

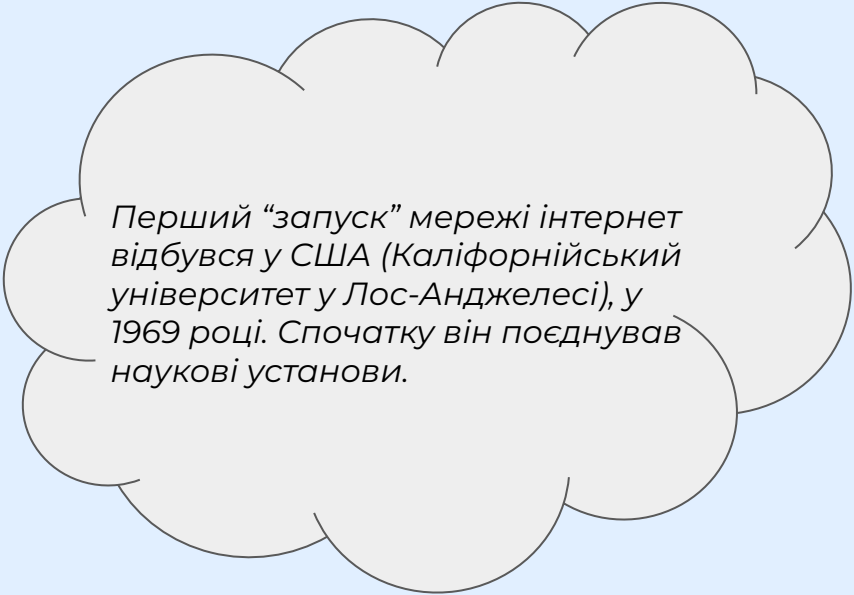
# ОСНОВИ WEB

# Хто такий UX/UI дизайнер

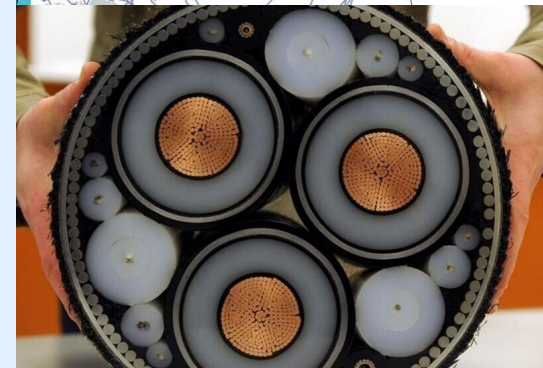
**Інтернет** – всесвітня система об'єднаних комп'ютерних мереж для зберігання та передачі інформації.

На основі інтернету працює "Всесвітня павутина" та безліч інших систем передачі даних.

**Всесвітня павутина** (World Wide Web, WWW) — розподілена система, що надає доступ до пов'язаних (по гіперпосиланням) між собою документів, розташованих на різних комп'ютерах, підключених до Інтернету.



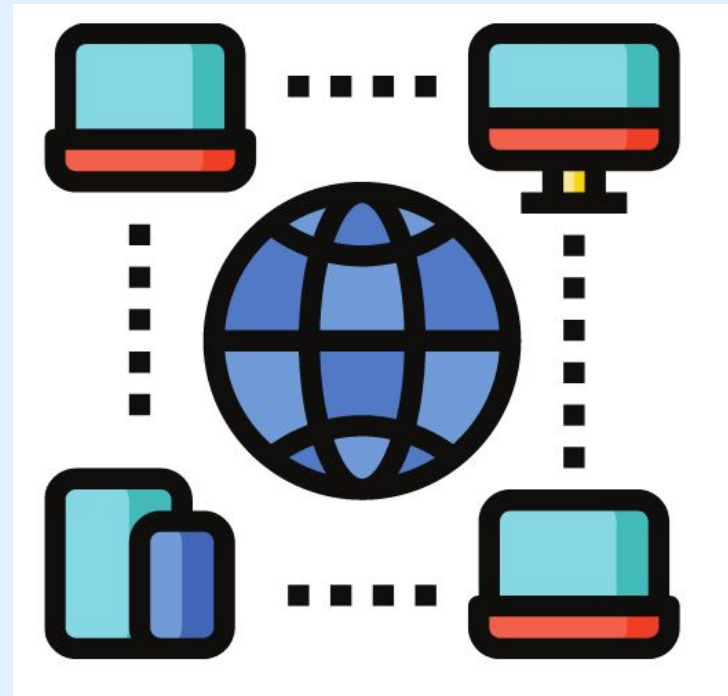
*Перший "запуск" мережі інтернет відбувся у США (Каліфорнійський університет у Лос-Анджелесі), у 1969 році. Спочатку він поєднував наукові установи.*



# Мережевий протокол

- набір правил і дій (черговість дій), що дозволяє здійснювати з'єднання та обмін даними між двома та більше включеними в мережу пристроями.

