



Урок 3

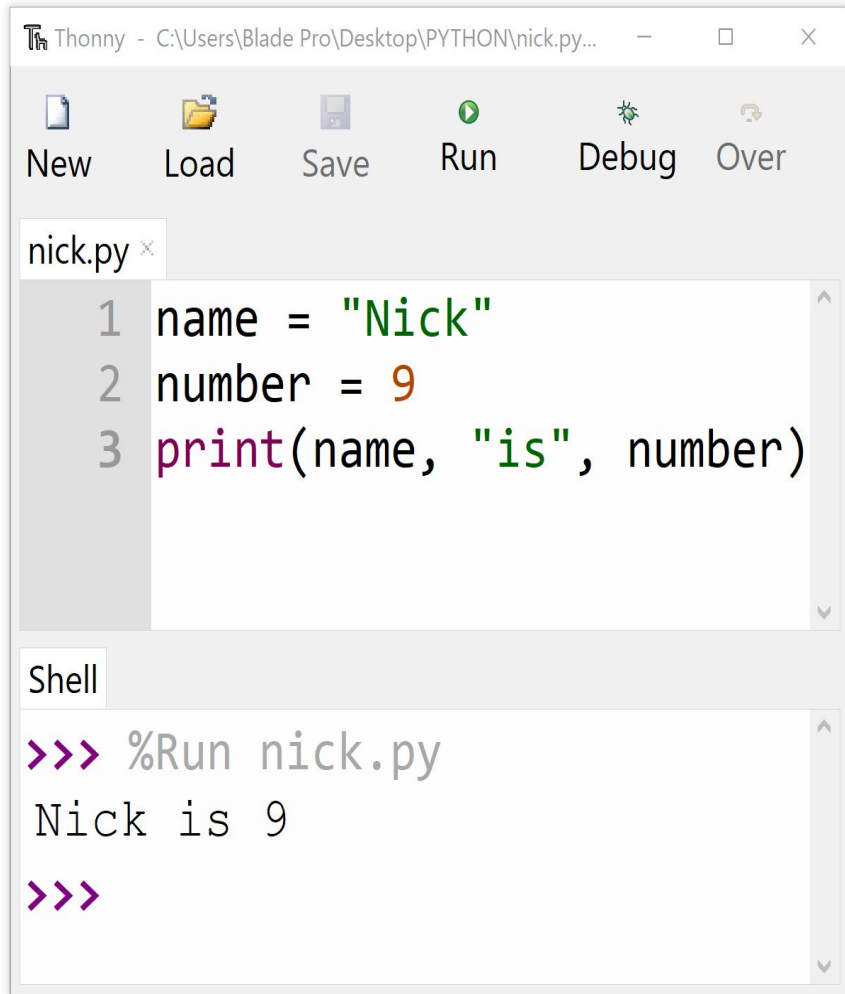
Операції зі стрічками

Форматування стрічок. Конкатенація. Множення стрічок.



Поєднання значень у стрічці

Можна вивести у командному вікні значення кількох змінних вказавши їх ідентифікатори (імена) через кому у функції **print()**. Для зв'язки слів можна додати окремі стрічки прямо в середині функції **print()**.



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\PYTHON\nick.py... The menu bar includes New, Load, Save, Run, Debug, and Over. The editor window, titled 'nick.py', contains the following Python code:

```
1 name = "Nick"
2 number = 9
3 print(name, "is", number)
```

