



# Python Advanced

Робота з мережею (Socket та протокол HTTP)

# Python Advanced

Після уроку обов'язково



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



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

# Python Advanced

## Зміст уроку

1. Основи мережевої моделі OSI та її рівнів.
2. Поняття адресації (IP та Port) та socket.
3. Опис протоколу UDP.
4. Створення UDP клієнта/сервера.
5. Створення TCP клієнта/сервера та порівняння з UDP.
6. Блокувальні та не блокувальні режими роботи socket.
7. Підхід ООП під час створення socket-серверів, використовуючи бібліотеку socketserver.
8. Створення простого socket-сервера засобами фреймворку Twisted.
9. Що таке протокол HTTP, як використовувати та основні типи запитів.
10. Поняття заголовків і статус-кодів.
11. Типи HTTP-запитів та їхні особливості.
12. Створення socket для демонстрації заголовків і відповідей сервера.
13. Стандартна бібліотека мови Python – urllib.
14. Бібліотека requests.
15. Конфігурація бібліотеки urllib: розмір, pull з'єднань і режими роботи.

## Робота з мережею (socket)

# Основи роботи з мережею

## Адресація та основні поняття

- Мережева модель OSI.
- Рівні моделі.
- Основні поняття.

# Основи роботи з мережею

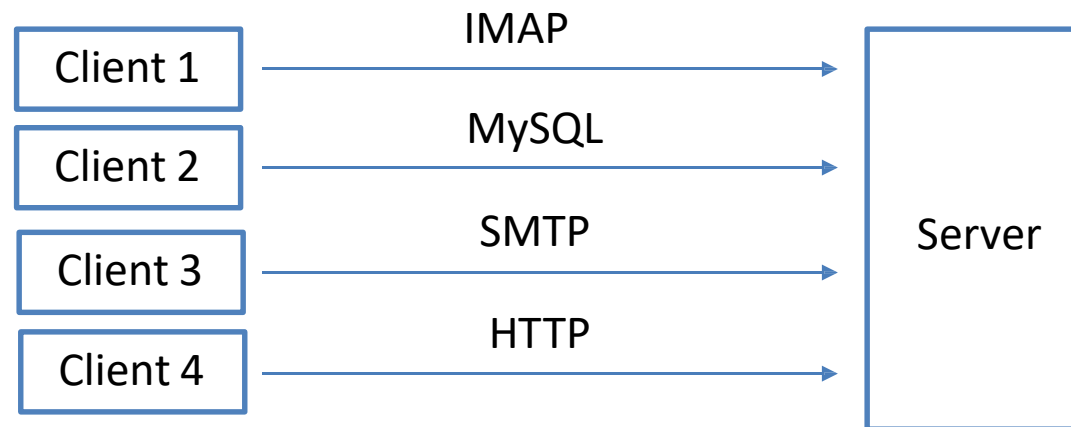
## Модель OSI

- Фізичний рівень.
- Канальний рівень.
- Мережевий рівень.
- Транспортний рівень.
- Сеансовий рівень.
- Представницький рівень.
- Прикладний рівень.

# Основи роботи з мережею

## Адресація компонентів мережі

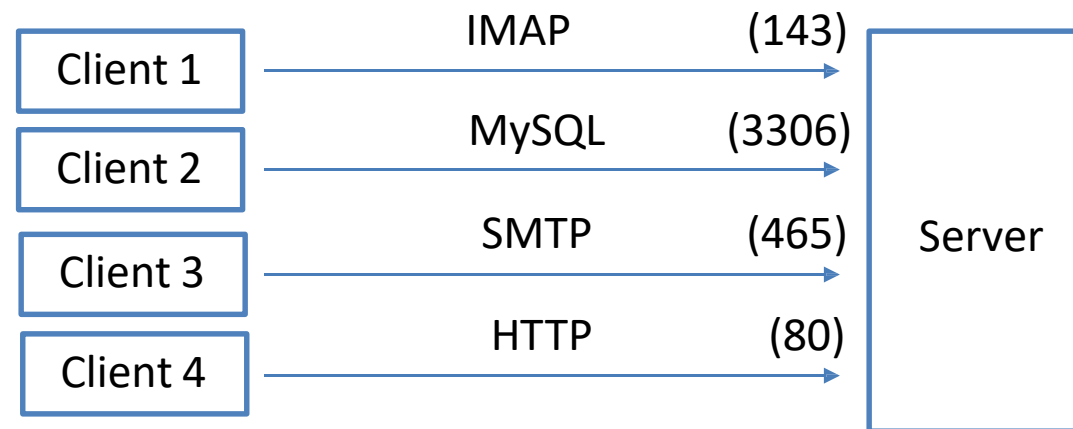
- Безліч сервісів на одному сервері.
- Адресація.



# Основи роботи з мережею

## Адресація компонентів мережі

- Доступний діапазон портів 0 – 65535.
- Можливість зайняти порт.

