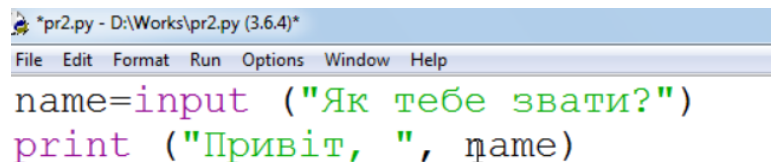
```
*pr2.py - D:\Works\pr2.py (3.6.4)*
File Edit Format Run Options Window Help
name=input ("Як тебе звати?")
print ("Привіт,|",)
```

Кома



У наступному рядку набираємо команду **print**.

Слайд
№ 14
Вправа
№ 1 -
демо



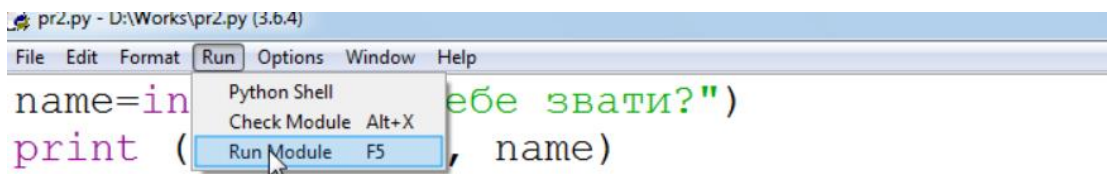
```
*pr2.py - D:\Works\pr2.py (3.6.4)*
File Edit Format Run Options Window Help
name=input ("Як тебе звати?")
print ("Привіт, ", name)
```



Після коми вказуємо ім'я змінної **name**.

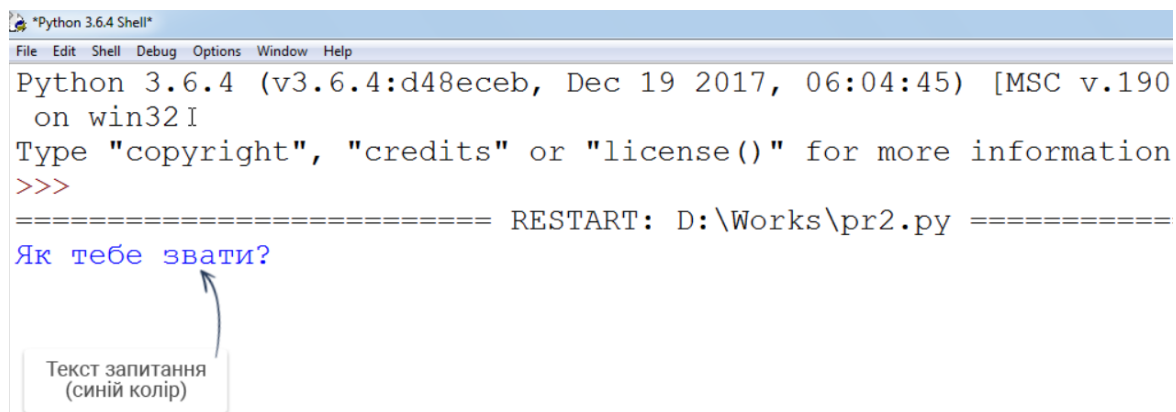
Натиснути **Enter**.