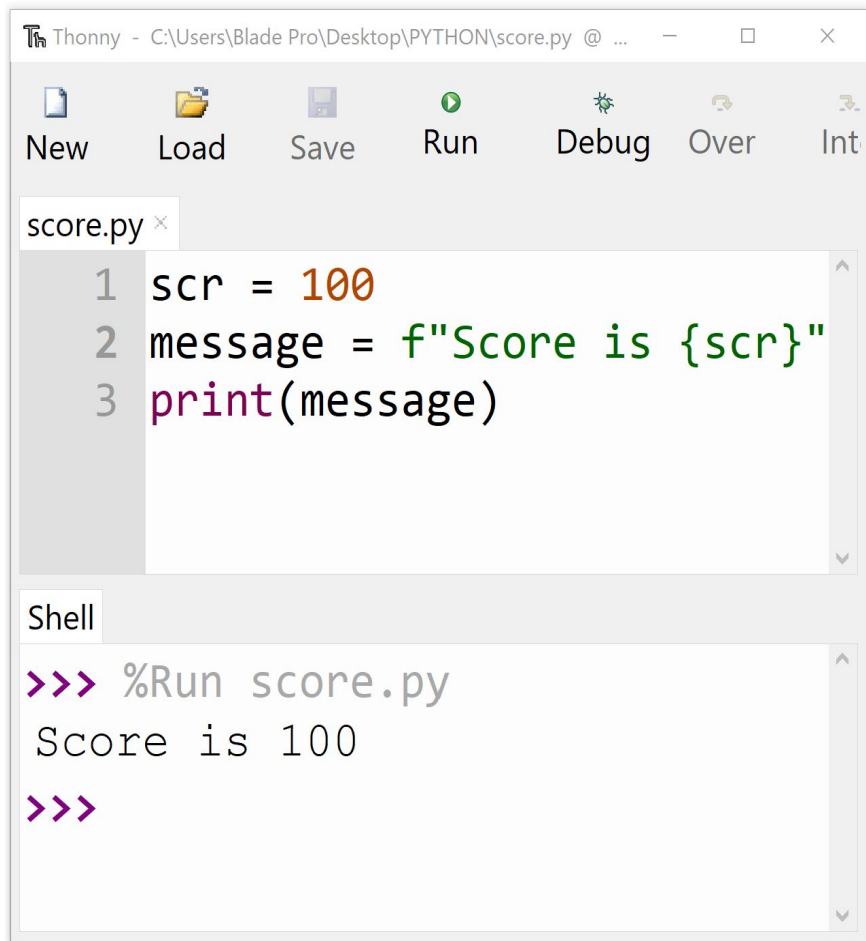
Below the editor is the Shell window, which shows the command prompt output after running the script:

```
>>> %Run nick.py
Nick is 9
>>>
```

Підстановка значень у стрічці

потрібна, коли треба додати у
стрічку значення певної змінної.

Це можна зробити за
допомогою оператора **%** або
скориставшись функцією
форматування **format()**.



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\Blade Pro\Desktop\PYTHON\score.py @ ...". The menu bar includes "New", "Load", "Save", "Run", "Debug", "Over", and "Int". The editor window, titled "score.py", contains the following Python code:

```
1 scr = 100
2 message = f"Score is {scr}"
3 print(message)
```

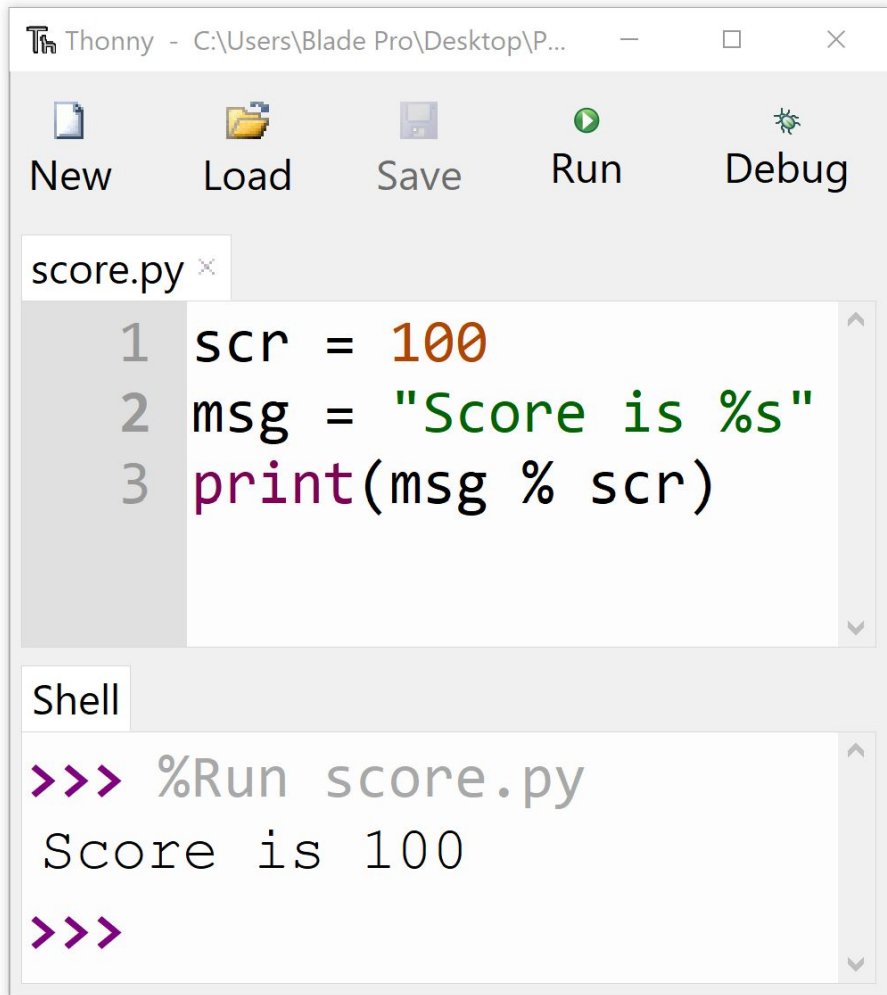
Below the editor is the "Shell" window, which shows the execution of the script:

```
>>> %Run score.py
Score is 100
>>>
```

