# 10. Фронтенд-розробка. Ключові технології та поняття

Фронтенд веб-сайту це все, що користувач бачить і з чим може взаємодіяти за допомогою браузера. Створення цієї візуальної частини називається фронтенд-розробкою. Дизайнерів, що створюють інтерфейси користувача, теж можна в якомусь сенсі назвати фронтенд-розробниками, оскільки вони спільно працюють над цією ж частиною проекту.

Для розробки фронтенду як базові інструменти використовуються: HTML (для створення базової структури сторінок та контенту), CSS (для стилізації зовнішнього вигляду) та JavaScript (для додавання інтерактивності). Такий же набір інструментів використовується в процесі створення прогресивних веб-додатків - мобільних додатків, які виглядають як нативні, але при цьому створюються за участю фронтенд-технологій.

Бекенд — це серверна частина веб-застосунку, прихована від очей користувача. Це поняття включає сервери, на яких розташовані веб-сторінки і певну логіку, яка управляє функціями і процесами сайту.

Бекенд розробляється з використанням іншого стеку технологій, включаючи мови серверного програмування Java, PHP, Ruby, C# та бекенд-фреймворків.

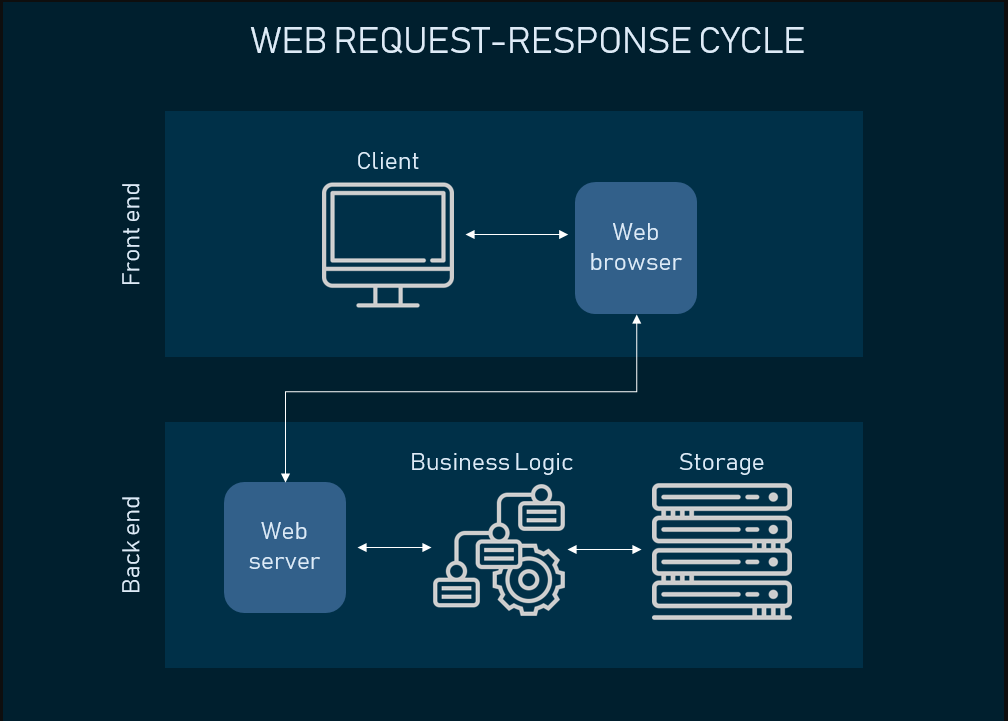


Рис.1. Цикл запит-відповідь

Отже, базовий набір інструментів для розробки фронтенду чітко визначений: HTML, CSS та JavaScript. Однак цей набір може бути значно розширений, включаючи диспетчери пакетів, CSS-препроцесори, фреймворки і багато іншого.

## HTML: ключова фронтенд-технологія

HTML (Hypertext Markup Language) - це мова гіпертекстової розмітки, призначена для створення веб-сайтів, які згодом можуть переглядатися за допомогою доступу до інтернету. HTML зазвичай використовується для структурування веб-документа. Він визначає такі елементи, як заголовки або абзаци, і дозволяє вставляти зображення, відео та інші медіафайли.

