

Python Essential

Модулі та пакети

Python

Після уроку обов'язково



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



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

Python Essential

Тема

Модулі та пакети

Модулі

Поняття модулів

Модуль – функціонально закінчений фрагмент програми, оформлений у вигляді окремого файлу з вихідним кодом або поіменованої безперервної її частини. Модулі дозволяють розбивати складні завдання на дрібніші відповідно до принципу модульності.

Зазвичай проектується в такий спосіб, щоб надавати програмістам зручну для багаторазового використання функціональність (інтерфейс) у вигляді набору функцій, класів, констант.

Файл, який містить вихідний код мовою Python, є модулем.



Модулі

Імпортування модуля

```
import module_name  
import module_name as new_name
```



Початкова назва імпортованого модуля доступна як його глобальна змінна `_name_`. У модуля, запущеного як скрипт, `_name_` дорівнює `"_main_"`.



Модулі завантажуються лише один раз. Усі подальші спроби їх імпортувати лише повертають посилання на вже завантажені модулі.

Модулі

Імпортування імен із модуля

```
from module import name
from module import name1, name2
from module import *

from module import name as new_name
from module import name1 as new_name1, name2 as new_name2
```



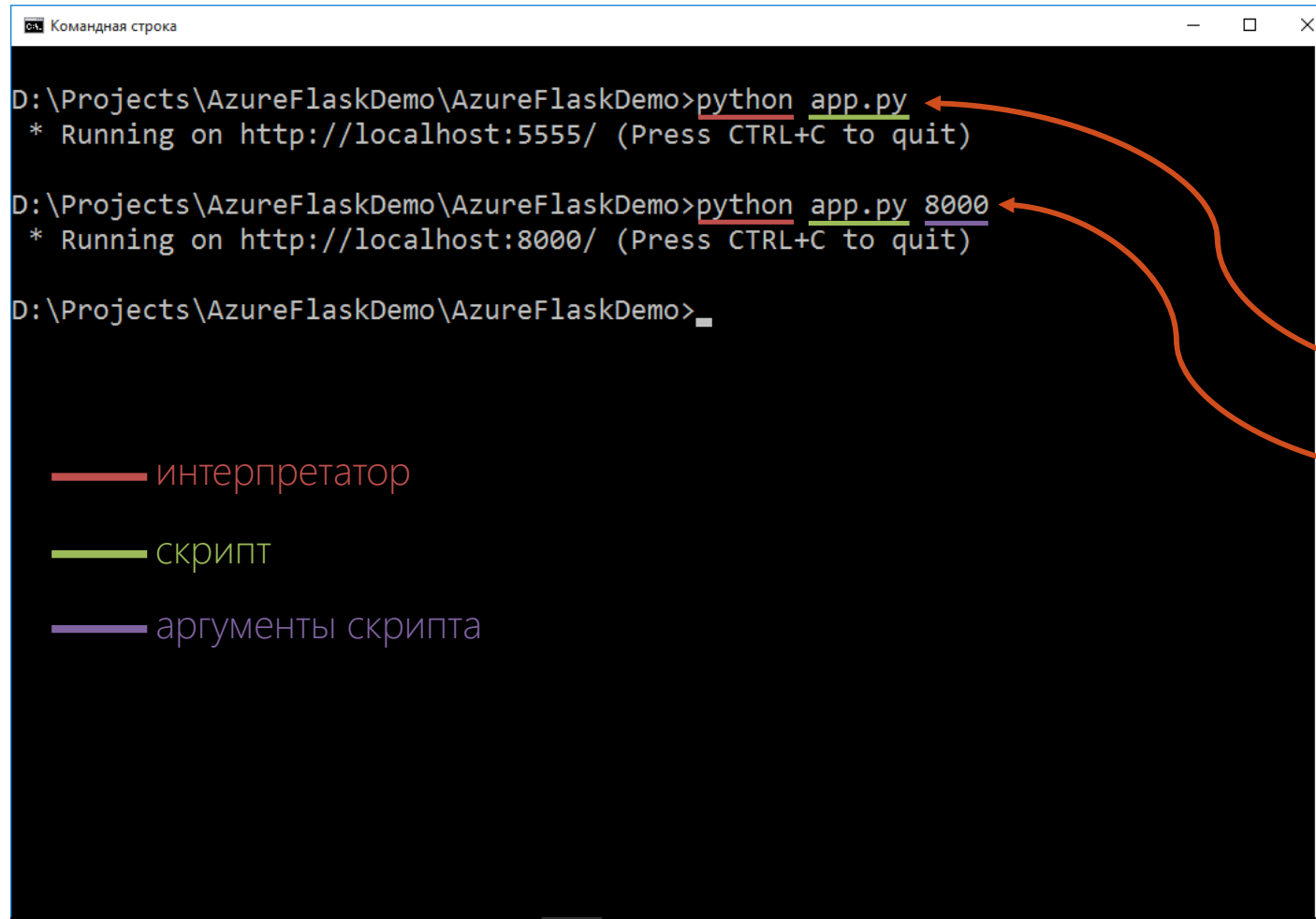
Якщо замість списку імен вказати символ *, то імпортуються всі імена, крім тих, що починаються з символу підкреслення.