Слайд
№ 15



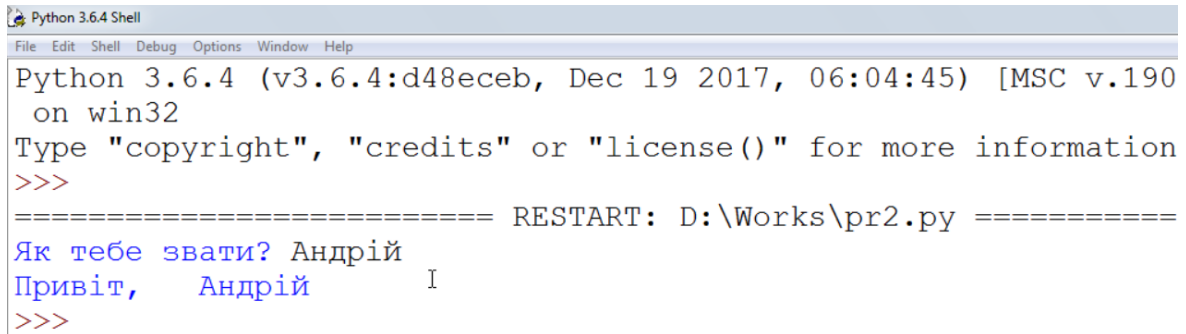
Запускаємо програму на виконання. Клацни пункт **Run Module**.

Слайд
№ 16



Автоматично відкривається вікно програми і відображається текст запитання. Поряд користувач вводить відповідь. Наприклад, **Андрій**.

Слайд
№ 17



Python 3.6.4 Shell

File Edit Shell Debug Options Window Help

Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.190 on win32

Type "copyright", "credits" or "license()" for more information

>>>

===== RESTART: D:\Works\pr2.py =====

Як тебе звати? Андрій

Привіт, Андрій

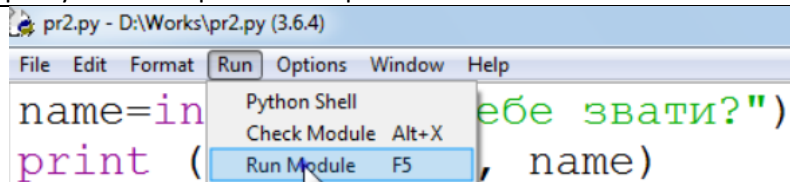
>>>



Після натискання клавіші **Enter** з'являється текст **Привіт, Андрій.**

Відповідь користувача — чорним кольором.

Слайд
№ 18



pr2.py - D:\Works\pr2.py (3.6.4)

File Edit Format Run Options Window Help

name=input("Як тебе звати?")

print (name)

Python Shell

Check Module Alt+X

Run Module F5



Повторно запускаємо програму на виконання.

Слайд
№ 19

```
Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.190
on win32
Type "copyright", "credits" or "license()" for more information
>>>
===== RESTART: D:\Works\pr2.py =====
Як тебе звати? Андрій
Привіт, Андрій
>>>
===== RESTART: D:\Works\pr2.py =====
Як тебе звати? Оленка
Привіт, Оленка
>>>
```



Після натискання клавіші **Enter** з'явиться текст **Привіт, Оленка**.

Слайд
№ 20

Змінна **name** в перший раз отримала значення "Андрій", а вдруге – "Оленка".
Текст, записаний у лапках, відображається без змін.

Код програми

```
*pr2.py - D:\Works\pr2.py (3.6.4)*
File Edit Format Run Options Window Help
name=input ("Як тебе звати?")
print ("Привіт, ", name)
Ln: 5 Col: 0
```

Результат виконання

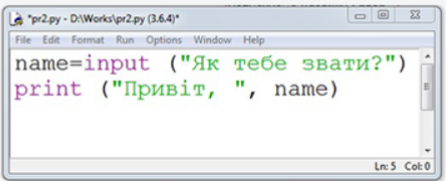
```
===== RESTART:
Як тебе звати? Андрій
Привіт, Андрій
>>>
===== RESTART:
Як тебе звати? Оленка
Привіт, Оленка
>>>
```



Порівняй код програми і результати її виконання.

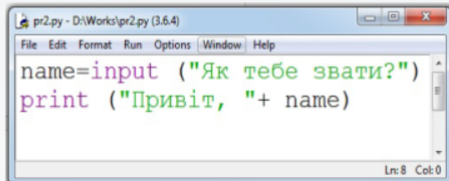
В операторі **print** замість коми можна ставити знак **+**.
Результат виконання буде таким самим.

