



Урок 15

Вбудовані функції в Python

Знайомство з корисними вбудованими функціями у Python

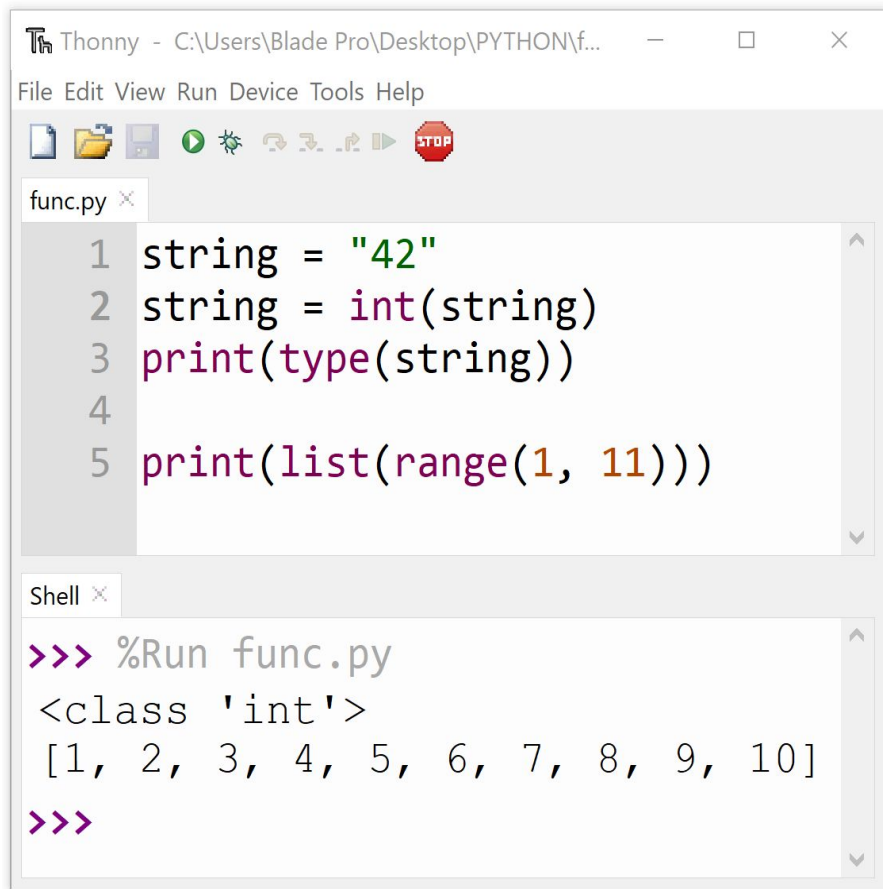


Функція

це **блок коду** для виконання певної дії, який можна застосовувати багато разів у різних частинах своєї програми.

У Python існують вбудовані функції: **print()**, **input()**, **list()**, **range()**, **int()**, **float()**, **str()**, **type()**, **len()** тощо.

Також, функції можна створювати самостійно.



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\Blade Pro\Desktop\PYTHON\func.py. The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for opening files, saving, running, and stopping. The main editor window displays the following Python code in func.py:

```
1 string = "42"
2 string = int(string)
3 print(type(string))
4
5 print(list(range(1, 11)))
```

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

```
>>> %Run func.py
<class 'int'>
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>>
```

Функція “ABS”