Використання оператору %

При створенні стрічки використовується оператор **%s** у тому місці, де буде підставлено значення змінної.

У команді виводу стрічки після її імені додається оператор **%** та ім'я відповідної змінної.



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\P... The menu bar includes New, Load, Save, Run, and Debug. The editor window displays a file named score.py with the following code:

```
1 scr = 100
2 msg = "Score is %s"
3 print(msg % scr)
```

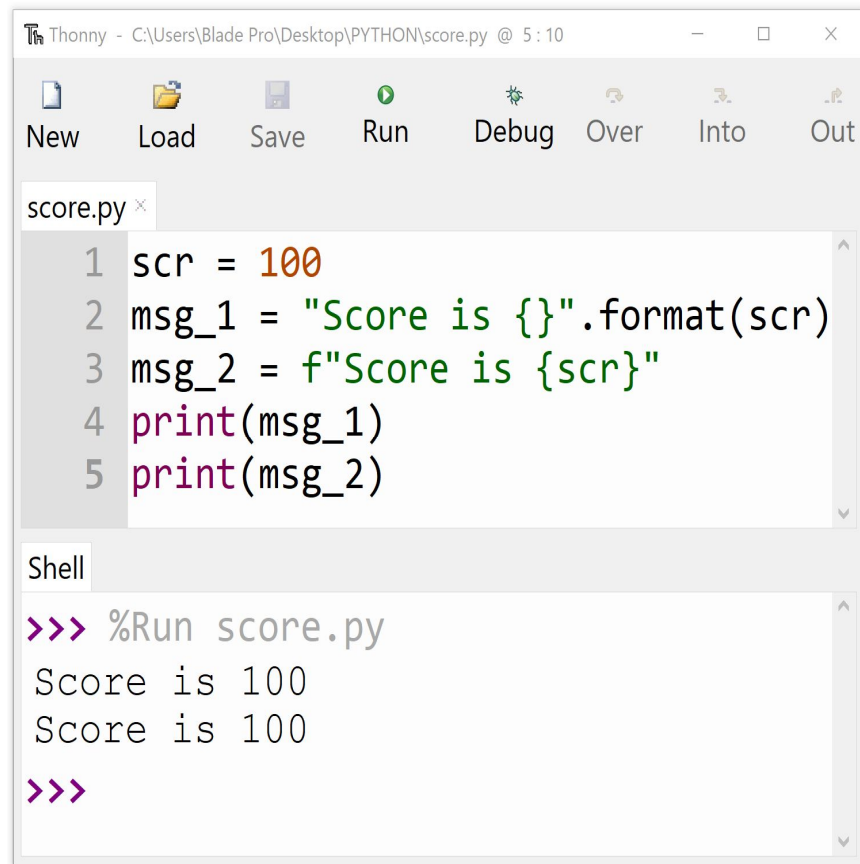
Below the editor is a Shell window showing the execution of the script:

```
>>> %Run score.py
Score is 100
>>>
```

Використання функції `format()`

При створенні стрічки використовується оператор `{}` у тому місці, де буде підсталено значення змінної. Після стрічки додається функція **`format()`** з ім'ям змінної у якості аргументу.

Можна використовувати скорочений варіант.