Імпортування всіх імен (from module import *) зручне в інтерактивній сесії інтерпретатора, але небажане в коді

Модулі

Запуск модуля як скрипта



```
Командная строка
D:\Projects\AzureFlaskDemo\AzureFlaskDemo>python app.py
* Running on http://localhost:5555/ (Press CTRL+C to quit)
D:\Projects\AzureFlaskDemo\AzureFlaskDemo>python app.py 8000
* Running on http://localhost:8000/ (Press CTRL+C to quit)
D:\Projects\AzureFlaskDemo\AzureFlaskDemo>
```

- интерпретатор
- скрипт
- аргументы скрипта

Глобальна змінна `name` ініціалізується значенням `"_main_"`, а не `"app"`, оскільки скрипт запущено, а не імпортовано як модуль.

В першому прикладі
`sys.argv = ['app.py']`

В другому прикладі
`sys.argv = ['app.py', '8000']`

Модулі

Шляхи пошуку модулів

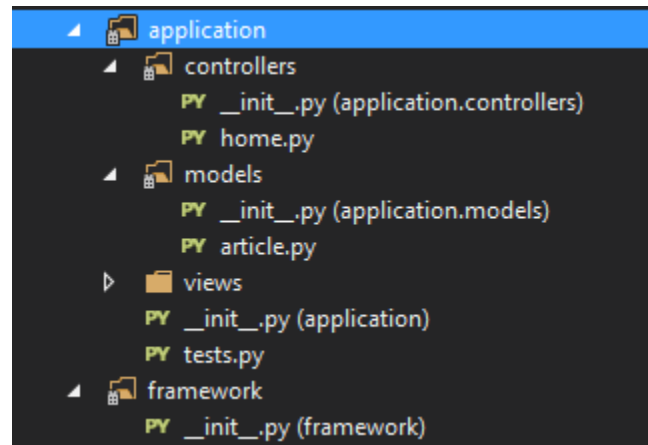
- При імпортуванні модулів інтерпретатор Python шукає їх у директоріях та архівах, список яких доступний як для читання, так і для модифікації у вигляді змінної `path` вбудованого модуля `sys`.
- За замовчуванням **`sys.path`** складається з директорії зі скриптом, що запускається, вмісту змінної оточення `PYTHONPATH` і стандартного розташування модулів, специфічного для конкретної платформи та інтерпретатору.
- Для прискорення завантаження модулів Python кешує байт-код і здійснює компіляцію модуля тільки в тому випадку, якщо вихідний код було змінено. Python 3 зберігає файли байт-коду `.pyc` у каталозі `__pycache__`.



Модулі

Пакети

- Модулі можуть об'єднуватись у пакети. Пакети служать як простір імен для модулів та спосіб їх структурування.
- Будь-який пакет є модулем, але не кожний модуль є пакетом.
- Як правило, модулі подаються у вигляді файлів, а пакети - каталогів у файловій системі (але не завжди).
- Для того, щоб каталог був пакетом, у ньому має бути файл `__init__.py`. Він автоматично виконується під час імпортування відповідного модуля та може містити певні дії для ініціалізації або бути порожнім.



Модулі

Імпортування з пакетів

```
import package.module  
import package.subpackage.module  
from package import module  
from package import item  
from package.subpackage import module, item  
from module import *
```



Щоб можна було імпортувати всі імена пакету (`from package.subpackage import *`), пакет повинен описувати список `_all_`, який містить імена підпакетів та модулів.