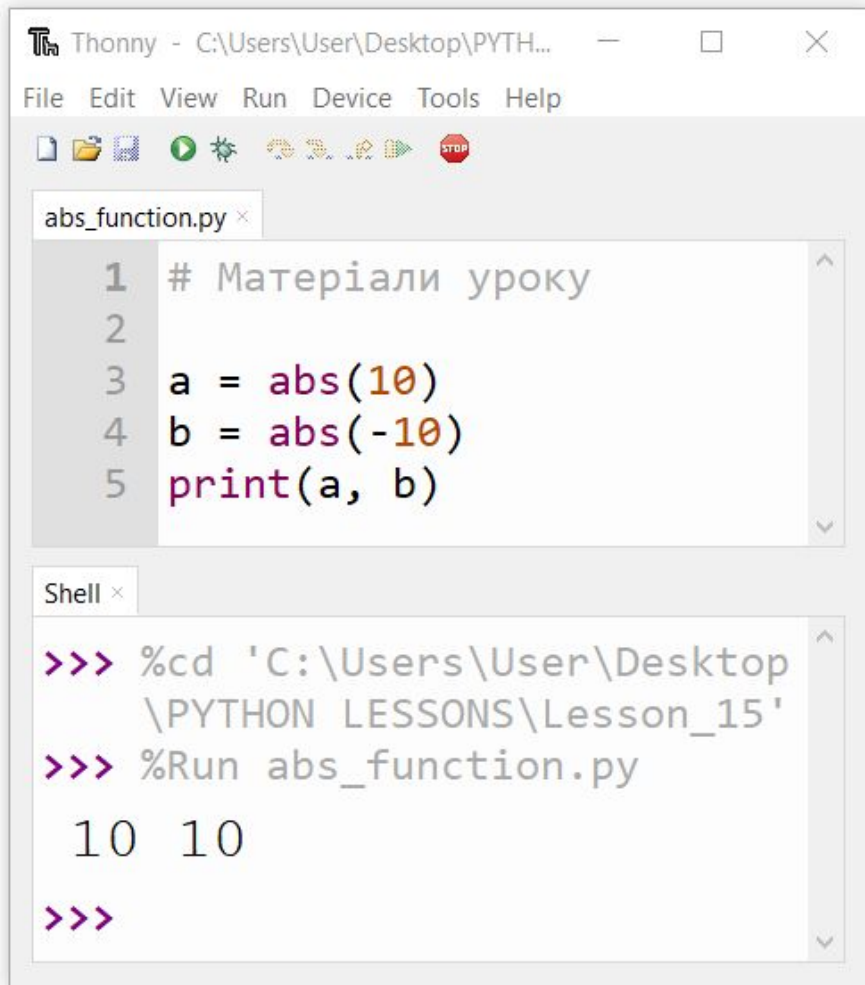
Ця функція повертає абсолютне значення числа, тобто без знаків **+** чи **-**.

abs(число)

Наприклад:

абсолютне значення числа **10** є **10**.

абсолютне значення числа **-10** також є **10**



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\User\Desktop\PYTHON LESSONS\Lesson_15". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". The toolbar contains icons for file operations, running, and debugging. The editor window, titled "abs_function.py", contains the following Python code:

```
1 # Матеріали уроку
2
3 a = abs(10)
4 b = abs(-10)
5 print(a, b)
```

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

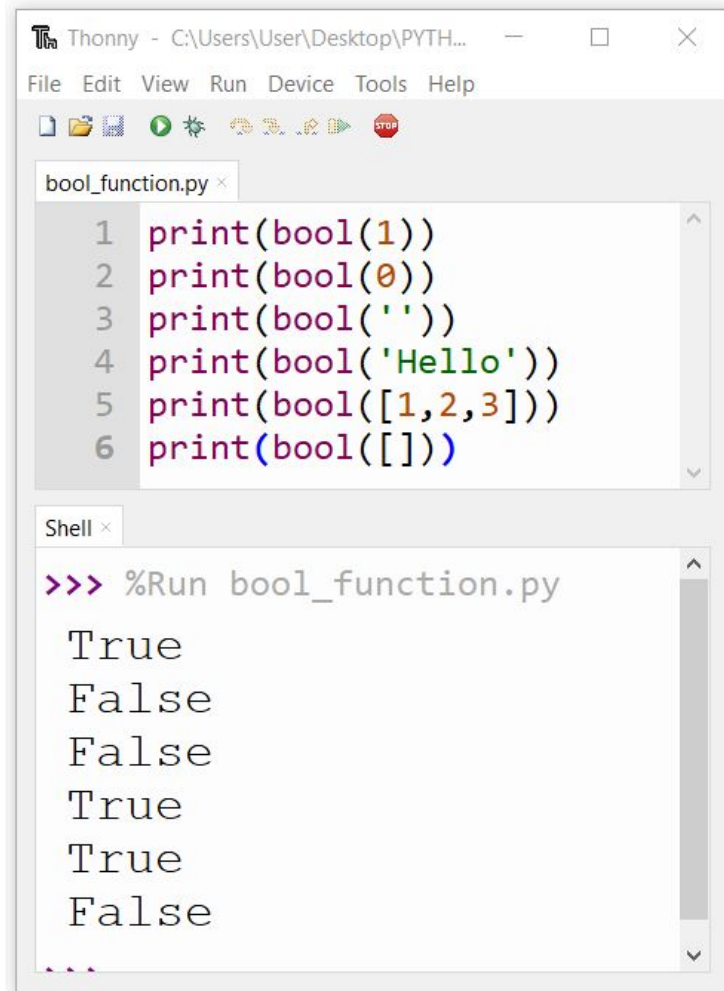
```
>>> %cd 'C:\Users\User\Desktop\PYTHON LESSONS\Lesson_15'
>>> %Run abs_function.py
10 10
>>>
```

Функція "BOOL"

Ця функція бере один параметр і повертає **True** чи **False**, залежно від його значення.

Якщо значення містить число **0**, **порожню стрічку**, **порожній список**, **кортеж** чи **словник** - повертається **False**.

