



Урок 16

Корисні модулі в Python

Знайомство з корисними вбудованими модулями в Python

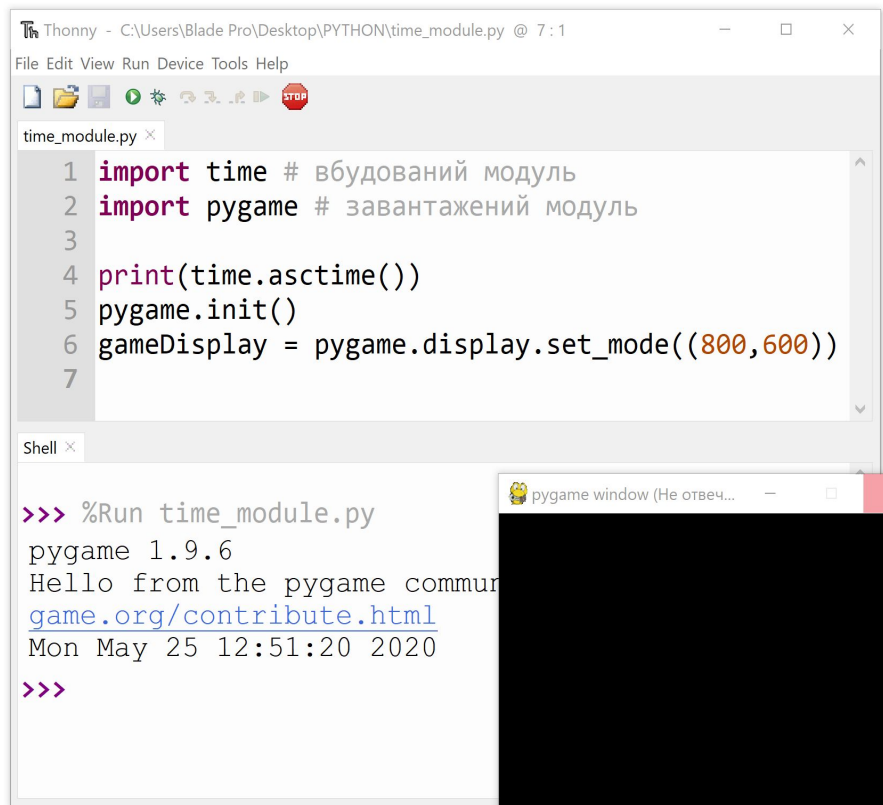


Модуль

це сукупність готових функцій та іншого корисного коду, які можна використовувати в своїй програмі.

Існують вбудовані модулі та ті, що треба завантажувати окремо.

Модуль підключається за допомогою команди **import** у верхній частині програми.



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

```
1 import time # вбудований модуль
2 import pygame # завантажений модуль
3
4 print(time.asctime())
5 pygame.init()
6 gameDisplay = pygame.display.set_mode((800,600))
7
```

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

```
>>> %Run time_module.py
pygame 1.9.6
Hello from the pygame commu
game.org/contribute.html
Mon May 25 12:51:20 2020
>>>
```

To the right of the Shell window, a small window titled "pygame window (He отвеч..." is visible, showing a black screen.