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: `C:\Users\Blade Pro\Desktop\PYTHON\score.py` at 5:10. The menu bar includes options: New, Load, Save, Run, Debug, Over, Into, and Out. The main editor window displays a Python script named `score.py` with the following code:

```
1 scr = 100
2 msg_1 = "Score is {}".format(scr)
3 msg_2 = f"Score is {scr}"
4 print(msg_1)
5 print(msg_2)
```

Below the editor is a Shell window showing the execution of the script:

```
>>> %Run score.py
Score is 100
Score is 100
>>>
```

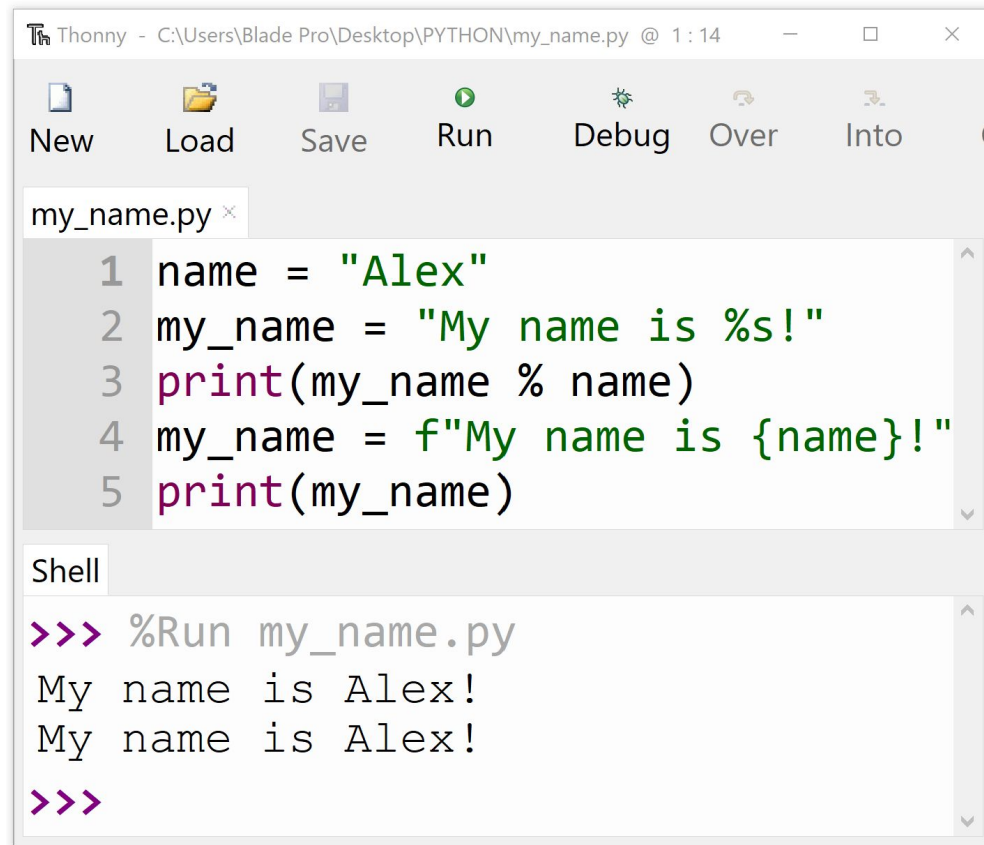


Практична робота

- 1) Створіть новий файл **(New)**
- 2) Надрукуйте наведені стрічки та підставте значення змінної **name** двома різними способами
- 3) Збережіть файл **(Save)** під назвою **my_name.py**
- 4) Виведіть значення змінної **my_name** у вікно **Shell** за допомогою функції **print()**

```
name = "Alex"
```

```
my_name = "My name is name!"
```



The image shows a screenshot of the Thonny Python IDE. The title bar indicates the file path is C:\Users\Blade Pro\Desktop\PYTHON\my_name.py. The interface includes a menu bar with options: New, Load, Save, Run, Debug, Over, and Into. Below the menu bar, the file name 'my_name.py' is displayed. The main editor area contains the following Python code:

```
1 name = "Alex"
2 my_name = "My name is %s!"
3 print(my_name % name)
4 my_name = f"My name is {name}!"
5 print(my_name)
```

Below the code editor is a 'Shell' window showing the execution output:

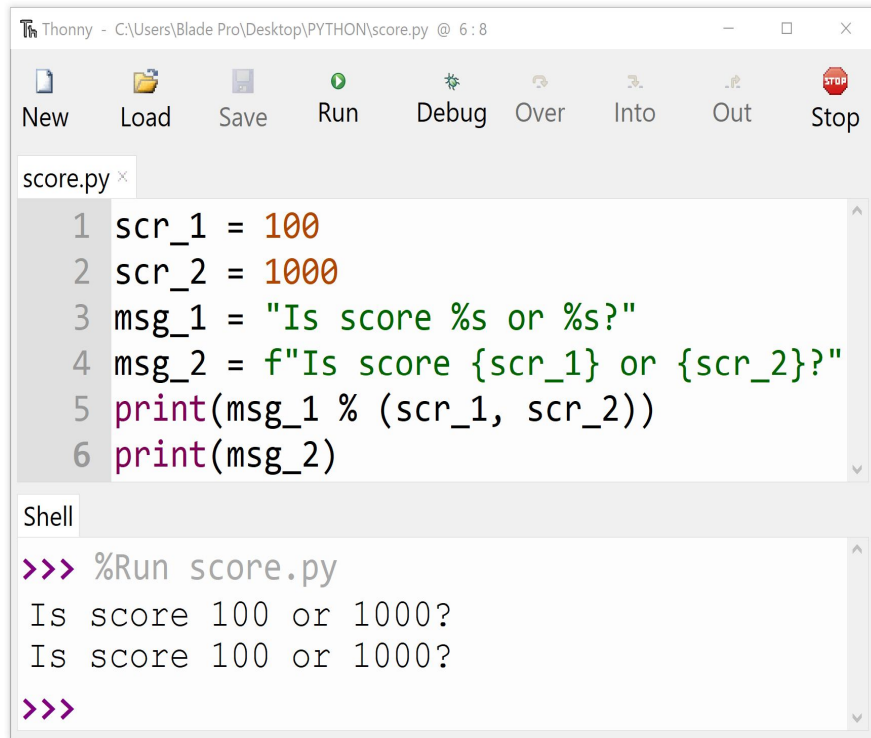
```
>>> %Run my_name.py
My name is Alex!
My name is Alex!
>>>
```

Якщо все зроблено правильно - ви побачите такий результат

Підстановка кількох значень

У стрічку можна підставити не одне, а кілька значень.

Якщо використовується підстановка за допомогою оператора `%`, то значення вказуються у скобках через кому.



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: `C:\Users\Blade Pro\Desktop\PYTHON\score.py @ 6 : 8`. The menu bar includes options: New, Load, Save, Run, Debug, Over, Into, Out, and Stop. The editor window displays the following Python code in `score.py`:

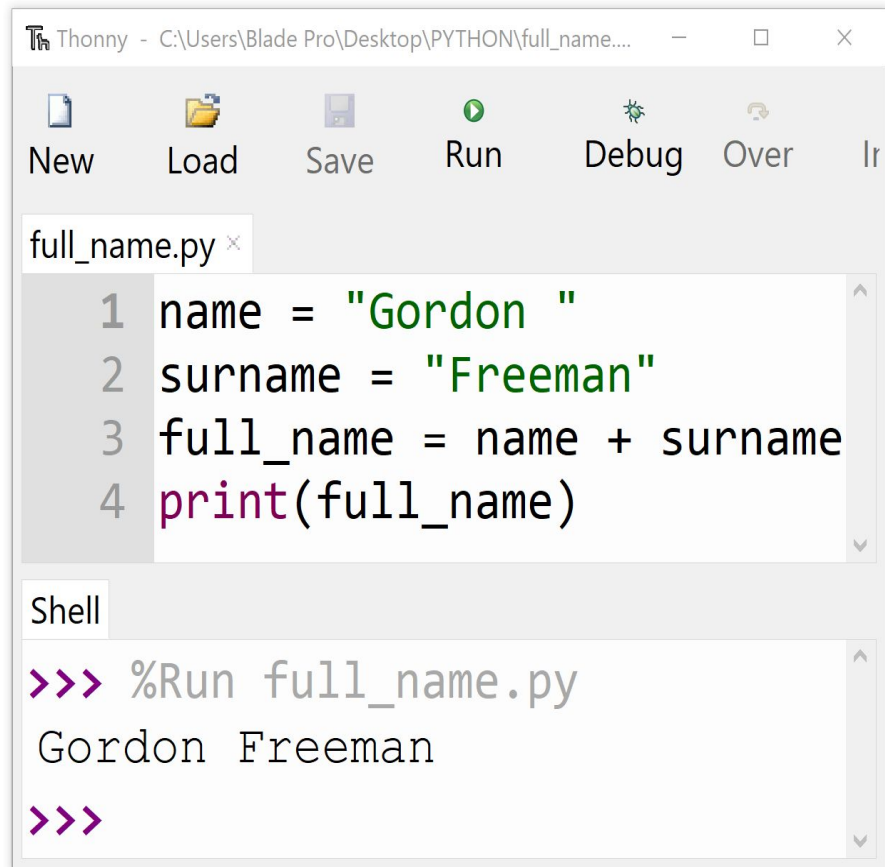
```
1 scr_1 = 100
2 scr_2 = 1000
3 msg_1 = "Is score %s or %s?"
4 msg_2 = f"Is score {scr_1} or {scr_2}?"
5 print(msg_1 % (scr_1, scr_2))
6 print(msg_2)
```