При використанні оператора `from package import item`, `item` може бути пакетом, модулем або будь-яким ім'ям, описаним у пакеті. При використанні оператора `import package.item`, `item` має бути модулем або пакетом.

Модулі

Відносне імпортування

Крім абсолютних, існують також відносні імпорти: точка вказує на поточний пакет, дві точки – на батьківський. Ці ж символи можуть бути використані відразу перед ім'ям пакету або модулю і впливати на те, де інтерпретатор його шукатиме.

```
from . import name  
from .. import name  
from .package import name  
from ..package import name
```



Дивіться наші уроки у відео форматі

ITVDN.com



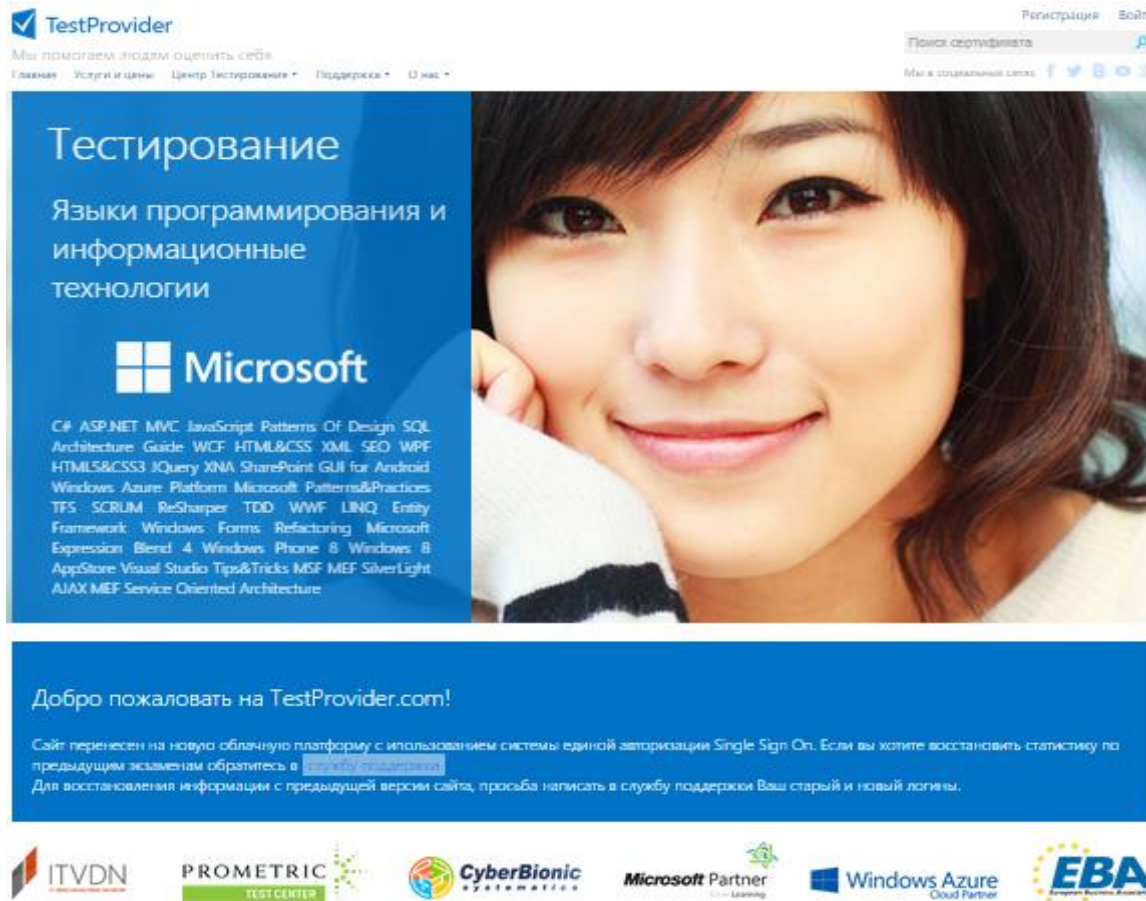
Перегляньте цей урок у відео форматі на освітньому порталі [ITVDN.com](http://itvdn.com) для закріплення пройденого матеріалу.

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



Перевірка знань

TestProvider.com



TestProvider – це online сервіс перевірки знань з інформаційних технологій. З 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 Essential

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