### Як працює HTML.

HTML-код це множина тегів, що пишеться і зберігається в форматі HTML-файлу, який можна відкрити і подивитися в браузері. Браузер сканує та інтерпретує код у візуальну форму і відображає сторінку саме так, як задумав дизайнер.

Розмітка визначає властивості, якими наділяється текст всередині HTML-тегів. Теги визначають, як браузери форматують та відображають вміст сторінки. Будучи мовою, він містить кодові слова та синтаксис, як і будь-яку іншу мову.

Наприклад:

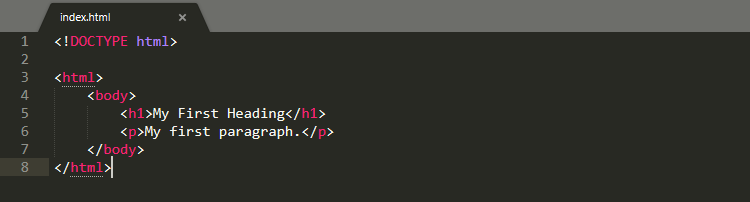


Рис.2. Приклад HTML-коду

Парний тег <html></html> визначає межі веб-сторінки, а текст між тегами <body></body> визначає видимий вміст сторінки.

<!DOCTYPE html> на початку оголошує тип документа для HTML5. Якщо його не позначити, різні браузери відображатимуть його по-своєму.

### HTML5

З моменту першого свого релізу в 1991 році HTML зазнав багато оновлень. HTML5 випущено у 2014 році. До нього були додані такі функції, як підтримка оффлайн сховищ мультимедійних даних, точніші елементи контенту (наприклад, хедер, футер, навігація) та підтримка вбудовування аудіо та відео.

## CSS: стилізація

CSS (Cascading Style Sheets) - це мова, яка використовується для стилізації сторінок. Він визначає те, як HTML-елементи виглядатимуть на веб-сторінці з погляду дизайну, макета на різних пристроях з різними розмірами екрану. CSS керує макетом різних веб-сторінок одночасно. CSS взаємодіє з HTML-елементами, компонентами веб-сторінки.

* Селектор — це частина CSS-коду, яка визначає, на яку частину HTML впливатимуть стилі CSS.
* Оголошення містить властивості та значення, що використовуються селектором.
* Властивості визначають розмір шрифту, колір та відступи. Кожна властивість має значення чи набір значень.

Візьмемо як приклад:



Рис.3. Приклад CSS -коду

де р (абзац) - це селектор, {font-size:24px; color:blue;} - це оголошення, font-size: і color: - властивості, а 24px; та blue; - значення.

CSS записується у вигляді простого тексту в текстовому редакторі. Є три способи додати CSS-код у HTML:

* Замість того, щоб додавати CSS-код до кожного HTML-елемента, який потрібно змінити, можна використовувати зовнішню таблицю стилів (External style sheets) — текстовий файл, що містить весь CSS-код. Зовнішня таблиця стилів підключається до .html файлу в контейнері <head>.
* При використанні внутрішньої таблиці стилів CSS-код розміщується прямо в тілі тега <head> конкретної html-сторінки.
* Під вбудованим стилем мається на увазі прописування CSS в HTML-коді безпосередньо того елемента, який потрібно стилізувати.

### CSS-фреймворки.

Фреймворк CSS – помічник будь-якого розробника та дизайнера у багатьох аспектах, навіть для тих, хто любить писати проекти на чистому CSS. Багато, з наведених нижче фреймворків можуть бути прогресивно застосовані, оскільки є можливість комбінування власного коду CSS та коду фреймворку. Майже всі найкращі фреймворки в 2023 році можуть бути застосовані незалежно від досвіду розробників. Вони підходять для початківців та розвинених користувачів.

CSS-фреймворк підтримує структуру коду розробників і слідує кращим практикам, що важливо при розвитку та зростанні всього проекту. Також, надає широкий набір елементів інтерфейсу користувача. CSS-фреймворки часто використовують розробники та дизайнери, з їх допомогою вдається реалізувати будь-які функції, додати візуальні частини всередині сайту у вигляді форм, кнопок, панелей, хлібних крихт тощо.

За допомогою фреймворків CSS спрощується створення сайту, забезпечується кросбраузерність у популярних браузерах, скорочується ймовірність помилок під час тестування.

