

Python Starter

Списки

Python Starter

План заняття

1. Поняття списку
2. Індокси, зрізи списків
3. Перевірка елементів на входження до списку
4. Робота зі списками (отримання значень елементів, їх повторів, додавання та видалення значень, копіювання та видалення списків і їх сортування, перетворення списку на рядок)

Python Starter

Після уроку обов'язково



Повторіть цей урок у відео форматі на [ITVDN.com](http://itvdn.com)

Доступ можна отримати через керівництво вашого навчального центру



Перевірте, як Ви засвоїли цей матеріал на [TestProvider.com](http://testprovider.com)

Python Starter

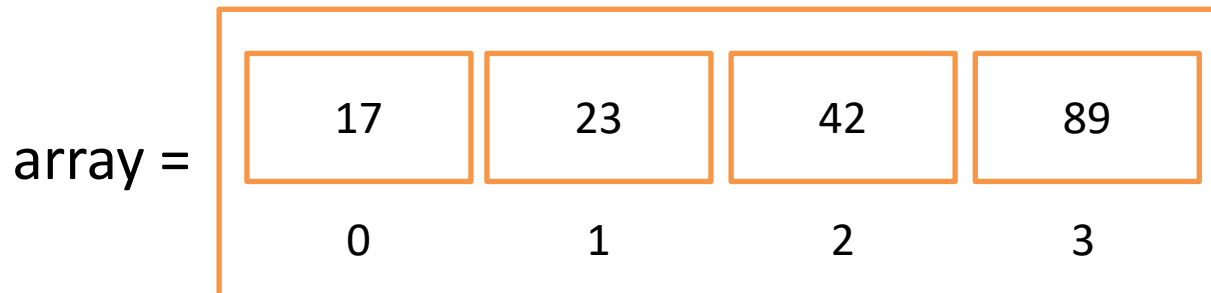
Тема

Списки

Python Starter

Поняття списків та масивів

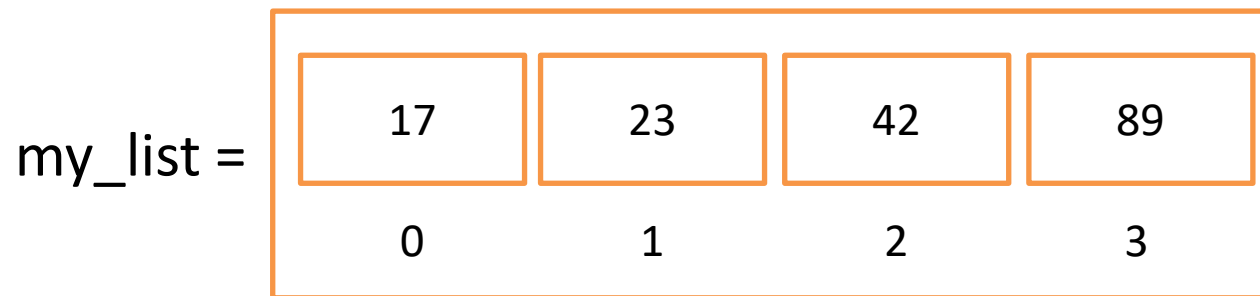
- Масив – набір фіксованої кількості однотипних елементів, розміщених у пам'яті безпосередньо один за одним, доступ до яких здійснюється за індексом (номером).
- Список – це тип даних, що є впорядкованою послідовністю певних значень, які можуть повторюватися. Кількість елементів списку може бути довільною



Python Starter

Списки в Python

- У Python замість масивів зазвичай використовуються списки.
- Тип даних, що реалізує списки – list.
- Список у Python є змінною (mutable) структурою даних.
- Список у Python є гетерогенною структурою даних.



`type(my_list) == list`

Python Starter

Створення списків

Для створення списку слід записати його елементи через кому у квадратних дужках.

```
int_list = [1, 2, 3, 5]
```

```
char_list = ['a', 'c', 'z', 'x']
```

```
empty_list1 = []
```

```
empty_list2 = list()
```

Python Starter

Індексація списків

- В Python списки можна індексувати як масиви.
- Відлік індексів йде з нуля.
- Для отримання елемента за індексом слід після імені списку вказати індекс у квадратних дужках.
- Також можна індексувати списки з кінця. У такому разі задається негативний індекс, починаючи з -1.

```
first_element = my_list[0]
```

```
second_element = my_list[1]
```

```
nth_element = my_list[n]
```

```
last_element = my_list[-1]
```


