

Заняття №30. Мережеве програмування. Інтернет  
Корисні посилання (обов'язкові до опрацювання):

<https://realpython.com/python-sockets/>

[https://www.tutorialspoint.com/python/python\\_networking.htm](https://www.tutorialspoint.com/python/python_networking.htm)

<https://habr.com/ru/company/alfa/blog/354728/>

[Echo client-server example](#)

[Модель OSI коротке відеопояснення](#)

[TCP vs UDP коротке відеопояснення](#)

[HTTP пояснення](#)

[nginx vs Apache відеорозбір і порівняння](#)

[socketserver](#)

Правила здачі:

- Назва файлу має відповідати назві завдання (програми), якщо явно не вказано протилежне. Файли з кодом мають бути з розширенням тої мови на якій написати код (.py, ...)
- Форматування файлу має відповідати [PEP8](#).
- Файли з рішеннями потрібно завантажити на ваш git репозиторій з назвою **itstep**, в окрему папку з назвою **lesson30**. Репозиторій має бути публічним, як це перевірити можна дізнатися в офіційній документації вашого git провайдера, або за допомогою Google.
- якщо ви хочете тестувати код в тому ж файлі, що пишете помістіть виклики в конструкцію  
**if \_\_name\_\_ == "\_\_main\_\_":**  
...
- Посилання на репозиторій помістіть в окремий файл під назвою **solution\_lesson30\_<your\_login>** де **<your\_login>** це ваш студентський логін та прикріпіть як рішення в mystat. Уважно перевірте, що ви прикріплюєте саме рішення до поточного заняття!
- за будь-яку невідповідність до вищезазначених пунктів оцінка знижується.

### Завдання.

1. Напишіть клієнт-серверний варіант консольної гри в хрестики нолики за допомогою сокетів, що дозволяє двом людям грати в гру. Той хто закриває програму під час гри вважається програвшим. Після завершення гри можна ініціювати повторну гру.
2. Реалізуйте для попередньої гри графічний інтерфейс за допомогою pygame.
3. Підготуйтеся до заняття по темі SQL та реляційні бази даних.

<https://habr.com/ru/post/481084/>