



Урок 5

Операції зі списками. Кортежі

Додавання та видалення елементів списку. Арифметичні дії зі списками. Кортежі.

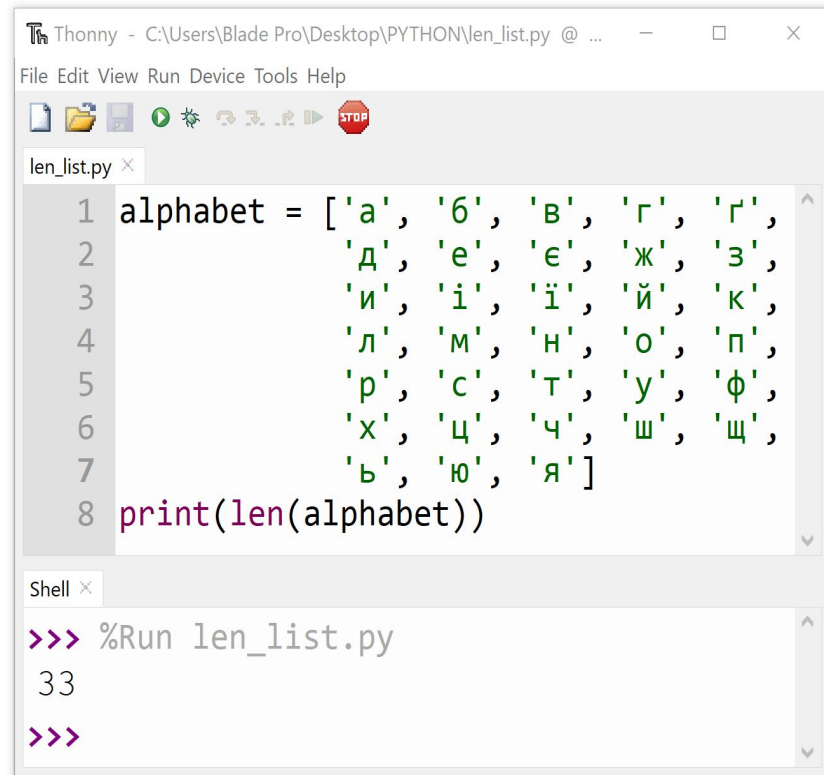


Список

це структура даних яка зберігає послідовність елементів, які вказуються у квадратних дужках `[]` через кому.

Список може включати елементи різних **типів даних**.

Щоб дізнатись кількість елементів у списку, використовують функцію **`len()`**.



The screenshot shows the Thonny Python IDE interface. The main editor window, titled 'len_list.py', contains the following Python code:

```
1 alphabet = ['a', 'б', 'в', 'г', 'ґ',  
2           'д', 'е', 'є', 'ж', 'з',  
3           'и', 'і', 'ї', 'й', 'к',  
4           'л', 'м', 'н', 'о', 'п',  
5           'р', 'с', 'т', 'у', 'ф',  
6           'х', 'ц', 'ч', 'ш', 'щ',  
7           'ь', 'ю', 'я']  
8 print(len(alphabet))
```

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

```
>>> %Run len_list.py  
33  
>>>
```

Додавання елементів до списку

Щоб розширити список, можна застосувати спеціальну функцію **append()**, яка додає додатковий елемент **в кінець списку**.

Функція - це маленька програма, яка вказує мові Python виконати певну дію.



The screenshot shows the Thonny Python IDE interface. The top window, titled 'demo.py', contains the following Python code:

```
1 sports_list = ["футбол", "теніс", "бокс", "хокей"]
2 print(sports_list)
3 sports_list.append("баскетбол")
4 print(sports_list)
```

The bottom window, titled 'Shell', shows the output of running the script:

```
>>> %Run demo.py
['футбол', 'теніс', 'бокс', 'хокей']
['футбол', 'теніс', 'бокс', 'хокей', 'баскетбол']
>>>
```



Практична робота

- 1) Створіть новий файл (**New**)
- 2) Створіть список видів спорту з чотирьох елементів, які ви знаєте з ім'ям **sports_list**
- 3) Додайте у кінець створеного списку ще один вид спорту за допомогою функції **append()**
- 4) Збережіть файл (**Save**) під назвою **sports.py**
- 5) Виведіть всі елементи списку 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\demo.py @ 4 : 19". 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 displays a Python script named "demo.py" with the following code:

```
1 sports_list = ["футбол", "теніс", "бокс", "хокей"]
2 print(sports_list)
3 sports_list.append("баскетбол")
4 print(sports_list)
```

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

```
>>> %Run demo.py
['футбол', 'теніс', 'бокс', 'хокей']
['футбол', 'теніс', 'бокс', 'хокей', 'баскетбол']
>>>
```

Якщо все зроблено правильно - ви побачите такий результат

Видалення елементів списку

Для видалення елементів зі списку використовується спеціальна команда **del**.

Щоб видалити відповідний елемент, треба написати **del** та назву списку з номером цього елементу у квадратних дужках - **del list[5]**.

Порядкові номери починаються з **0**, потім **1**, **2** і т.д.