В інших випадках повертається **True**.



The screenshot shows the Thonny Python IDE interface. The top menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main editor window displays a file named `bool_function.py` with the following Python code:

```
1 print(bool(1))
2 print(bool(0))
3 print(bool(''))
4 print(bool('Hello'))
5 print(bool([1,2,3]))
6 print(bool([]))
```

Below the editor is a Shell window showing the output of running the script:

```
>>> %Run bool_function.py
True
False
False
True
True
False
```

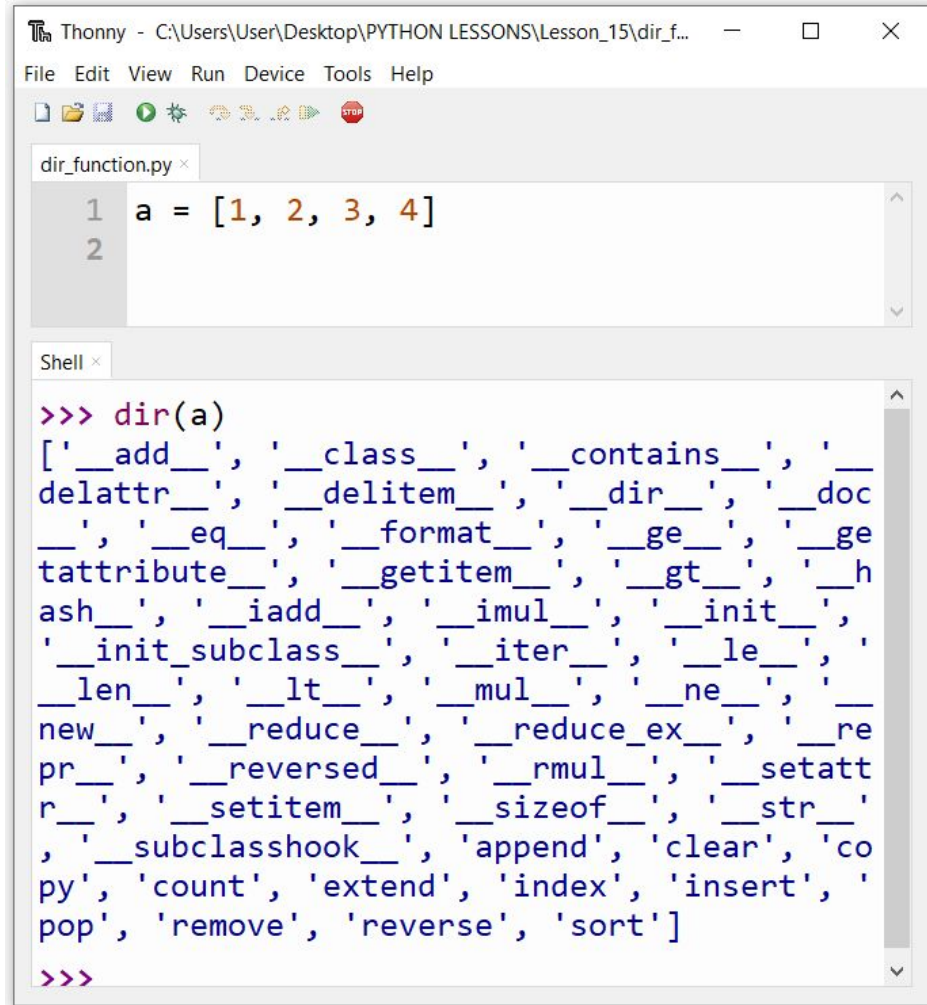
Функція “DIR”

Ця функція повертає інформацію про ті функції, які можна застосувати з введеним значенням.

dir(['список'])

Щоб отримати більше інформації про ці функції, можна скористатись функцією **help()**:

help(змінна.upper)



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\User\Desktop\PYTHON LESSONS\Lesson_15\dir_f... The menu bar includes File, Edit, View, Run, Device, Tools, and Help. The toolbar contains icons for file operations, running, and debugging. The editor window, titled 'dir_function.py', contains two lines of Python code: `1 a = [1, 2, 3, 4]` and `2`. Below the editor is a Shell window showing the output of the `dir(a)` command. The output lists various methods and attributes of a list object, including `['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']`. The prompt `>>>` is visible at the bottom of the shell.

```
Thonny - C:\Users\User\Desktop\PYTHON LESSONS\Lesson_15\dir_f...
File Edit View Run Device Tools Help

dir_function.py x
1 a = [1, 2, 3, 4]
2

Shell x
>>> dir(a)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__', '__imul__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
>>>
```