Python Starter

Зрізи списків

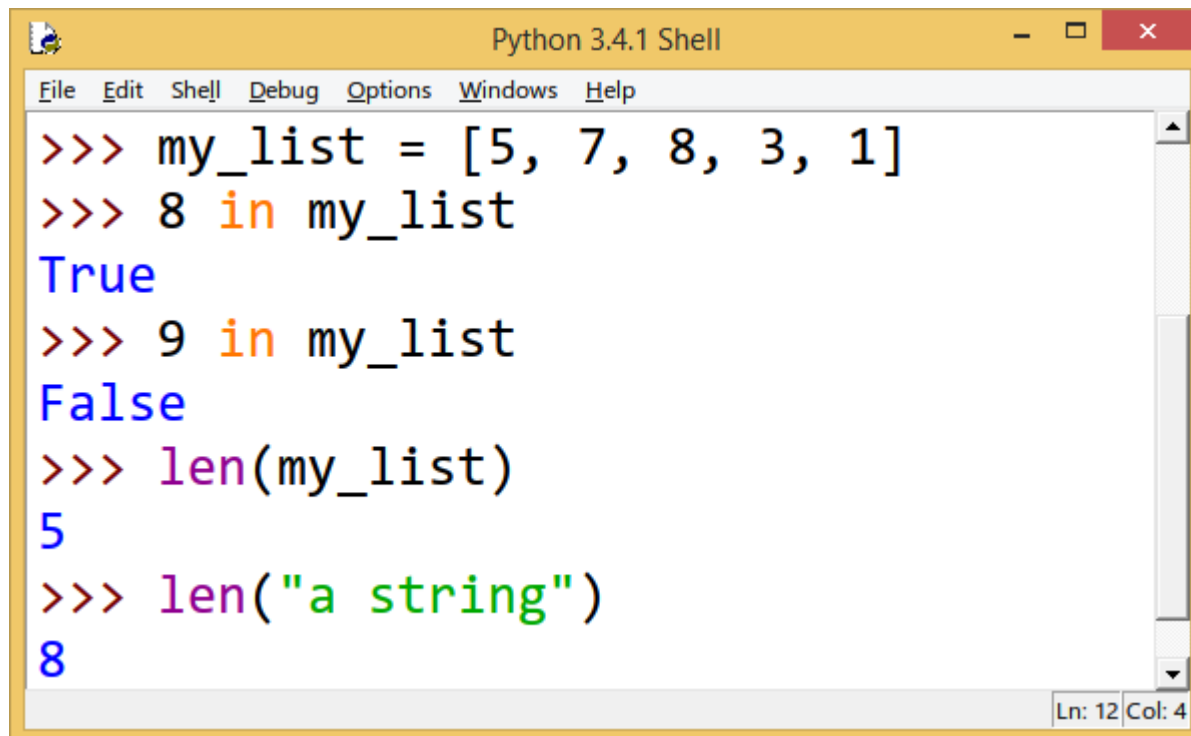
- Можна також отримати групу елементів за їх індексами. Ця операція називається зрізом списку (list slicing).
- Для отримання зрізу списку в квадратних дужках вказується індекс першого елементу, потім через двокрапку записується індекс, на одиницю більший за індекс останнього необхідного елементу. Можна також додати ще одну двокрапку та крок зрізу.
- Можна не вказувати початковий та/або кінцевий індекси. Значенням за замовчуванням для початкового індексу є 0, для кінцевого – довжина списку. Якщо не вказано ні початковий, ні кінцевий індекси, ні крок, то `my_list[:]` поверне копію списку `my_list`.

```
new_list = my_list[start:end:step]
```

Python Starter

Операції зі списками

- Для перевірки, чи є вказане значення елементом списку, використовується операція **in**.
- Для отримання кількості елементів у списку використовується функція **len**.
- Функцію **len**, індексацію, отримання перерізів та перевірку елемента на входження можна також застосовувати до рядків.

A screenshot of a Python 3.4.1 Shell window. The window has a yellow title bar with the text "Python 3.4.1 Shell" and standard window controls. Below the title bar is a menu bar with "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main area is a text editor with a white background and a vertical scrollbar on the right. It contains the following Python code and output:

```
>>> my_list = [5, 7, 8, 3, 1]
>>> 8 in my_list
True
>>> 9 in my_list
False
>>> len(my_list)
5
>>> len("a string")
8
```

The status bar at the bottom right shows "Ln: 12 Col: 4".