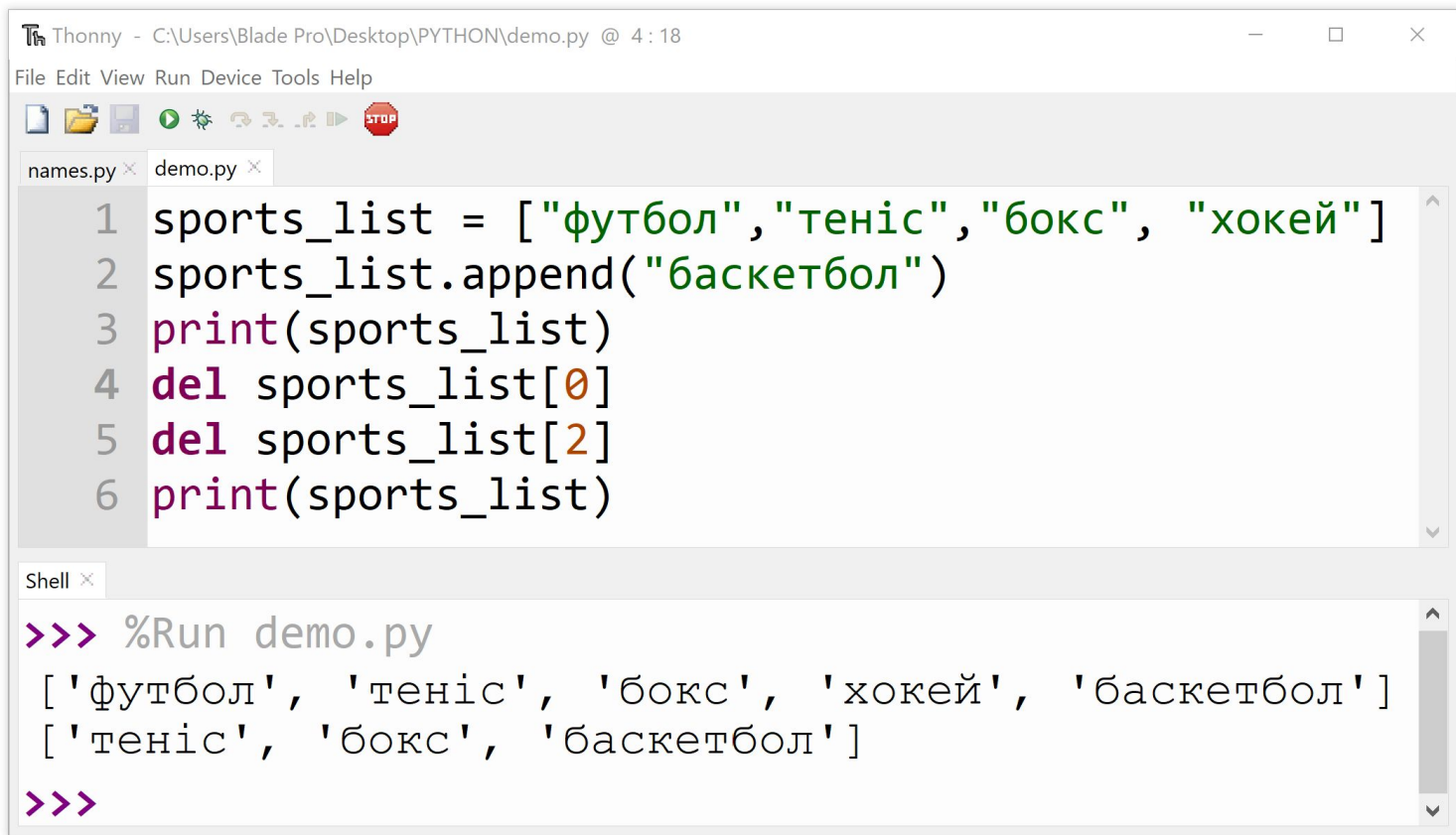
```
Thonny - C:\Users\Blade Pro\Desktop\PYTHON\names.py @ 2:17
File Edit View Run Device Tools Help
names.py x
1 name_list = ["Alex", "Bob", "John", "Poul"]
2 print(name_list)
3 del name_list[1]
4 print(name_list)

Shell x
>>> %Run names.py
['Alex', 'Bob', 'John', 'Poul']
['Alex', 'John', 'Poul']
>>>
```



Практична робота

- 1) Видалить **перший** та **третій** елемент зі свого списку **sports_list**
- 2) Виведіть всі елементи списку у вікно **Shell** за допомогою функції **print()**.



Thonny - C:\Users\Blade Pro\Desktop\PYTHON\demo.py @ 4:18

File Edit View Run Device Tools Help

names.py x demo.py x

```
1 sports_list = ["футбол", "теніс", "бокс", "хокей"]
2 sports_list.append("баскетбол")
3 print(sports_list)
4 del sports_list[0]
5 del sports_list[2]
6 print(sports_list)
```

Shell x


```
>>> %Run demo.py
['футбол', 'теніс', 'бокс', 'хокей', 'баскетбол']
['теніс', 'бокс', 'баскетбол']
>>>
```

Якщо все зроблено правильно - ви побачите такий результат

Арифметика зі списками

Списки можна **додавати (+)** до інших списків та **множити (*)** на ціле число.