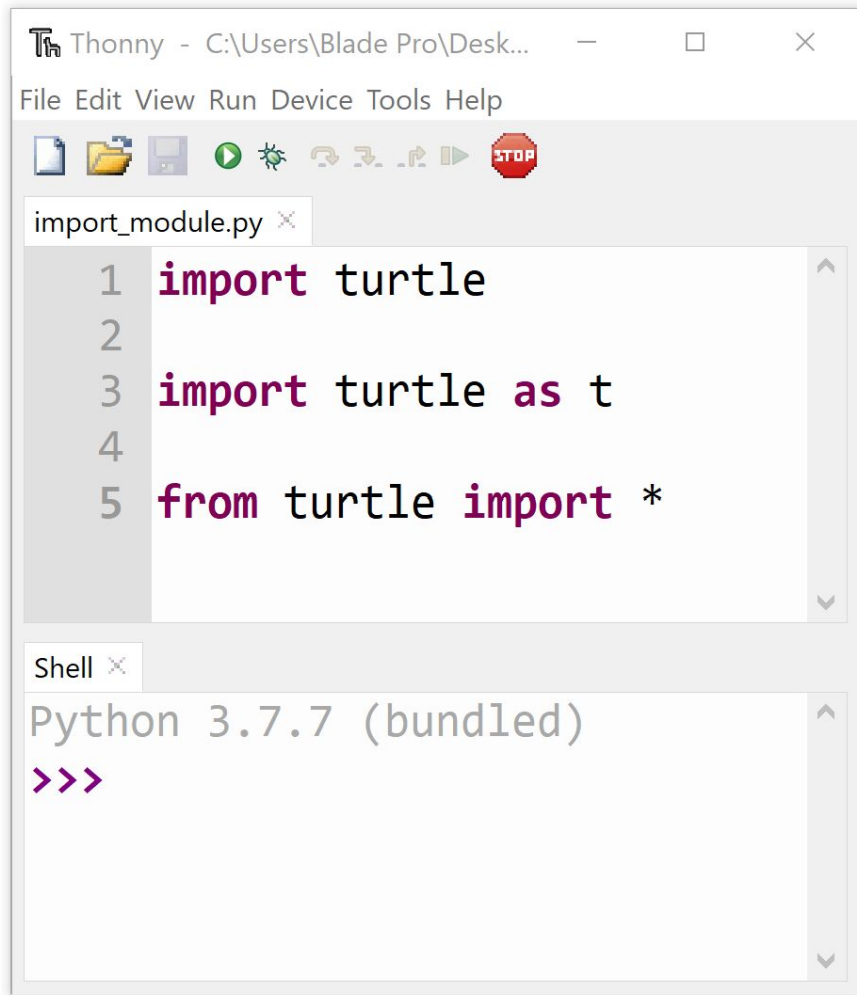
Підключення модулів

Модуль можна підключити кількома способами:

import turtle - ця команда імпортує модуль, але для доступу до його методів треба писати назву модулю.

import turtle as t - ця команда імпортує модуль як і попередня, але для доступу до методів треба писати **t**.

from turtle import * - ця команда імпортує модуль та всі його методи у програму. При використанні методів не треба використовувати його назву.



```
Thonny - C:\Users\Blade Pro\Desktop...
File Edit View Run Device Tools Help

import_module.py x
1 import turtle
2
3 import turtle as t
4
5 from turtle import *

Shell x
Python 3.7.7 (bundled)
>>>
```

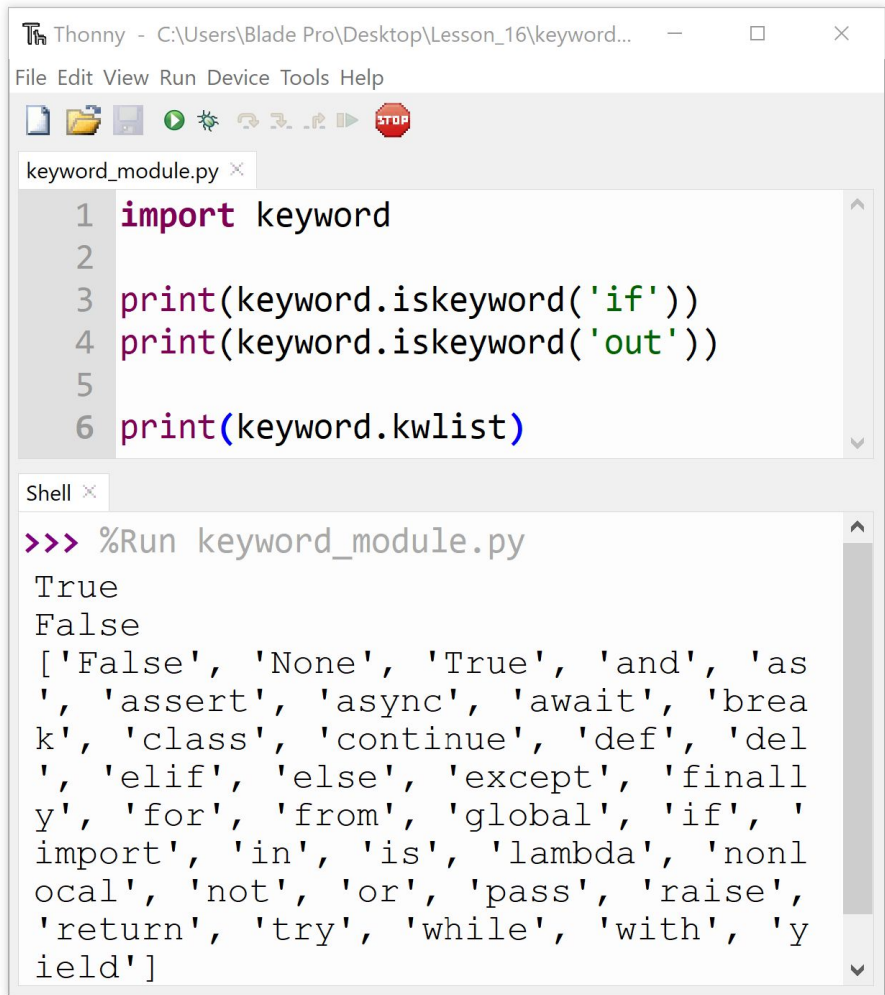
Модуль “KEYWORD”

Цей модуль допомагає визначити зарезервовані слова мови Python.