Python Starter

Операції зі списками

- Для додавання нового елементу до списку використовується метод `append()`:
`my_list.append(value)`
- Для додавання кількох нових елементів одночасно до списку використовується метод `extend()`:
`value = value1, value2, value3`
`my_list.extend(value)`
- Для видалення елементу списку використовується метод `remove()` – у разі відсутності елемента `value` у списку буде помилка:
`my_list.remove(value)`
- Для видалення елемента списку за індексом використовується :
 - оператор `del`: `del my_list[index]`
 - метод `pop()`: `pop(index)` – (якщо індекс не вказано – видалення останнього елементу з кінця):
`my_list.pop(index)`

Ще однією особливістю даного методу є те, що елемент, який видаляється, можна покласти в змінну :

`my_var = my_list.pop(index)`

Python Starter

Операції зі списками

- Для копіювання списку використовується метод `copy()`:
`my_list2 = my_list1.copy()`
- Для вставки елемента у список використовується метод `insert()`:
`my_list.insert(index, value)`
- Для підрахунку кількості певних значень у списку використовується метод `count()` :
`my_list.count(value)`
- Для отримання порядкового номера першого елемента зі значенням `value` в діапазоні `від` та `до`:
`my_list.index(value, lower_range_index, upper_range_index)`
- Для склеювання елементів списку в рядок використовується метод `join()`:
 - `char_list = ["a", "e", "i", "o", "u", "y"]`
 - `str_list = "; ".join(char_list)`

Python Starter

Операції зі списками

- Для очищення списку використовується метод `clear()`:
`my_list.clear()`
- Для сортування списку за зростанням використовується метод `sort()`, який змінює вихідний:
`my_list.sort()`
- Для сортування списку за спаданням використовується метод `sort()`, який змінює вихідний:
`my_list.sort(reverse=True)`
- Для сортування списку за зростанням використовується метод `sorted()`, який повертає новий:
`my_list2 = sorted(my_list1)`
- Для сортування списку за спаданням використовується метод `sorted()`, який повертає новий:
`my_list2 = sorted(my_list1, reverse=True)`

Python Starter

Операції зі списками

- Для виведення списку в оберненому порядку використовується метод `reverse()`, який змінює вихідний:
`my_list.reverse()`
- Для виведення списку в оберненому порядку використовується метод `reversed()`, який повертає ітерований об'єкт, тому що побачити список треба залучити функцію `list()`:
`print(list(reversed(my_list)))`

Python Starter

Обхід списку

Для перебору елементів списку використовується цикл for:

```
for element in my_list:  
    # ...  
    # виконання операцій з element  
    # ...  
pass
```

Дивіться наші уроки у відео форматі

ITVDN.com



Перегляньте цей урок у відео форматі на освітньому порталі [ITVDN.com](http://itvdn.com) для закріплення пройденого матеріалу.

Усі курси записані сертифікованими тренерами, які працюють у навчальному центрі CyberBionic Systematics



Перевірка знань

TestProvider.com

TestProvider

Мы помогаем людям оценить себя

Регистрация Войти

Поиск сертификата

Главная Услуги и цены Центр Тестирования Поддержка О нас

Мы в социальных сетях

Тестирование

Языки программирования и информационные технологии

Microsoft

C# ASP.NET MVC JavaScript Patterns OF Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Добро пожаловать на TestProvider.com!

Сайт перенесен на новую облачную платформу с использованием системы единой авторизации Single Sign On. Если вы хотите восстановить статистику по предыдущим экзаменам обратитесь в [службу поддержки](#). Для восстановления информации с предыдущей версии сайта, просба написать в службу поддержки Ваш старый и новый логины.

ITVDN PROMETRIC TEST CENTER CyberBionic Microsoft Partner Windows Azure Cloud Partner EBA

TestProvider – це online сервіс перевірки знань з інформаційних технологій. За його допомогою Ви можете оцінити Ваш рівень та виявити слабкі місця. Він буде корисним як у процесі вивчення технології, так і для загальної оцінки знань IT спеціаліста.

Після кожного уроку проходите тестування для перевірки знань на TestProvider.com

Успішне проходження фінального тестування дозволить Вам отримати відповідний Сертифікат.



Python Starter

Q&A

Python Starter

Дякую за увагу!

Інформаційний відеосервіс для розробників програмного забезпечення