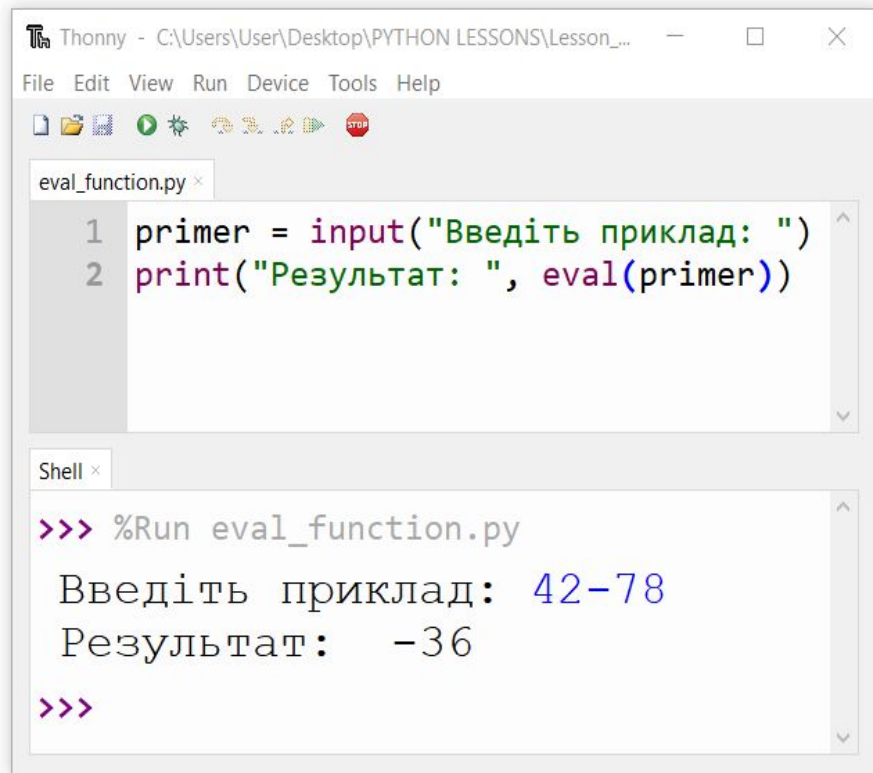
Функція “EVAL”

Ця функція бере стрічку у параметрі та виконує її як **вираз** мови Python.

```
eval("2 + 2") >>> 2 + 2 >>> 4
```

Функція працює тільки з простими значеннями.

Її використовують при роботі з даними, що введені користувачами.



The screenshot shows the Thonny Python IDE interface. The title bar indicates the file path: C:\Users\User\Desktop\PYTHON LESSONS\Lesson_... The menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main editor window, titled 'eval_function.py', contains the following Python code:

```
1 primer = input("Введіть приклад: ")
2 print("Результат: ", eval(primer))
```

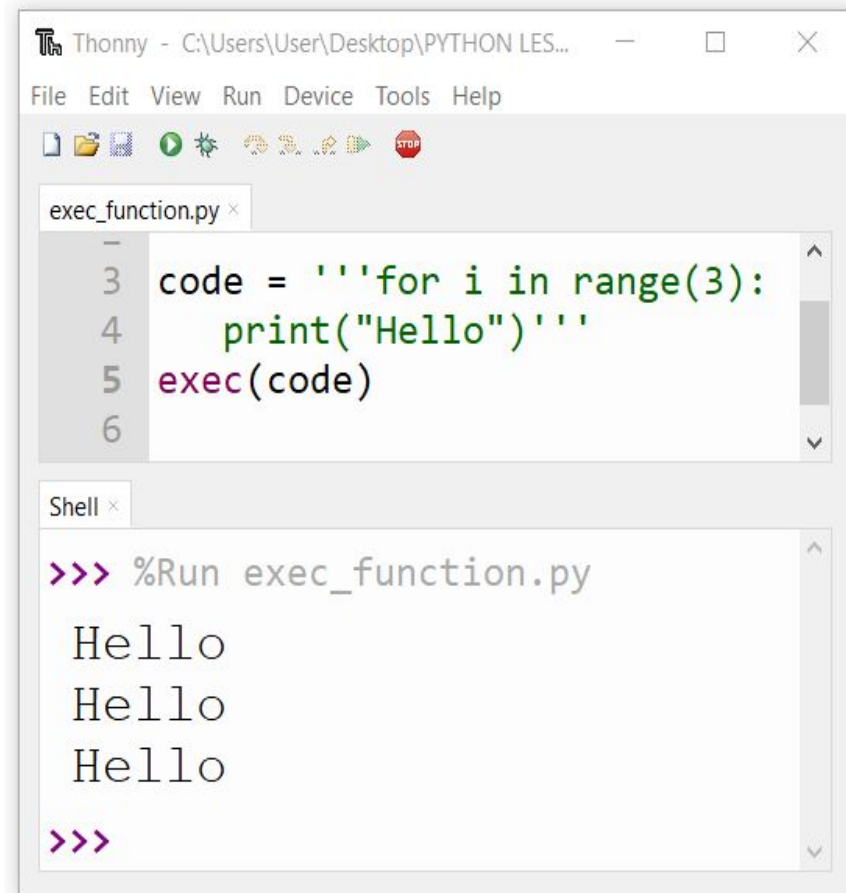
Below the editor is a 'Shell' window showing the execution output:

```
>>> %Run eval_function.py
Введіть приклад: 42-78
Результат:  -36
>>>
```