Функція **keyword.iskeyword(“слово”)** - дозволяє перевірити, є це слово ключовим словом Python (**True**) чи ні (**False**).

Також, у модулі є змінна **kwlist**, яка містить всі ключові слова мови Python. Їх можна побачити, виконавши команду:

```
print(keyword.kwlist)
```



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

```
1 import keyword
2
3 print(keyword.iskeyword('if'))
4 print(keyword.iskeyword('out'))
5
6 print(keyword.kwlist)
```

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

```
>>> %Run keyword_module.py
True
False
['False', 'None', 'True', 'and', 'as',
', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del',
', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

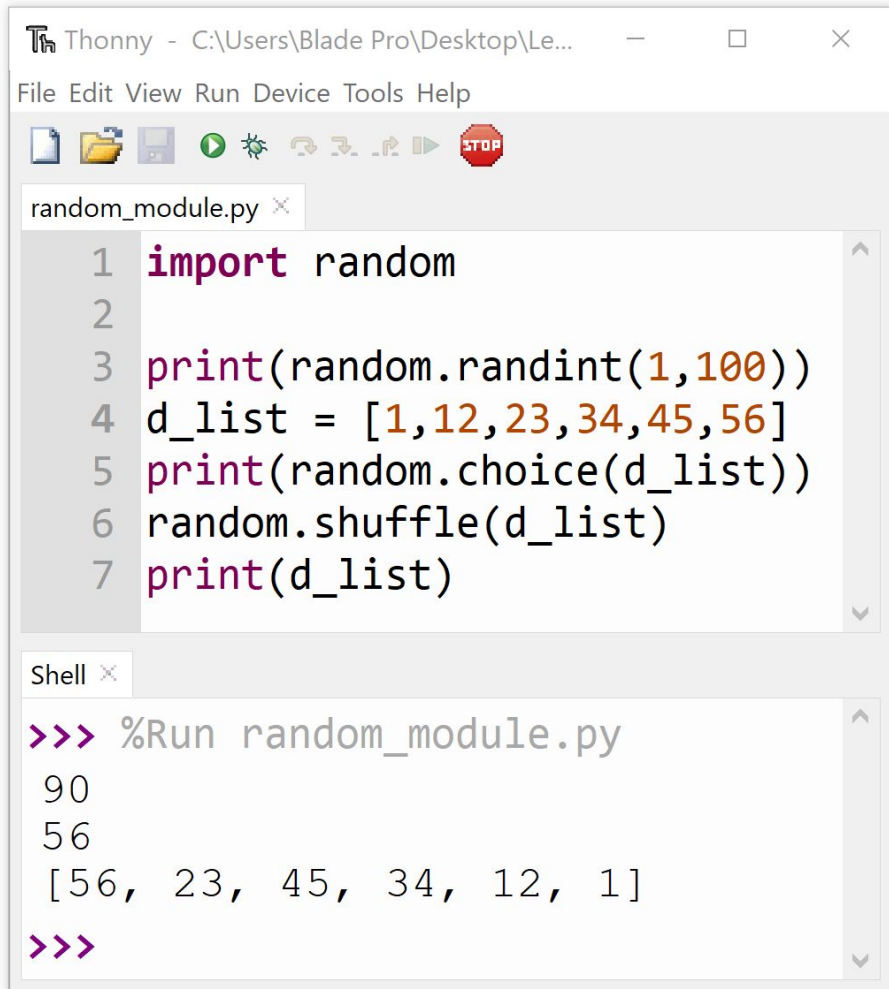
Модуль "RANDOM"

Цей модуль містить кілька функцій для генерування випадкових чисел, наприклад:

randint() - видає випадкове число із заданого діапазону чисел

choice() - видає випадковий елемент зі списку

shuffle() - перемішує елементи списку місцями



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