За рахунок наявності готових таблиць стилів фреймворки допомагають значно спростити веб-розробку, прискорити процедуру реалізації проекту. Користувачам не потрібно глибоко вивчати та занурюватися в код CSS, щоб отримати виконання потрібних завдань. Ще одна перевага - можливість швидко розгорнути зручний і візуально привабливий інтерфейс користувача, з можливістю його зміни в ході оновлень проекту, не роблячи роботу з нуля.

### ТОП кращих CSS-фреймворків

Існують різні види фреймворків:

* Повнофункціональні (Bootstrap, Foundation, Semantic UI та ін.).
* Орієнтовані на матеріальний дизайн: (Materialize and Material Design Lite).
* Легкі (Pure).

При виборі потрібно переконатися відповідно до технічних параметрів проекту та заздалегідь визначити функції, які вміє виконувати конкретний фреймворк. Наприклад, полегшена структура може бути актуальною для дизайну мобільних додатків, але не для великих програм зі складними завданнями. Ще треба врахувати, що згодом запити готового проекту змінюватимуться і треба думати у довгостроковій перспективі.

Вибраний фреймворк має відрізнятися зручністю, інтуїтивно зрозумілим керуванням та нескладним навчанням. Якщо з ним зручно працювати, то в процесі не буде великих проблем, що уповільнюють або повністю зупиняють розробку. Найкраще орієнтуватися на фреймворки, які постачаються з офіційними документами, посібниками для навчання та допомогою спільноти.

### Bootstrap (https://getbootstrap.com/)

Bootstrap – найпопулярніший безкоштовний фреймворк, що має відкритий вихідний код, шаблони на базі CSS і JavaScript для елементів інтерфейсу, можливості адаптивного дизайну.

Переваги Bootstrap:

* Велика екосистема, яка не має рівних. У ній зібрана велика бібліотека макетів, тем і частин інтерфейсу користувача, є багато панелей, кнопок, карток та ін. Все це допомагає розробникам і дизайнерам реалізовувати свої ідеї і вибирати відповідні варіанти для цього. На додаток у фреймворку найкраща у своїй сфері підтримка спільноти.
* Із застосуванням фреймворку спрощується процес написання HTML-код дизайнерами, можна додати класи CSS та покращити чуйність сайту. Дизайнерам не доведеться витрачати багато часу на налаштування сумісності браузерів.
* Підтримка SASS і LESS, хоча не всі розробники застосовують LESS, але важливі розробки покладаються на нього. SASS – бажаний елемент під час роботи з сайтами та додатками.
* Наявність багатьох навчальних посібників, інструкцій із практичним використанням, форумів. Навчитися працювати всередині фреймворку досить просто, навіть самостійно.
* Проста файлова структура з легким доступом, достатньо мати базові знання HTML, CSS і JS.
* Багато налаштувань за замовченням, але є можливість додавати власні, змінювати сітку.
* Повнофункціональна система із зразками коду на сайті розробника. Замість створення повністю нового коду вручну, можна скопіювати та вставити його.

### Ant Design (https://ant.design/)

Ant Design – це набір елементів для бібліотеки ReactJS. Сітка дозволяє розділити екран на 24 колонки, збільшуючи можливість налаштувань видимого сайту. Серед переваг можна виділити хороший зворотний зв'язок, доступність та можливість розробникам зберегти власний час під час роботи з проектом. До недоліків можна зарахувати відсутність документації.

### Tachyons (http://tachyons.io/)

Tachyons – зростаючий фреймворк CSS, мета якого полягає у розподілі правил CSS кілька невеликих, керованих частин із можливостями повторного використання. Подібна особливість усунула необхідність написання рядків CSS, а розробникам вдається створювати чудові сайти. Головна концепція – повторне використання шаблонів у рамках єдиного проекту.

### Tailwind CSS (https://tailwindcss.com/)

Tailwind CSS – інструмент, орієнтований на утиліти. Надається з класами, що підходять для розробки інтерфейсів користувача прямо в розмітці користувача. Tailwind є зручним для реалізації вбудованих стилів, що допомагає працювати дизайнерам з інтерфейсами без написання кодів CSS.