Функція “EXEC”

Ця функція схожа на функцію **eval()**, але вона застосовується для зчитування скриптів зі стрічок.

Якщо правильно розставити відступи, то можна “зчитувати” навіть багаторядкові конструкції.



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\User\Desktop\PYTHON LES...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". Below the menu is a toolbar with icons for file operations and execution. The main editor window, titled "exec_function.py", contains the following Python code:

```
3 code = '''for i in range(3):
4     print("Hello")'''
5 exec(code)
6
```

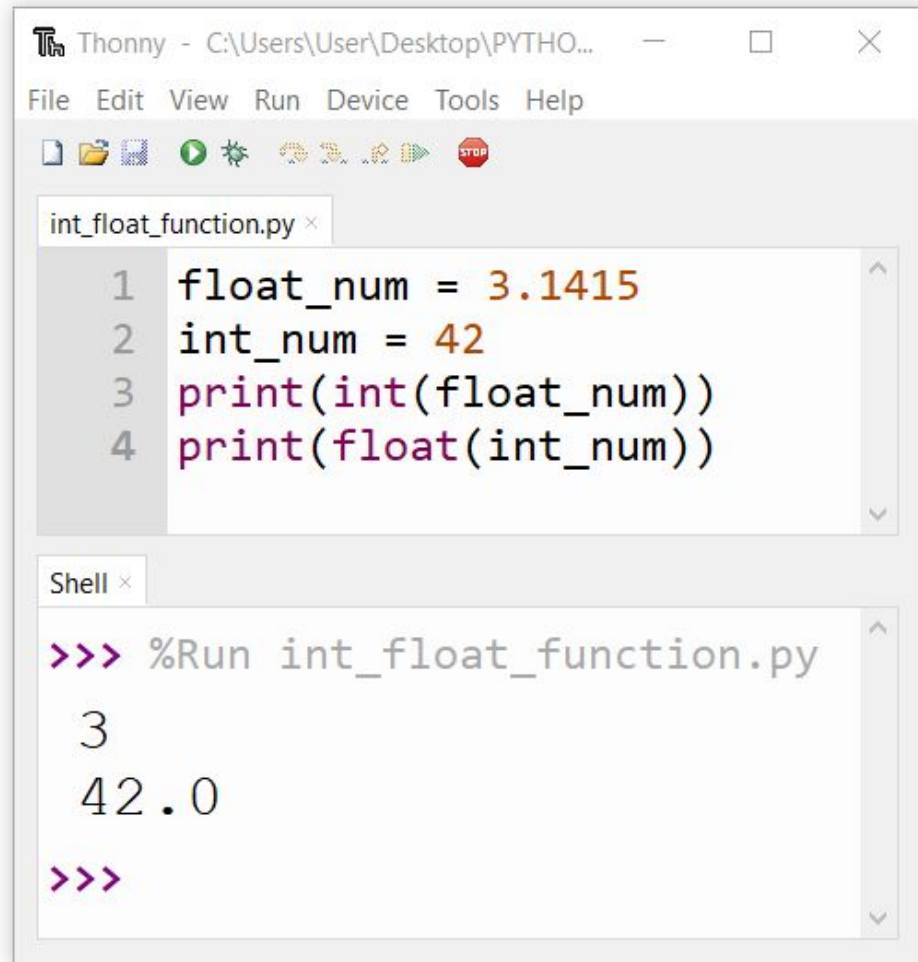
Below the editor is a "Shell" window. It shows the command prompt output after running the script:

```
>>> %Run exec_function.py
Hello
Hello
Hello
>>>
```

Функції “FLOAT” та “INT”

Щоб перетворити число з крапкою (не ціле) або стрічку на **ціле** число, використовують функцію **int()**.
Все, що було після крапки, зникне.

Щоб перетворити ціле число або стрічку на **число з крапкою (не ціле)**, використовують функцію **float()**.