Below the editor is the Shell window, which shows the execution of the script using the `%Run` command:

```
>>> %Run score.py
Is score 100 or 1000?
Is score 100 or 1000?
>>>
```


Конкатенація

це поєднання стрічок за допомогою оператора +



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\PYTHON\full_name.... The menu bar includes New, Load, Save, Run, Debug, and Over. The main editor window displays a Python script named full_name.py with the following code:

```
1 name = "Gordon "  
2 surname = "Freeman"  
3 full_name = name + surname  
4 print(full_name)
```

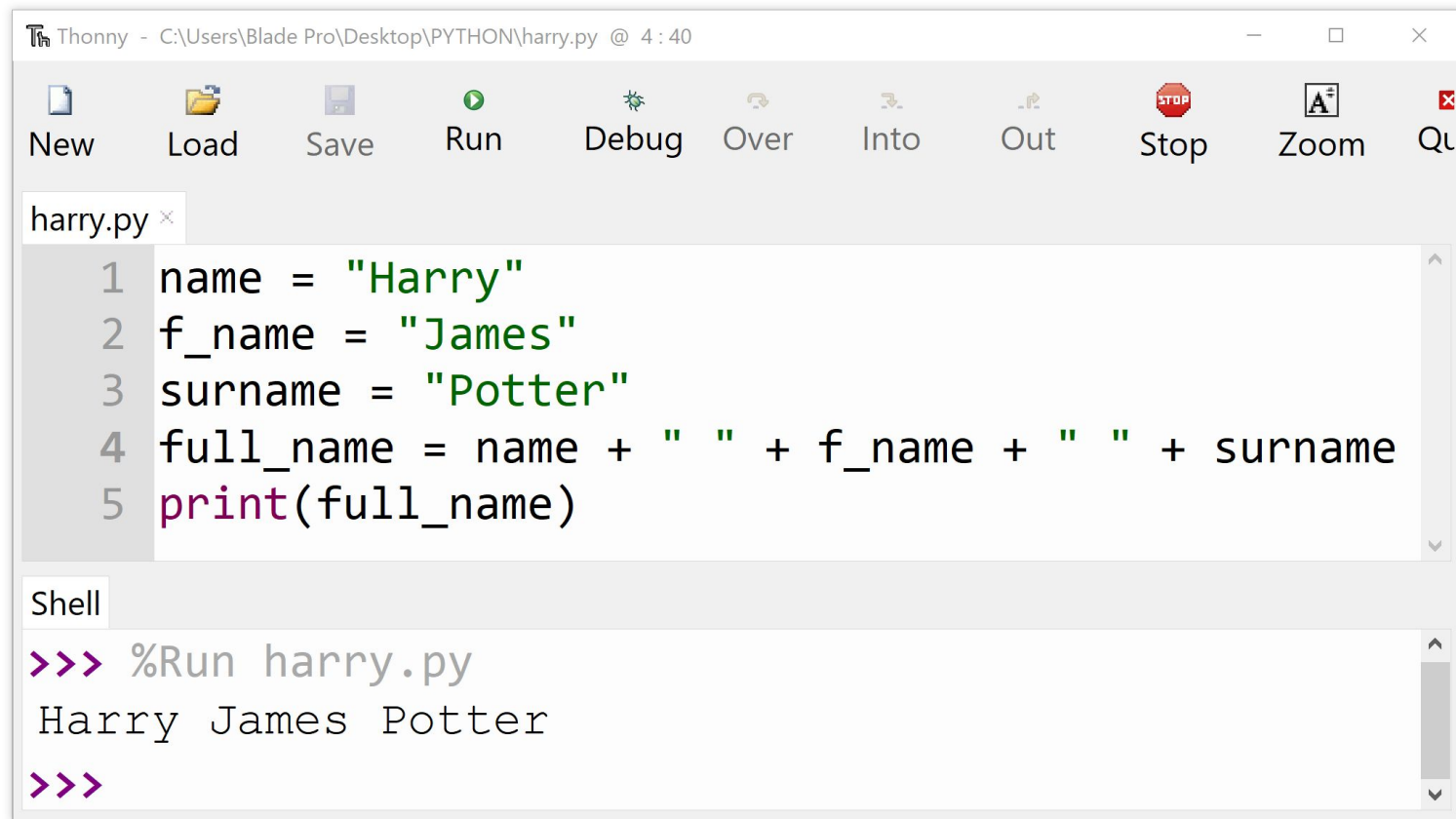
Below the editor is a Shell window. It shows the command %Run full_name.py being executed, followed by the output Gordon Freeman.

```
>>> %Run full_name.py  
Gordon Freeman  
>>>
```