**Інтернет мережа побудована з урахуванням стека протоколів TCP/IP.**



# TCP/IP

4

Прикладний

Безліч протоколів прикладного рівня, що виконують конкретні прикладні завдання: **HTTP, FTP, SMTP, SSH, DNS** та ін.

3

Транспортний

Протокол **TCP** використовується при передачі даних. Контролює довжину повідомлення, швидкість обміну, трафік, здійснює повторний запит даних у разі втрати та усуває дублювання.

2

Мережевий

Протокол **IP** визначає структуру мережі та доставку пакетів.  
Але не гарантує надійної доставки пакета до адресата.

1

Канальний

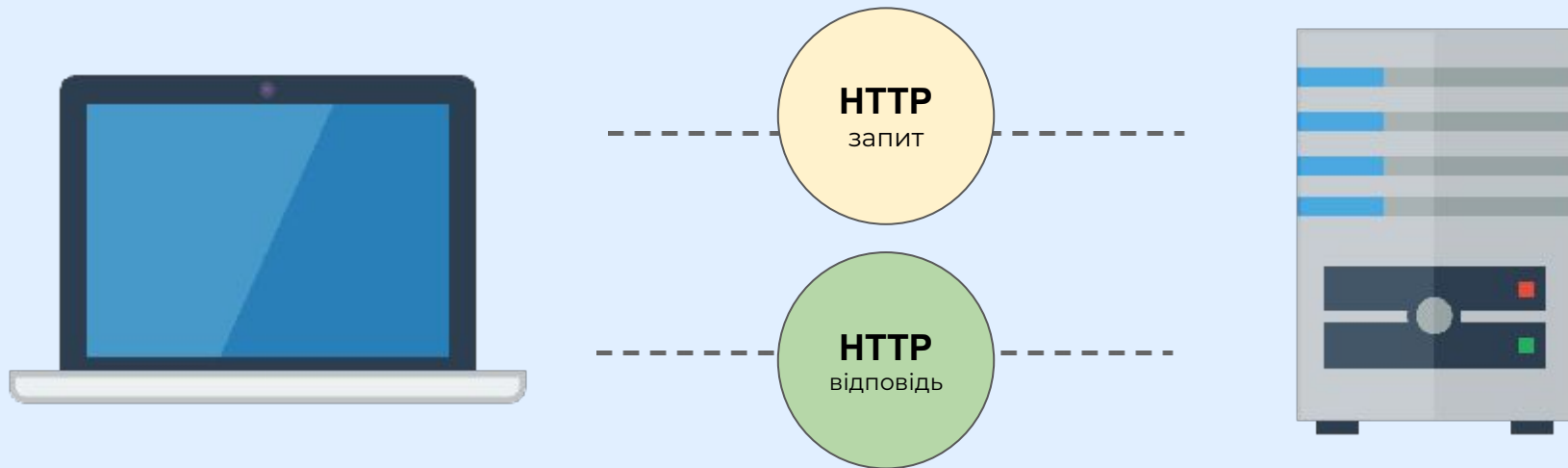
Фізичне середовище та принципи кодування інформації. Наприклад, інтерфейс Ethernet, що описує передачу даних по кручений парі.

# TCP/IP vs OSI

TCP/IP model	Protocols and services	OSI model
Application	HTTP, FTP, Telnet, NTP, DHCP, PING	Application
		Presentation
		Session
Transport	TCP, UDP	Transport
Network	IP, ARP, ICMP, IGMP	Network
Network Interface	Ethernet	Data Link
		Physical

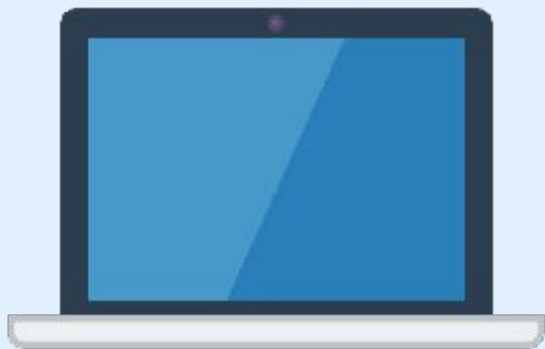
# HTTP: HyperText Transfer Protocol

**Протокол прикладного рівня** передачі даних. Основним об'єктом маніпуляції в **HTTP** є ресурс, на який вказує **URI** (Universal Resource Identifier) у запиті клієнта. Зазвичай такими ресурсами є файли, що зберігаються на сервері.