Віднімання та ділення списків призведе до помилки.



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

```
1 food_list = ["пиріг", "пюре", "суп"]
2 drink_list = ["сік", "чай", "кофе"]
3
4 print(food_list + drink_list)
5
6 print(drink_list * 3)
```

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

```
>>> %Run food.py
['пиріг', 'пюре', 'суп', 'сік', 'чай', 'кофе']
['сік', 'чай', 'кофе', 'сік', 'чай', 'кофе', 'сік', 'чай', 'кофе']
>>>
```



Практична робота

Помножте список **sports_list** на **3** та виведіть результат у вікно **Shell** за допомогою функції **print()**.



The screenshot shows the Thonny Python IDE interface. The title bar reads "Thonny - C:\Users\Blade Pro\Desktop\PYTHON\demo.py @ 1 : 49". 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 editor window has two tabs: "names.py" and "demo.py *". The "demo.py" tab is active and contains the following Python code:

```
1 sports_list = ["футбол", "теніс", "бокс", "хокей"]
2 print(sports_list * 3)
```

Below the editor is a "Shell" window. It displays the output of the code execution:

```
['футбол', 'теніс', 'бокс', 'хокей', 'футбол', 'теніс', 'бокс', 'хокей', 'футбол', 'теніс', 'бокс', 'хокей']
```

At the bottom of the shell window, there are three purple prompt characters: `>>>`.

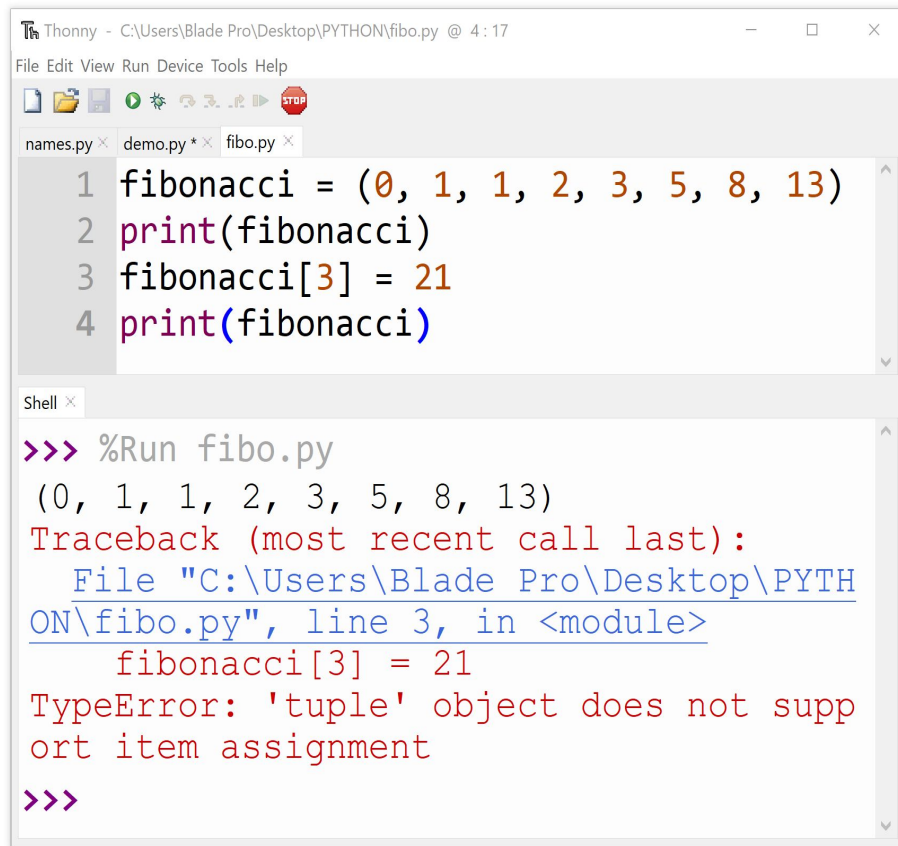
Якщо все зроблено правильно - ви побачите такий результат

Кортеж

це список, вміст якого неможливо змінити після створення.

Кортежі створюються за допомогою круглих дужок ().

При спробі змінити елементи кортежу після створення виникає помилка.



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

```
1 fibonacci = (0, 1, 1, 2, 3, 5, 8, 13)
2 print(fibonacci)
3 fibonacci[3] = 21
4 print(fibonacci)
```

The bottom window, labeled "Shell", shows the execution output:

```
>>> %Run fibonacci.py
(0, 1, 1, 2, 3, 5, 8, 13)
Traceback (most recent call last):
  File "C:\Users\Blade Pro\Desktop\PYTHON\fibonacci.py", line 3, in <module>
    fibonacci[3] = 21
TypeError: 'tuple' object does not support item assignment
>>>
```



Підсумки

Дізнались, як **додавати** та **видаляти** елементи списку.

Навчилися виконувати **арифметичні операції** зі списками.

Познайомились з **кортежами**.