```
1 import random
2
3 print(random.randint(1,100))
4 d_list = [1,12,23,34,45,56]
5 print(random.choice(d_list))
6 random.shuffle(d_list)
7 print(d_list)
```

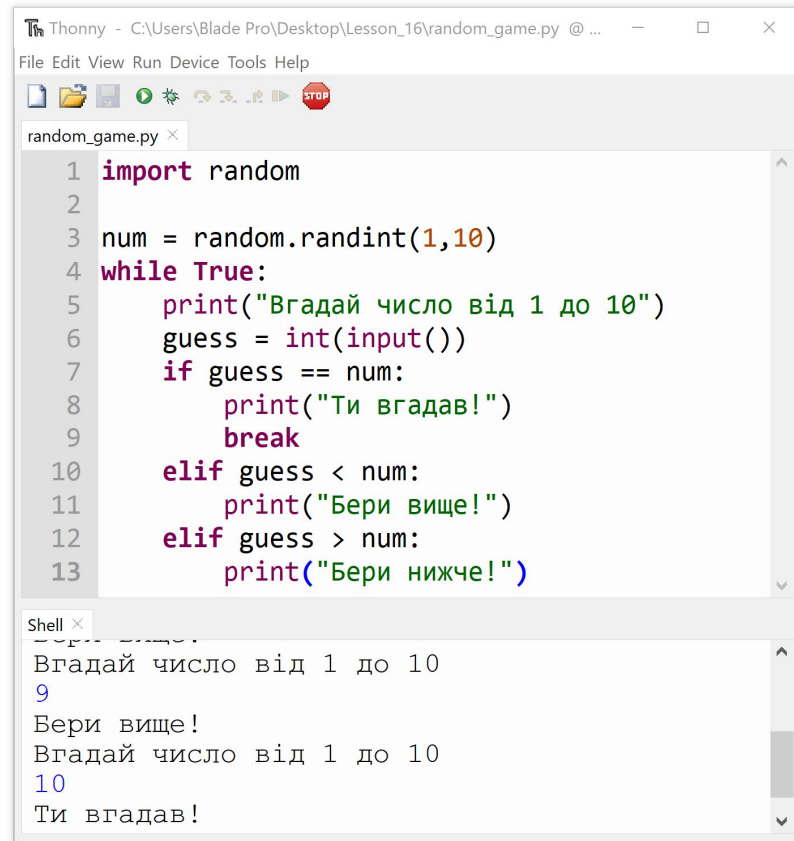
Below the editor is a Shell window showing the execution of the script:

```
>>> %Run random_module.py
90
56
[56, 23, 45, 34, 12, 1]
>>>
```



Практична робота

- 1) Створіть новий файл **(New)**
- 2) Створіть гру, де гравцю треба вгадати число, випадково згенероване програмою у діапазоні від 1 до 10. Якщо гравець вгадав - він отримує повідомлення про це. Якщо помилився - він отримає підказку, чи це число більше від того, яке він ввів, чи менше. Гра закінчується, коли гравець вгадає це випадкове число. Тут у нагоді стануть цикл **while** та умови **if** та **elif**.
- 3) Збережіть файл **(Save)** під назвою **random_game.py**.



The image shows a Thonny Python IDE window. The title bar reads "Thonny - C:\Users\Blade Pro\Desktop\Lesson_16\random_game.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 editor window, titled "random_game.py", contains the following Python code:

```
1 import random
2
3 num = random.randint(1,10)
4 while True:
5     print("Вгадай число від 1 до 10")
6     guess = int(input())
7     if guess == num:
8         print("Ти вгадав!")
9         break
10    elif guess < num:
11        print("Бери вище!")
12    elif guess > num:
13        print("Бери нижче!")
```

Below the editor is a "Shell" window showing the program's output:

```
Вгадай число від 1 до 10
9
Бери вище!
Вгадай число від 1 до 10
10
Ти вгадав!
```

Якщо все зроблено правильно - ви побачите такий результат

Модуль "TIME"

Цей модуль містить функції для роботи з часом.

Наприклад:

time.time() - час, що збіг від 01.01.1970.

time.asctime() - повертає поточний час та дату.

time.localtime() - повертає поточний час та дату як об'єкт. Це дає можливість отримувати інформацію про окремі елементи цього об'єкта.

time.sleep(sec) - зупиняє виконання програми на задану кількість секунд.



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

```
1 import time
2
3 print(time.time())
4 time.sleep(1)
5 print(time.asctime())
6 time.sleep(1)
7
8 print(time.localtime()[0])
9 print(time.localtime()[1])
10 print(time.localtime()[2])
```

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

```
>>> %Run time_module.py
1594886631.268333
Thu Jul 16 11:03:52 2020
2020
7
16
```




Практична робота

- 1) Створіть новий файл **(New)**
- 2) Створіть власний модуль **allabouttime**, за допомогою вбудованого модулю **time**. Цей модуль повинен містити функції **year()**, яка повертає значення поточного року, **month()**, яка повертає назву поточного місяця, **day()**, яка повертає назву поточного дня тижня
- 3) Збережіть файл **(Save)** під назвою **allabouttime.py**



Підсумки

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

Навчилися використовувати **вбудовані модулі** в своїх програмах