### Materialize CSS (https://materializecss.com/)

Є комбінацією адаптивного інтерфейсу та фреймворку, що є найкращим рішенням для розробників сайтів та додатків під ОС Android. Надається з готовими компонентами та класами. З ним розробники за короткий час розпочинають роботу із шаблонами. У фреймворку є багато інтерактивних елементів, анімація, інтерфейс користувача з рухами. Всі вони зроблені для тих, хто має мінімум часу на розгортання проекту, але потрібно проявити творчий підхід. Ще рішення підходить для людей, які працюють лише з екранами мобільних пристроїв.

### Semantic UI (https://semantic-ui.com/)

Semantic UI – адаптивний інтерфейсний фреймворк із зручним HTML. Користувачам доступно більше 3000 змінних тем, що дозволяють зробити адаптивний макет, а також надається більше 50 елементів інтерфейсу користувача. У фреймворку інтеграція з багатьма бібліотеками: Meteor, Ember, React тощо. Подібне наповнення допомагає розробити власний шар інтерфейсу користувача спільно з логікою програми.

### Foundation (https://get.foundation/)

Foundation вважається найпередовішою інфраструктурою у світі і є адаптивним інтерфейсним середовищем. Користувач отримує сітку та компоненти HTML, SASS та CSS, код, шаблони та багато іншого для роботи над проектом. Є функціонал із розширеннями JavaScript.

### UIKit (https://getuikit.com/)

UIKit – модульна та полегшена оболонка CSS зроблена для зовнішнього оформлення. Вибір на користь цього фреймворку є актуальним для тих, хто бажає зробити потужний і швидкий інтерфейс. Всередині зібрана велика колекція елементів CSS, JS та HTML. За допомогою всіх компонентів зовнішня складова буде розширюваною, простою у застосуванні та налаштуванні.

### PureCSS (https://purecss.io/)

Модулі CSS є невеликими, але відрізняються чуйністю, що допомагає швидко створювати красиві та адаптивні сайти. Основний функціонал включає вбудований дизайн з адаптивність та звичний CSS, що вимагає мінімум місця.

### Bulma (https://bulma.io/)

Платформа на базі Flexbox, надається разом із великою кількістю вбудованих функцій. З їхньою допомогою прискорюється процес виконання проекту, а кодування CSS в ручному режимі зводиться до мінімального.

### Skeleton (http://getskeleton.com/)

Легкий за структурою фреймворк, що надає потрібний набір елементів CSS для швидких розробок. Добре підходить тим, хто планує важливий, але скромний сайт. Відрізняється мінімальною стилізацією форм, вкладок та кнопок. Підходить варіант для новачків, які починають шлях у вивченні CSS, хочуть швидко створити прості та одночасно привабливі сайти.

### Препроцесори - SASS та LESS.

Написання CSS є рутиною, і дрібні завдання, такі як: пошук значень кольору, закриття тегів або будь-які інші операції, що повторюються, забирають багато часу. Ось де знадобиться препроцесор. CSS-препроцесор є скриптова мова і розширює можливості CSS.

Найпоширеніші препроцесори - це SASS та LESS. Вони мають деякі загальні основи:

* Синтаксичні елементи
* Зворотна сумісність із звичайними файлами CSS.

Проте з-поміж них є й відмінності.

* Sass розшифровується як Syntactically Awesome Style Sheets («синтаксично чудові таблиці стилів»). Sass працює на Ruby та обробляється на стороні сервера. Оскільки коріння його походження сягає мови Ruby, установка здійснюється через так звані gem-и (кілька бібліотек Ruby/Rails).
* LESS розшифровується як Leaner Style Sheets ("компактна таблиця стилів"). На даний момент його можна назвати швидше за JavaScript-бібліотекою, яка обробляється на стороні клієнта в браузері. Розробники набагато частіше вибирають LESS під час використання JavaScript з таблицями стилів. Ця технологія дозволяє повторно використовувати фрагменти CSS-коду у файлах LESS.

## DOM: структура веб-сторінки

Об'єктна модель документа (DOM, Document Object Model) – це програмний інтерфейс для документів HTML та XML. Він інтерпретує сторінку, щоб програми могли видозмінювати структуру, стиль та вміст документа. DOM відображає документ у вигляді вузлів та об'єктів, дозволяючи мовам програмування підключатися до сторінки.