Практична робота

- 1) Створіть новий файл (**New**)
- 2) У змінній **name** збережіть ім'я **Harry**, у змінній **f_name** - **James**, у змінній **surname** - **Potter**
- 3) Поєднайте всі ці значення у змінній **full_name**
- 4) Збережіть файл (**Save**) під назвою **harry.py**
- 5) Виведіть значення змінної **full_name** у вікно **Shell** за допомогою функції **print()** та дізнайтесь повне ім'я Гаррі Поттера



The image shows a screenshot of the Thonny Python IDE. The title bar at the top reads "Thonny - C:\Users\Blade Pro\Desktop\PYTHON\harry.py @ 4 : 40". Below the title bar is a toolbar with icons for New, Load, Save, Run, Debug, Over, Into, Out, Stop, Zoom, and Quit. The main editor window displays a Python script named "harry.py" with the following code:

```
1 name = "Harry"
2 f_name = "James"
3 surname = "Potter"
4 full_name = name + " " + f_name + " " + surname
5 print(full_name)
```

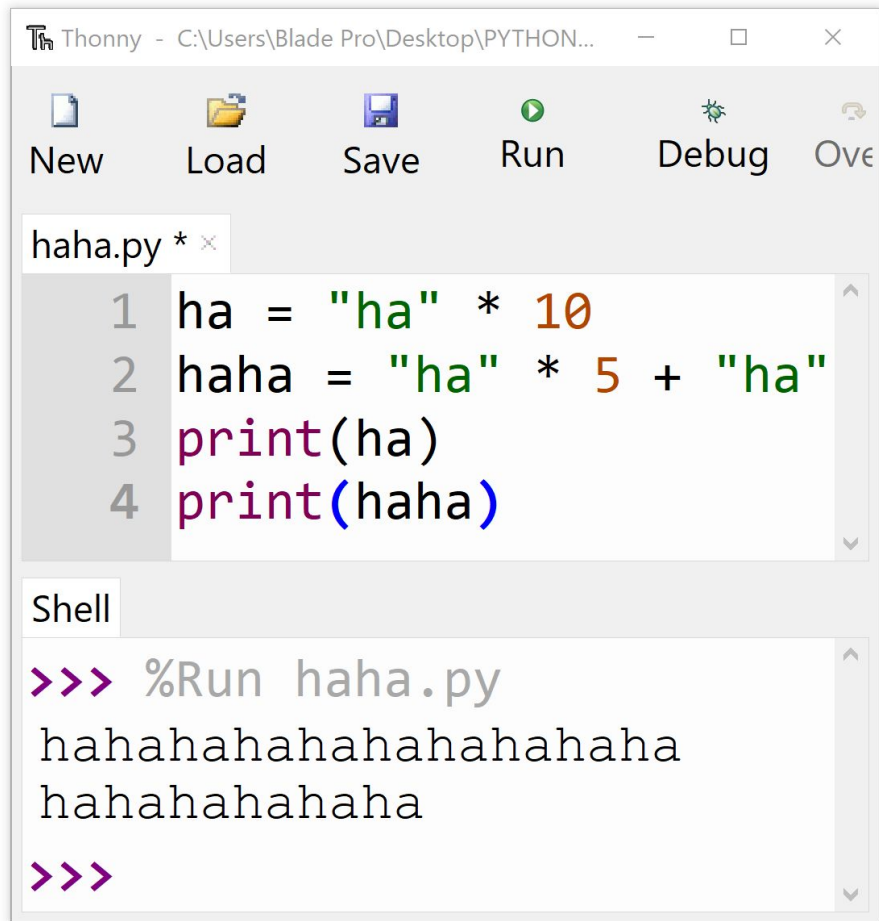
Below the editor is a "Shell" window showing the execution of the script:

```
>>> %Run harry.py
Harry James Potter
>>>
```