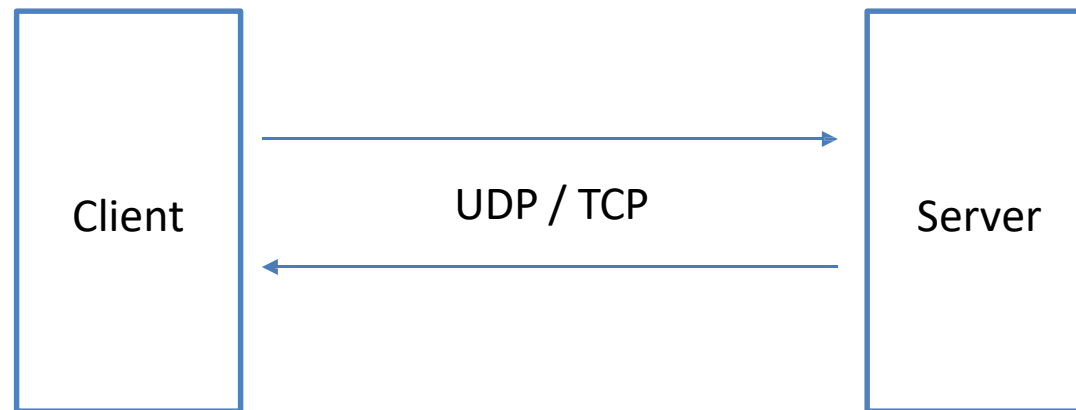




# Основи роботи з мережею

## Мережеві протоколи

- UDP (user datagram protocol).
- TCP (transmission control protocol).



# Основи роботи з мережею

## UDP-протокол

- Ненадійний протокол.
- Порядок пакетів може бути порушено.
- Відсутність «рукоштовування».



# Основи роботи з мережею

## ТСР-протокол

- Надійний протокол.
- Порядок пакетів дотримується.
- «Рукоштовування» перед встановленням з'єднання.
- Обмін даними.



# Основи роботи з мережею

## Socket у Python

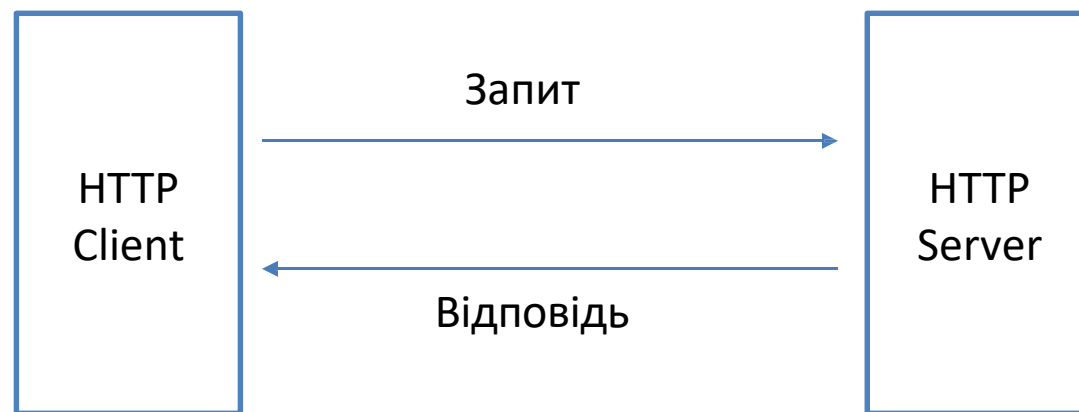
- Бібліотека socket.
- Бібліотека socketserver.
- Фреймворки.

## Робота з мережею (протокол HTTP)

# Протокол HTTP

## Що таке HTTP?

- Протокол прикладного рівня.
- Клієнт/Сервер.
- Ідентифікація ресурсів.
- Структура протоколу.



# Протокол HTTP

## Формат протоколу

- Заголовки.
- Статус-коди.
- Запит/Відповідь.

### *Запит*

```
GET / HTTP/1.0  
Host: example.com
```

### *Відповідь*

```
HTTP/1.1 200 OK  
Date: Wed, 30 Apr 2017 12:20:59 GMT  
Server: nginx  
Content-Language: ru  
Content-Type: text/plain; charset=utf-8  
Content-Length: 128  
Connection: close
```

Text data

# Протокол HTTP

## Розшифрування статус-кодів

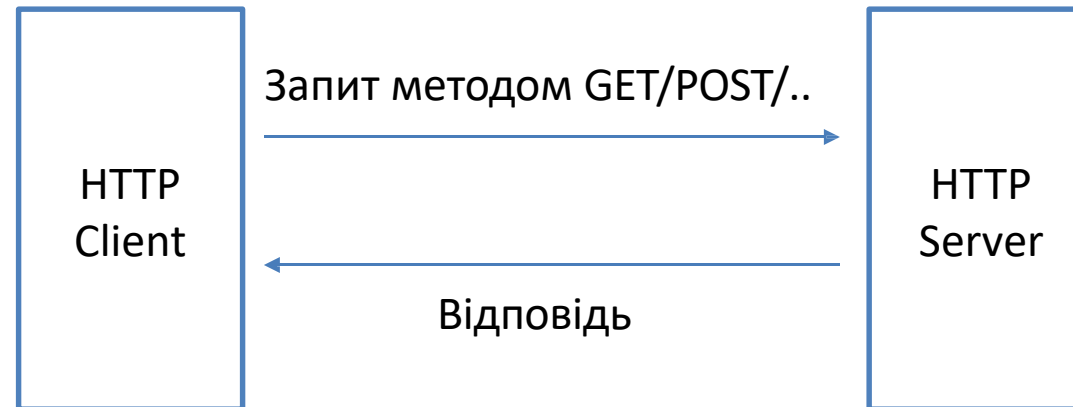
Коди	Значення
1x	Інформаційні
2x	Успіх виконання
3x	Перенаправлення
4x	Помилки на стороні клієнта
5x	Помилки на стороні сервера



# Протокол HTTP

## Типи запитів

- GET
- POST
- PUT
- DELETE
- PATCH
- OPTIONS
- HEAD
- etc.



# Протокол HTTP у Python

## HTTP у Python

- Бібліотека socket.
- Бібліотека urllib.
- Бібліотека requests.



# Python Advanced

Q&A

# Інформаційний відеосервіс для розробників програмного забезпечення