The screenshot shows the Thonny Python IDE interface. The main editor window displays a file named `int_float_function.py` with the following Python code:

```
1 float_num = 3.1415
2 int_num = 42
3 print(int(float_num))
4 print(float(int_num))
```

Below the editor is the Shell window, which shows the command to run the script and its output:

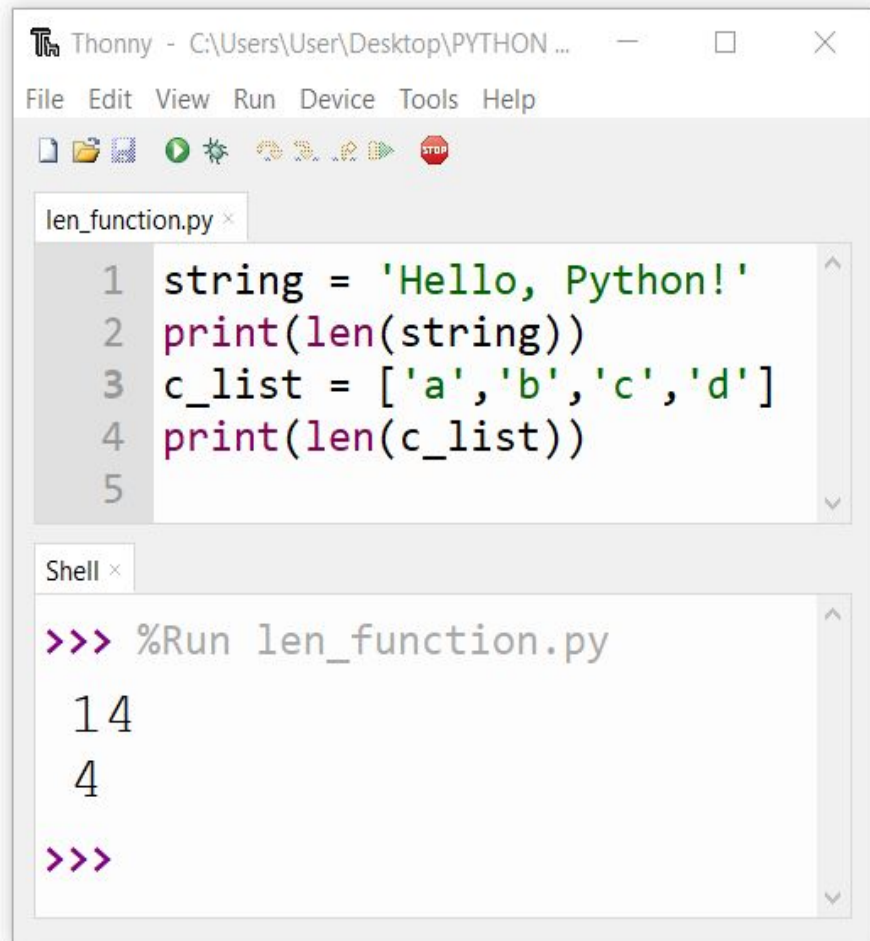
```
>>> %Run int_float_function.py
3
42.0
>>>
```


Функція “LEN”

Ця функція повертає довжину (**кількість елементів**) об'єкта або, у випадку стрічки, кількість символів стрічки.

```
len('Hello')
```

```
>>> 5
```



The screenshot shows the Thonny Python IDE interface. The main editor window displays a file named `len_function.py` with the following Python code:

```
1 string = 'Hello, Python!'
2 print(len(string))
3 c_list = ['a', 'b', 'c', 'd']
4 print(len(c_list))
5
```