Якщо все зроблено правильно - ви побачите такий результат

Множення стрічок

Стрічки можна множити на цілі числа за допомогою оператора множення *



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path is C:\Users\Blade Pro\Desktop\PYTHON... The menu bar includes New, Load, Save, Run, Debug, and Over. The editor window displays a Python script named 'haha.py' with the following code:

```
1 ha = "ha" * 10
2 haha = "ha" * 5 + "ha"
3 print(ha)
4 print(haha)
```

Below the editor is the Shell window, which shows the output of running the script:

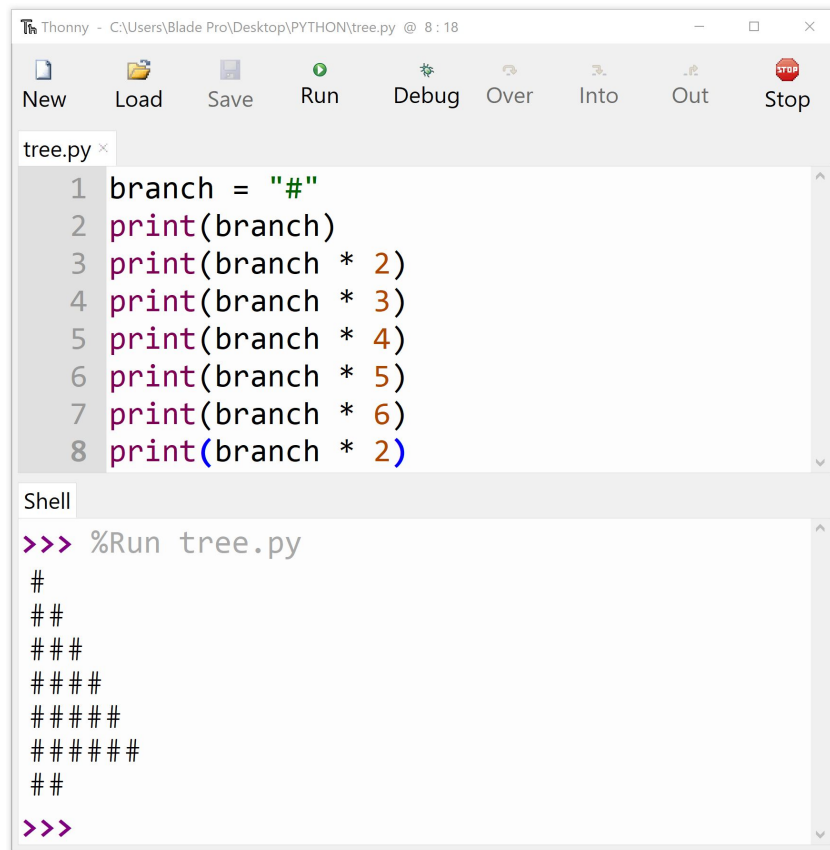
```
>>> %Run haha.py
hahahahahahahahaha
hahahahahahaha
>>>
```



Практична робота

- 1) Створіть новий файл **(New)**
- 2) За допомогою множення стрічок та змінної **branch** із значенням **#**, створіть такий візерунок
- 3) Збережіть файл **(Save)** під назвою **tree.py**
- 4) Виведіть візерунок у вікно **Shell** за допомогою функції **print()**

```
#  
##  
###  
####  
#####  
#####  
#####  
##
```



The image shows a screenshot of the Thonny Python IDE. The window title is "Thonny - C:\Users\Blade Pro\Desktop\PYTHON\tree.py @ 8 : 18". The menu bar includes "New", "Load", "Save", "Run", "Debug", "Over", "Into", "Out", and "Stop". The editor displays a Python script named "tree.py" with the following code:

```
1 branch = "#"  
2 print(branch)  
3 print(branch * 2)  
4 print(branch * 3)  
5 print(branch * 4)  
6 print(branch * 5)  
7 print(branch * 6)  
8 print(branch * 2)
```

The "Shell" pane at the bottom shows the execution output:

```
>>> %Run tree.py  
#  
##  
###  
####  
#####  
#####  
#####  
##  
>>>
```

Якщо все зроблено правильно - ви побачите такий результат



Підсумки

Дізнались, як **підставляти значення змінних** у стрічки різними способами

Навчилися поєднювати стрічки за допомогою **конкатенації**

Познайомились з **множенням стрічок**