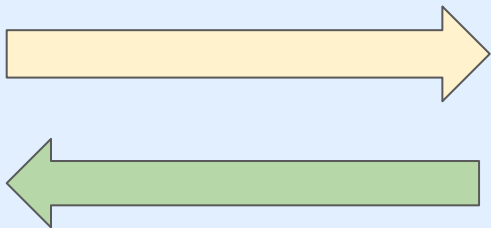


# Клієнт-сервер

— це мережна архітектура, в якій завдання або мережеве навантаження розподілені між постачальниками послуг, які називаються **серверами**, та замовниками послуг, які називають **клієнтами**.



Клієнт



Сервер



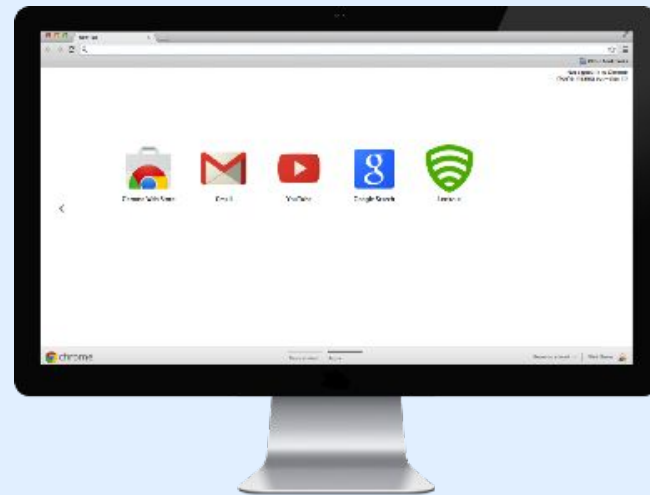
# Клієнт, веб-браузер та frontend

Для перегляду інформації, отриманої від веб-сервера, використовується **браузер**.

Основні функції:

- встановлення зв'язку з веб-сервером, на якому зберігається документ, та завантаження всіх його компонентів;
- форматування та відображення веб-сторінок;
- пошук веб-сторінок та спрощення доступу до сторінок, що відвідувалися раніше;

**Frontend** – інтерфейс взаємодії між користувачем та **backend**.



# Сервер, веб-сервер та backend

**Сервер** - це комп'ютер із встановленими на ньому програмами, що виконують різні (серверні) завдання. Одна з таких програм називається "веб-сервер".

**Веб-сервер** — це комп'ютерна програма, яка запускається на підключеному до мережі комп'ютері та використовує протокол HTTP для передачі даних.

*Часто, для спрощення, терміни “веб-сервер” і “сервер” одночасно мають на увазі і програму і сам комп'ютер, на якому вона встановлена.*

**backend** - це серверна сторона сайту, написана серверними мовами програмування: **C#, java, node.js, python**, які, у тому числі, забезпечують взаємодію з базою даних.



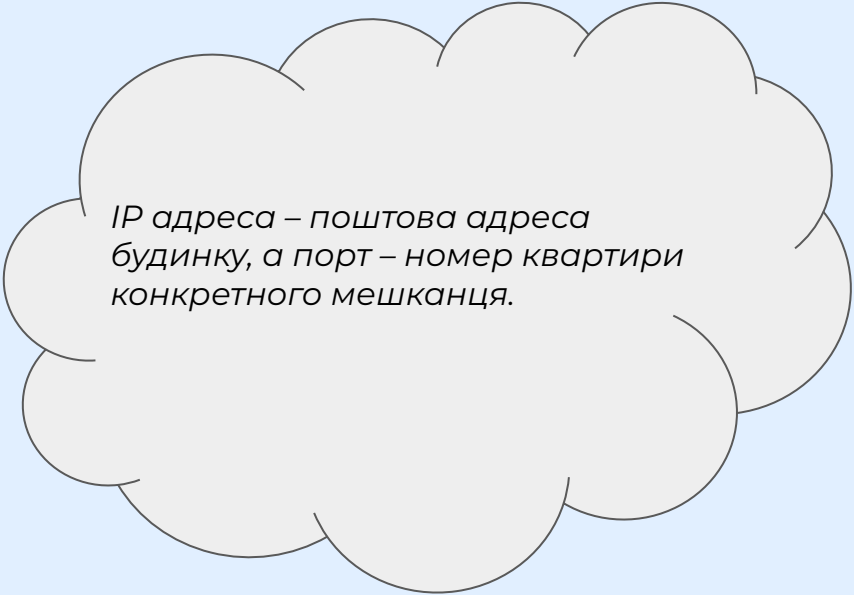
# Мережевий порт

У TCP пакетах вказуються *порт джерела* та *порт призначення*.