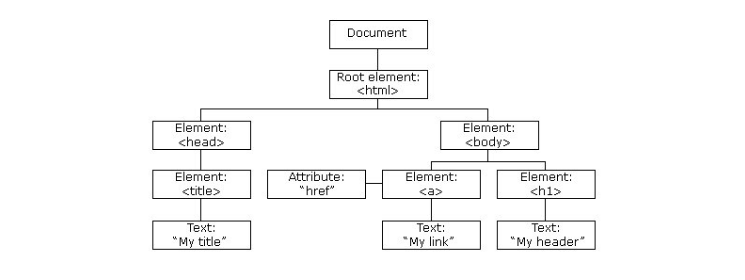


Рис.4. Дерево об'єктів HTML DOM

Як це працює. Вважатимемо, що веб-сторінка — це документ, який може бути представлений або у вікні браузера, або як вихідний HTML-код. DOM є поданням цього документа, тому його можна змінити. DOM – це об'єктно-орієнтоване представлення веб-сторінки, яке можна змінити за допомогою сценарної мови, наприклад JavaScript.

DOM повинен відповідати стандартам специфікацій W3C та WHATWG, які виконуються у більшості сучасних браузерів. Сучасний DOM будується за допомогою декількох API, що працюють разом. DOM визначає об'єкти, які повністю описують документ та об'єкти у ньому.

## JavaScript: оживляючи веб

JavaScript (JS) - одна з найпопулярніших сценарних мов. Він в основному відомий своєю застосовністю як для фронтенд-, так і для бекенд-розробки. У фронтенді він використовується для надання веб-сторінкам динаміки.

### Як працює JavaScript.

JS підвищує загальну інтерактивність сайту. Він дозволяє моделювати анімовані компоненти інтерфейсу користувача, такі як: слайдери, спливаючі вікна, розширені меню навігації по сайту і багато іншого. За допомогою JavaScript веб-сайт можна наділити різними функціональностями, що не можна досягти лише за допомогою HTML і CSS. Веб-сторінки, розроблені за допомогою JavaScript, реагують на дії користувачів та оновлюються динамічно. Завдяки JavaScript цей процес не вимагає перезавантаження сторінок, щоб відобразити зміни.

### Фреймворки та бібліотеки JavaScript

Новачкам у цій галузі знань може здатися, що фреймворки та бібліотеки роблять одне й те саме, завдяки чому різні візуальні елементи взаємодіють один з одним. Будемо чесні: це не так далеко від істини. Проте є кілька відмінних рис. Отже, давайте визначимо, що є JS-фреймворк, що таке JS-бібліотека і для якої мети вони обидва служать.

**Фреймворки** є шаблонами для створення веб-сайту або веб-програми. Вони забезпечують структуру (наприклад, основу чи заготівлі — scaffolding), де можна розмістити весь проект. Коли фреймворк встановлює шаблони сторінок, вони створюють структуру з певними виділеними областями для вбудовування коду фреймворку.

Отже, фреймворки JavaScript - це повні набори інструментів для формування та налаштування веб-сайту або додатків.

**Бібліотеки** — це набори попередньо написаних фрагментів коду, які використовуються для реалізації основних функцій JavaScript. При необхідності фрагмент коду можна легко інтегрувати до існуючого коду проекту.

Таким чином, бібліотека є спеціалізованим інструментом для конкретних вузьких потреб, а не універсальною машиною для підготовки всього проекту.

## Основні фреймворки та бібліотеки

* **Angular** – це JavaScript-фреймворк від Google, сумісний з більшістю поширених редакторів коду. Angular призначений для створення динамічних односторінкових веб-застосунків (SPA — Single Page Applications) та прогресивних веб-додатків. Ще з випуску початкової версії цей фреймворк найбільше цінувався за здатність перетворювати документи на основі HTML в динамічний контент. Angular є одним із найпопулярніших фронтенд-фреймворків. Почитати про нього докладніше можна в нашій статті про плюси та мінуси розробки на Angular.
* **Vue.js** — ще один фреймворк із відкритим вихідним кодом для односторінкових програм, що потребує знання HTML та CSS. Він використовує модель розробки на основі компонентів та дозволяє приєднувати компоненти до проекту. Vue.js — приклад бібліотеки, більше схожої на фреймворк, тому ми віднесли його до фреймворків. Він пропонує цілу купу шаблонів та патернів, які застосовуються при розробці. Vue насамперед відомий невеликим розміром документів та синтаксисом на основі HTML. Щоб ознайомитися докладніше з перевагами та недоліками Vue.js, перейдіть за посиланням.
* **Ember.js** – це фреймворк для розробки односторінкових, мобільних та десктопних програм. Він використовує шаблон проектування Model-View-ViewModel (MVVM). Інструменти Ember дозволяють проектувати середовище розробки, а його командний рядок надає інструменти для автоматизації сценаріїв.