Варіант 1



```
name=input("Як тебе звати?")
print("Привіт, ", name)
```

Варіант 2



```
name=input("Як тебе звати?")
print("Привіт, "+ name)
```



Операція «**+**» утворює з двох рядків один, а знак «**,**» означає «вивести після першого рядка другий».

Вправа

Вправа № 1



Вправа 1. Тепер розв'яжи задачу самостійно.

Задача

Створити програму, що вітатиметься з користувачем, запитавши його ім'я.
Рядок виведення сформувати через «**+**».
Передбачити варіанти виведення, як на зразку.

Зразок результату роботи програми

Як тебе звати? Ольга
Привіт, Ольга

Як тебе звати? Ольга
Привіт, Ольга!

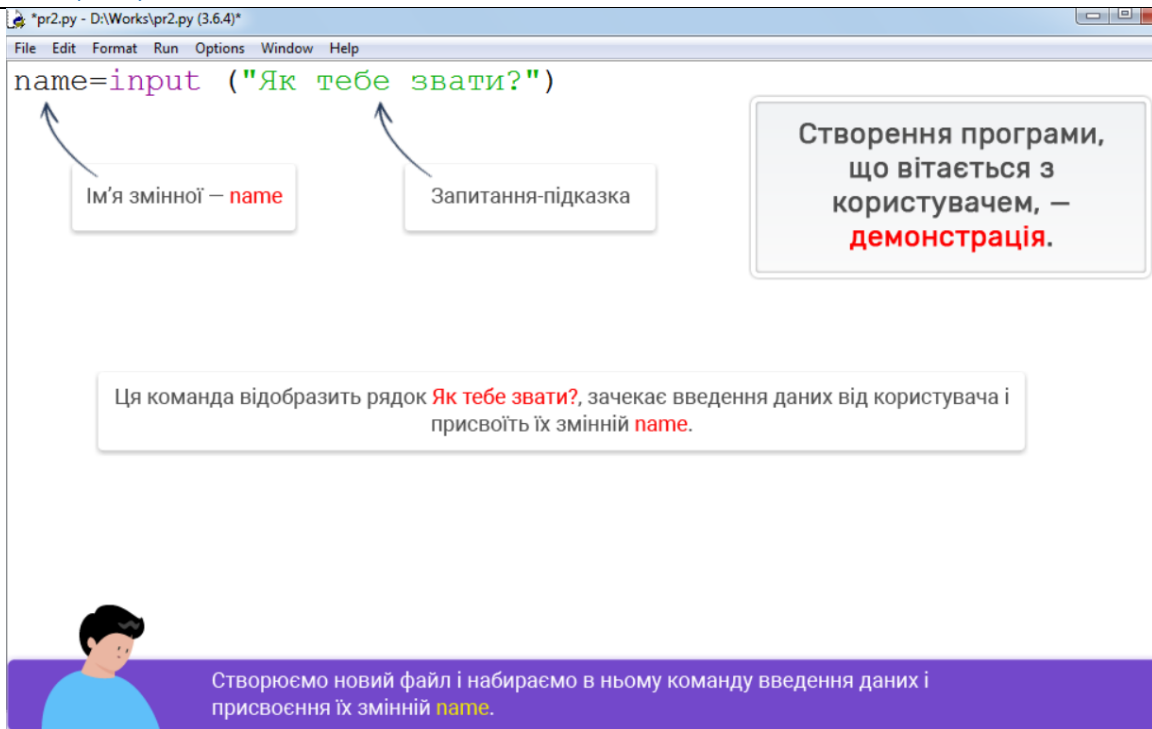
Варіант 1

Варіант 2



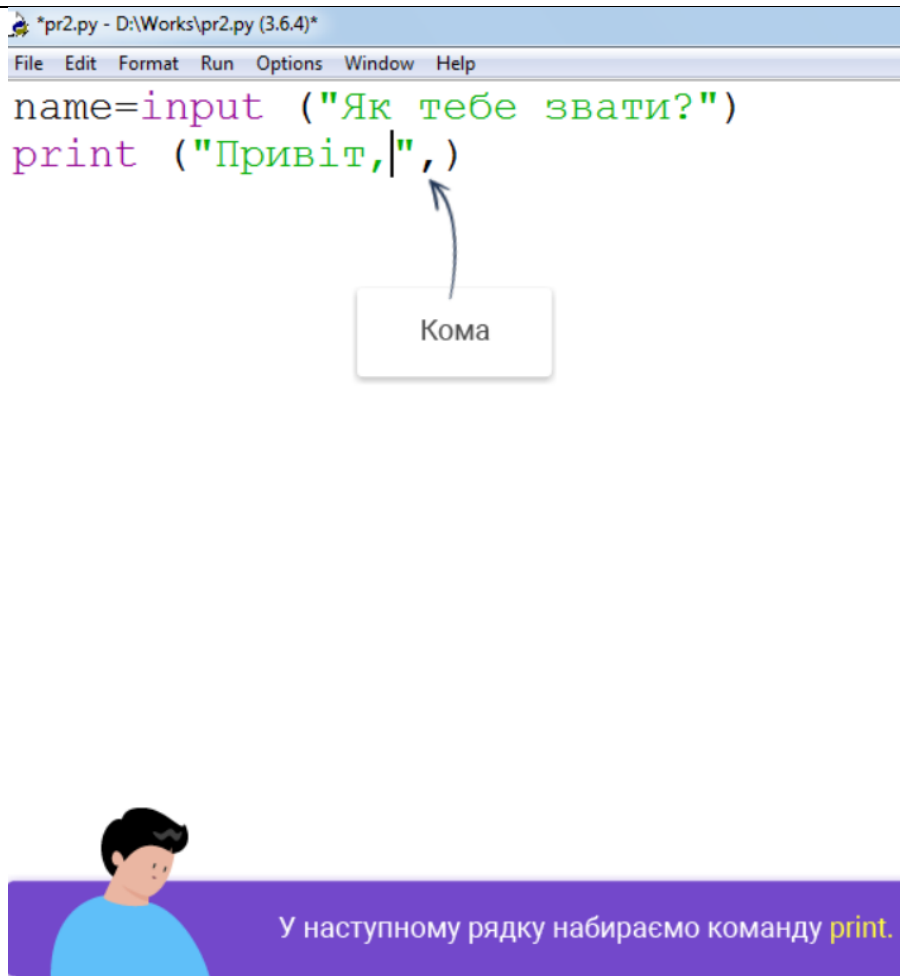
Створи у файлі програму, збережи та запусти файл.

Слайд № 22



The screenshot shows a Python IDE window titled "*pr2.py - D:\Works\pr2.py (3.6.4)*". The code is `name=input ("Як тебе звати?")`. Two callouts point to parts of the code: one to `name` with the text "Ім'я змінної – name" and another to the string `"Як тебе звати?"` with the text "Запитання-підказка". A larger callout on the right says "Створення програми, що вітається з користувачем, – демонстрація." Below the code, a text box explains: "Ця команда відобразить рядок Як тебе звати?, зачекає введення даних від користувача і присвоїть їх змінній name." At the bottom, a purple bar with a person icon contains the text: "Створюємо новий файл і набираємо в ньому команду введення даних і присвоєння їх змінній name."