Мережевий порт - умовне число від 1 до 65535, що вказує, якому додатку призначається пакет.

У кожному пакеті присутні IP-адреса вузла-джерела та IP-адреса вузла-призначення.

За замовчанням у протоколі **HTTP** використовується порт **80**, а протоколі **HTTPS** — порт **443**

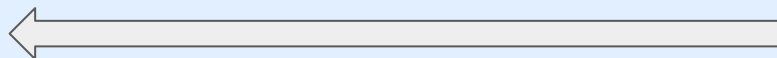
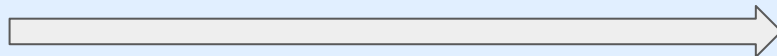


*IP адреса – поштова адреса будинку, а порт – номер квартири конкретного мешканця.*

# DNS

— комп'ютерна розподілена система для отримання інформації про домени. Найчастіше використовується для отримання IP-адреси за ім'ям хосту (доменному імені).

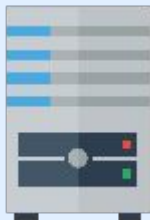


**Клієнт****2****HTTP запит:** 172.67.223.5**HTTP відповідь:** HTML, CSS,  
JavaScript, медіа-файли**Сервер****3****Програми на сервері:**

Веб-сервер: порт 443  
PostgreSQL  
FastAPI або Django  
uvicorn або nginx

**1**<https://goit.global>

172.67.223.5

**DNS**

# Основні частини HTTP запиту

## URI / URL

Адреса, яку ми набираємо в браузері

`https://goit.global/`

## HTTP METHOD

Послідовність із будь-яких символів, що вказує на основну операцію над ресурсом

GET, POST, PUT, DELETE

## HTTP STATUS CODE

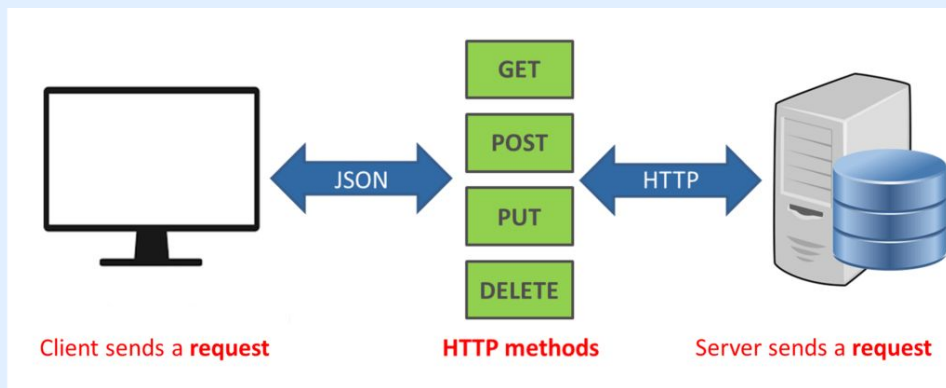
Код відповіді сервера на запитану операцію **200**

**OK, 404 Not Found**



# HTTP-методи GET/POST/PUT/DELETE

- **GET** просить подання ресурсу. Запити з цього методу можуть лише витягувати дані.
- **POST** використовується для надсилання сутностей до певного ресурсу. Часто викликає зміну стану або якісь побічні ефекти на сервері.
- **PUT** замінює всі поточні уявлення ресурсу даними запиту.
- **DELETE** видаляє вказаний ресурс.
- **PATCH** використовується для часткової зміни ресурсу.



# Хостинги

## Хмарний хостинг



## Віртуальний виділений сервер (VPS/VDS)

- + гарантована частина ресурсів сервера;
- + можливість встановлювати програмне забезпечення.
- невисока потужність через обмеження ресурсів, що виділяються;
- турбота про налаштування та роботу ПЗ переходить на бік клієнта;
- платна технічна підтримка;
- дорожче за віртуальний хостинг.