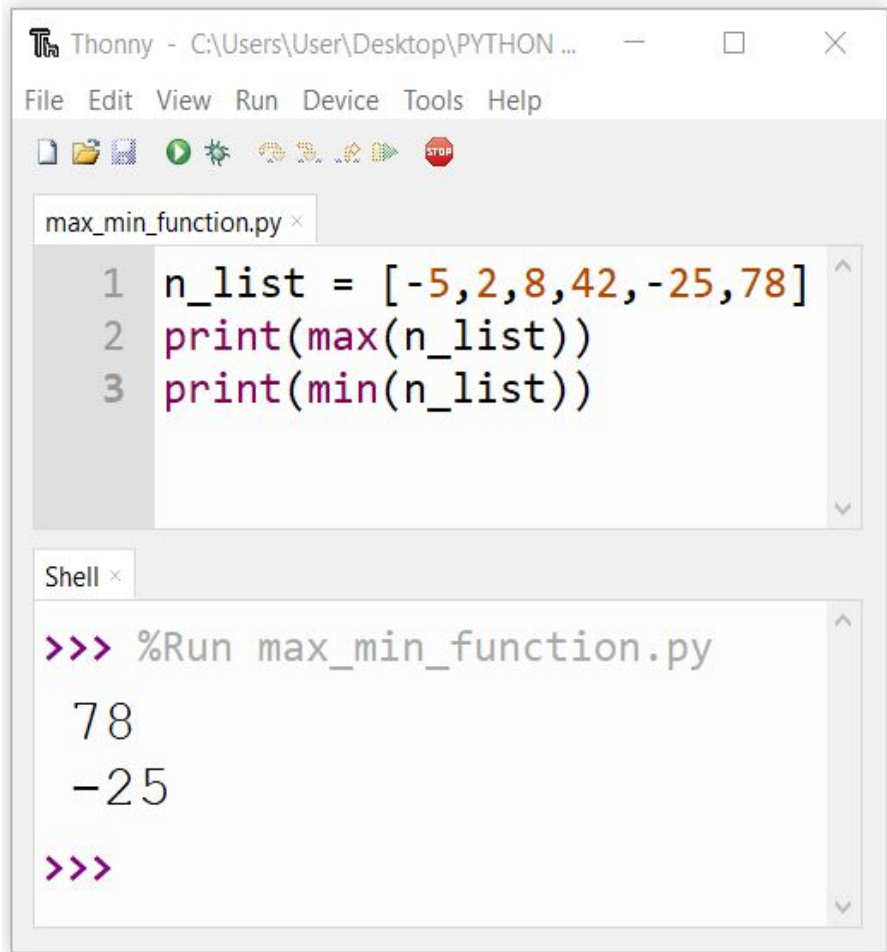
Below the editor is a `Shell` window showing the execution of the script:

```
>>> %Run len_function.py
14
4
>>>
```

Функції “MAX” та “MIN”

Функція **max()** повертає найбільший елемент списку, кортежу та стрічки.

Функція **min()** повертає найменший елемент списку, кортежу та стрічки.



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\User\Desktop\PYTHON ...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". Below the menu is a toolbar with icons for file operations and execution. The main editor window displays a file named "max_min_function.py" with the following Python code:

```
1 n_list = [-5,2,8,42,-25,78]
2 print(max(n_list))
3 print(min(n_list))
```

Below the editor is a "Shell" window showing the execution output:

```
>>> %Run max_min_function.py
78
-25
>>>
```

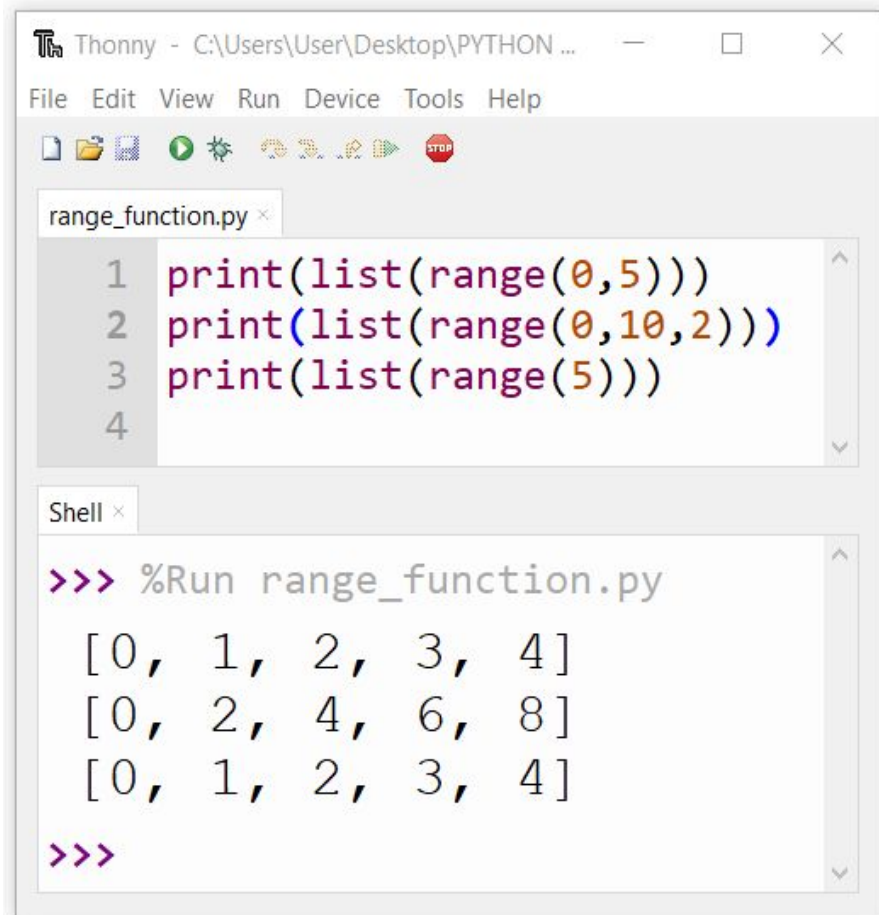
Функція "RANGE"

Ця функція дає змогу задати певний діапазон значень.

range(0, 5)

Числа, які генерує **range**, починаються з числа, заданого як перший параметр, і закінчується числом, яке на одиницю менше за другий параметр.