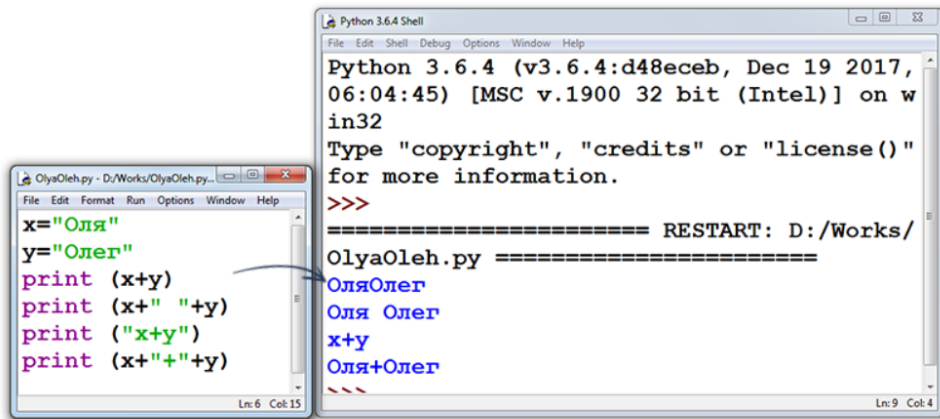
Слайд № 23



The screenshot shows the same Python IDE window as slide 22, but with an additional line of code: `print ("Привіт,|",)`. A callout points to the comma in the second line with the text "Кома". At the bottom, a purple bar with a person icon contains the text: "У наступному рядку набираємо команду print."

Слайд № 24

Нижче показано, як співвідносяться команди виведення з результатом їх виконання.



```
OlyaOleh.py - D:/Works/OlyaOleh.py
File Edit Format Run Options Window Help
x="Оля"
y="Олег"
print (x+y)
print (x+" "+y)
print ("x+y")
print (x+" "+y)
Ln: 6 Col: 15

Python 3.6.4 Shell
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/Works/
OlyaOleh.py =====
ОляОлег
Оля Олег
x+y
Оля+Олег
Ln: 9 Col: 4
```



Ліворуч — вікно програмного коду, праворуч — результат його виконання.

Слайд № 25

Однак змінним можна надавати не довільні імена!

Слід дотримуватися таких правил:

1. Ім'я змінної може містити літери, цифри та знак підкреслення _
2. Ім'я змінної має починатися з літери або знаку підкреслення.
3. Ім'я змінної не може збігатися зі стандартними командами.

Приклади допустимих імен змінних

x1	_x
_x2y1	ABC
A_B_C	A1B2C3

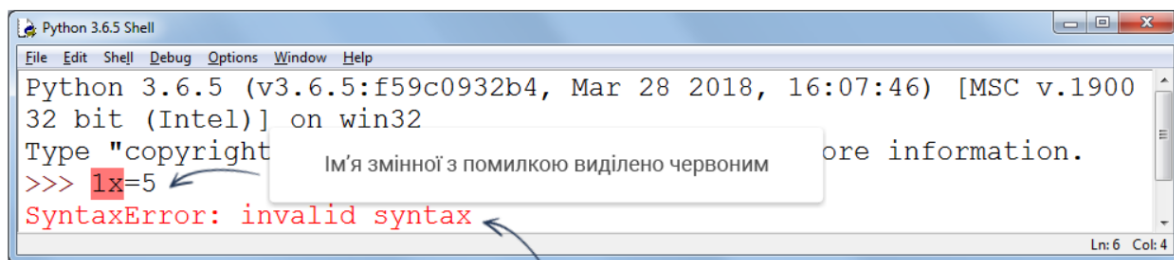
Приклади **не**допустимих імен

1x	Починається з цифри
x!	Містить знак !
A B C	Містить пробіли
input	Збігається з командою



Зауваж, що великі та малі літери в іменах змінних у Python **розрізняються!**

Ось що трапиться після запуску програми, якщо зробити помилку в імені змінної.



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900
32 bit (Intel)] on win32
Type "copyright"
>>> 1x=5
SyntaxError: invalid syntax
```

Повідомлення про помилку



Буде відображено повідомлення про синтаксичну помилку!

Домашнє завдання

Домашнє завдання



Домашнє завдання.

Задача

Створити програму, яка зможе підписувати дипломи учасників конкурсу згідно зразка. Користувач вводить таку інформацію:

- Місто,
- номер школи,
- клас,
- прізвище, ім'я учня

Зразок результату роботи програми

Диплом
переможця конкурсу "Байтик"
Нагороджується
Мудрієвський Олександр ,
учень 7-а класу
ЗОШ I-III ступенів № 7777
міста Києва

Програма має виводити текст як на зразку, підставляючи в нього введені користувачем дані.



Створи у файлі програму, збережи її та перевір роботу.