*\*VPS та VDS - це різні назви одного і того ж*



# HTML

**HTML** – декларативна мова програмування. Використовується для розмітки документів в Інтернеті. HTML “рендерується” (відображається, виконується) у браузерях. Таким чином, може відображатися на всіх пристроях (мобільні, планшети, стаціонарні комп'ютери тощо), де є встановлений браузер.

```
<!DOCTYPE html>
<!-- доктайп (зазначення специфікації) -->
<html lang="ru">
  <!-- відкриваючий тег HTML -->
  <head>
    ...
    <!-- налаштування документа, підключення стилів відображення-->
  </head>
  <body>
    ...
    <!-- тіло документа -->
  </body>
</html>
```

**Tag** - елемент розмітки, що може застосовуватися з налаштуваннями (атрибутами) або без них. Текст або інший контент "всередині" тега відображається в залежності від типу, і назви тега а також залежать від того які атрибути "стоять" у тега.

```
<tagname attrname="attrprop"> tag content </tagname>
```

```
<tagname attrname="attrprop" />
```

# Основні частини HTML документу

## **<doctype>**

вказівка специфікації HTML

## **<html>**

тег-контейнер для “голови” та “тіла” документа. Більше ніяких елементів в середині нього лежати не повинно.

## **<head>**

тег, що візуально не відображається (не рендерується) браузером. Цей тег містить параметри документа, метатеги, теги для розпізнавання seo, заголовки сторінки, підключення стилів тощо.

## **<title>**

тег, що вказує браузер на назву сторінки. Яке відобразиться у заголовку вкладки.

## **<body>**

тег визначальний "тіло" документа. Саме в цей тег поміщаються вся решта розмітки та підключення скриптів JavaScript.



# Посилання

Для того щоб переходити від одного документа до іншого існує тег посилання - **<a>** (Anchor Tag)

```
<a href="https://google.com"> посилання на гугл </a>
```

# Атрибути посилання

**href** - адреса, куди веде посилання. Може бути стороннім ресурсом або локальним файлом (так само може бути вказаний ідентифікатор блоку на сторінці, до якого варто прокрутити сторінку).

**title** - додає підказку для посилання.

**download** - вказує, що файл за посиланням потрібно не відкрити, а скачати.

**target** - вказує у якому вікні відкрити посилання (у поточному чи новому).

```
<a href="http://google.com" target="_blank"> посилання на гугл </a>
```

# Шляхи посилання

Шляхи бувають абсолютними та відносними.

**Абсолютний шлях** - шукає файл від "кореневої папки" залежно від конфігурації сервера (починається зі слешу "/").

```
<a href="/pages/contacts.html"> Контакти </a>
```

**Відносний шлях** - шукає файл від поточного файлу, де прописаний шлях (починається з точок).

```
<a href="../pages/contacts.html"> Контакти </a>
```

# CSS



# CSS

**CSS** (каскадні таблиці стилів) - цією мовою описано те, як буде представлений HTML. Як і html це декларативна мова, тобто в ній немає алгоритмічної складової, а лише опис виду чи поведінки зазначених елементів.



Правила CSS завжди передаються у вигляді пари – назва властивості та його значення, що розділені двокрапкою. Кожне правило обов'язково має закінчуватися крапкою з комою.

```
key: value;
```

CSS селектор - вказівка до якого саме елемента або набору елементів буде використано стилі.

```
a {  
  font-size: 16px;  
}
```

# Вибірка елементів

# Вибірка за тегом

під цей селектор потрапляють всі теги, вказані в селекторі

html

```
<h1>буде стилізовано</h1>  
<h1>І це теж потрапить</h1>  
<h3>а цей залишиться не  
змінним</h3>
```

CSS

```
h1 {  
  color: crimson;  
}
```

# Вибірка за id

**id** – унікальний ідентифікатор елемента. Застосовується як атрибут до будь-якого тегу. Може бути використаний на сторінці лише один раз! CSS селектор починається зі знака "шарп" - #

html

```
<h1 id="unique">  
  я унікальний  
</h1>
```

CSS

```
#unique {  
  color: crimson;  
}
```

# Вибірка за класом

**class** - селектор який, на відміну від id, може бути застосований до будь-якої кількості елементів.

Завдяки цьому відбувається успадкування стилів. CSS селектор починається з точки

html

```
<p class="same-selector">Текст параграфа.</p>

<a class="same-selector">
  Посилання, яке намагається бути параграфом
</a>
```

css

```
.same-selectors {
  font-size: 13px;
  color: red;
}
```

# parent та child

## html

```
<div class="luke">
    Незалежний Люк!
</div>

<div id="vader">
    Люк я твій батько.

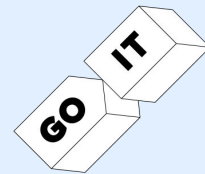
    <div class="luke">
        Так і знав, що я твій син!
    </div>
</div>
```

## CSS

```
.luke {
    font-size: 13px;
    color: red;
}

#vader .luke {
    color: blue;
}
```





# Дякую!