Існують й інші варіанти використання аргументів цієї функції.



The screenshot shows the Thonny Python IDE interface. The top menu bar includes File, Edit, View, Run, Device, Tools, and Help. Below the menu is a toolbar with icons for file operations and execution. The main editor window displays a file named 'range_function.py' with the following Python code:

```
1 print(list(range(0,5)))
2 print(list(range(0,10,2)))
3 print(list(range(5)))
4
```

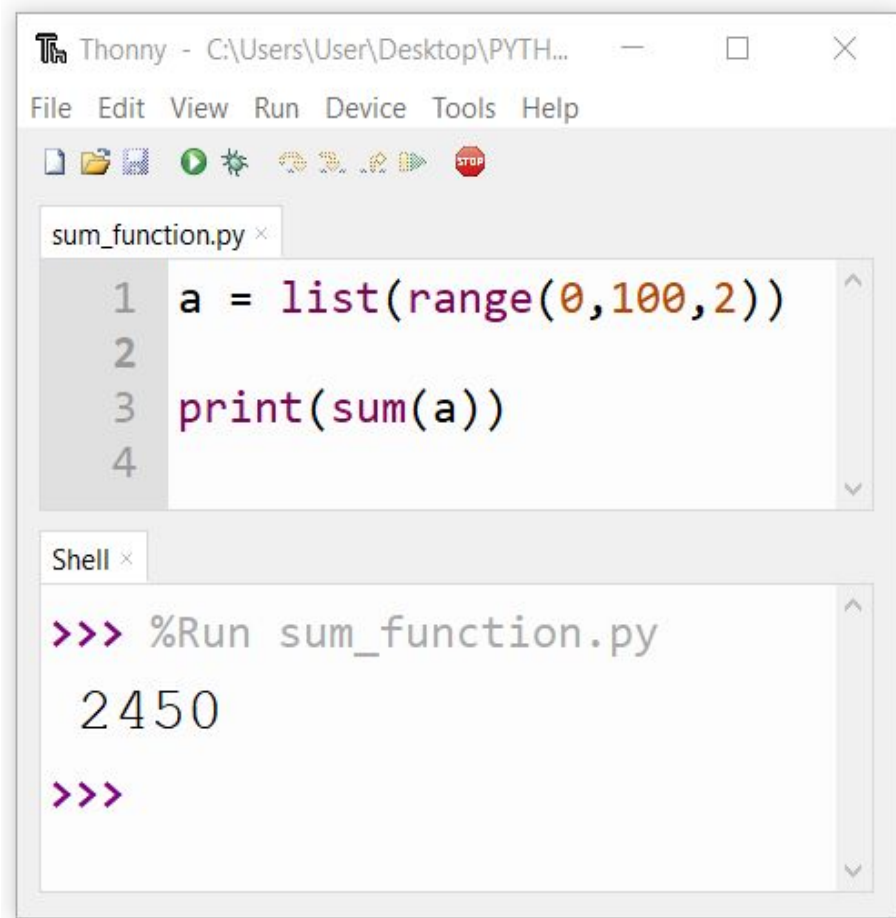
Below the editor is a 'Shell' window showing the execution of the script. The prompt is '>>> %Run range_function.py'. The output consists of three lines of lists:

```
[0, 1, 2, 3, 4]
[0, 2, 4, 6, 8]
[0, 1, 2, 3, 4]
>>>
```

Функція "SUM"

Ця функція додає елементи списку один до одного й повертає їх суму.

`sum(list_of_numbers)`



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\User\Desktop\PYTH...". The menu bar includes "File", "Edit", "View", "Run", "Device", "Tools", and "Help". Below the menu is a toolbar with icons for file operations, running, and debugging. The main editor window, titled "sum_function.py", contains the following Python code:

```
1 a = list(range(0,100,2))
2
3 print(sum(a))
4
```

Below the editor is a "Shell" window. It shows the command prompt with the command `>>> %Run sum_function.py` and the output `2450`. The prompt `>>>` is shown again on the next line.



Практична робота

- 1) Створіть новий файл (**New**)
- 2) Створіть програму-калькулятор, використовуючи деякі вбудовані функції Python
- 3) Збережіть файл (**Save**) під назвою `my_new_calc.py`
- 4) Виведіть результат у вікно **Shell** за допомогою функції `print()`



Підсумки

Дізнались, які корисні **функції** існують в Python

Навчилися використовувати **вбудовані функції**