А тепер коротко про кілька бібліотек:

* **React** — бібліотека з відкритим вихідним кодом для створення динамічних інтерфейсів, розроблена Facebook. Застосовується для створення веб-застосунків з множинними динамічними компонентами. React заснований на JavaScript та JSX і дозволяє створювати HTML-елементи для багаторазового використання. React також включає React Native, спеціальне кросплатформове середовище для розробки мобільних додатків. Щоб дізнатися більше про плюси та мінуси React та React Native, читайте статтю на цю тему.
* **jQuery**, у свою чергу, призначений для управління HTML-документами. Вона має простий API для управління подіями та розробки анімації у браузерах. Крім того, jQuery застосовується для управління об'єктною моделлю документа (DOM), а також є інструментом розробки плагінів. Вона також поставляється з легшою крос-браузерною бібліотекою, jQuery UI для мобільного фреймворку jQuery Mobile та для побудови графічного інтерфейсу.
* **D3.js** – це керована даними бібліотека для візуалізації даних. За допомогою прив'язування тимчасових даних до DOM та впровадження в документ змін, керованих даними, бібліотека дозволяє керувати даними та створювати динамічну візуалізацію даних. Вона може підтримувати та обробляти великі набори даних та динамічні відповіді для взаємодії та анімації. Функціональний стиль D3 допускає повторне використання коду та працює з CSV та HTML.

Як ми бачимо, бібліотеки JavaScript мають широку функціональність, забезпечуючи фронтенд-розробників універсальними рішеннями. Але ми не говоримо про заміну фреймворків на бібліотеки чи навпаки — завжди знайдеться місце як для одного, так і для іншого.

Якщо ви починаєте свій шлях у фронтенді, було б розумно почати з бібліотек, оскільки вони мають м'якшу криву навчання. Після вивчення основ деяких із них можна переходити до фреймворків, оскільки вони потребують глибшого розуміння JS.

### JavaScript для бекенда

Говорячи про JavaScript як комплексне середовище розробки, ми не можемо не згадати його значущість для бекенд-розробки. Оскільки JavaScript дуже популярний, світ розробки програмного забезпечення адаптував JS до специфіки бекенда.

Для розробки на стороні веб-сервера за допомогою JavaScript одним із найзначніших інструментів, ймовірно, став Node.js. Однак це не фреймворк і не бібліотека. Node.js – це середовище виконання, яке працює на основі ядра JavaScript V8. Щоб дізнатися більше про Node.js, прочитайте статтю про плюси та мінуси розробки веб-додатків на Node.js. Або подивіться відео:

## Взаємодія між фронтендом та бекендом

Фронтенд немає ізольовано від бекенда чи сервера, який фактично надає дані. Погляньмо на це з точки зору клієнт-серверної взаємодії.

### Асинхронні операції

У 2010-х роках однією з основних концепцій, що описують роботу Інтернету, був AJAX. Термін розшифровується як «асинхронний JavaScript та XML» (від англ. Asynchronous JavaScript and XML). По суті це комплекс засобів для створення веб-сайтів і веб-додатків. Розгляньмо ключові компоненти, з яких складається AJAX.

* Асинхронність - фундаментальний аспект всього інструментарію. Ключова роль підходу полягає в асинхронному оновленні веб-контенту. Це означає, що браузер не перезавантажує всю веб-сторінку, коли змін вимагає невелика частина контенту на сторінці. Наприклад, не потрібно перезавантажувати сторінку з усіма продуктами, щоб застосувати фільтр до результатів пошуку. Вам просто потрібно, щоби результати оновилися.
* JavaScript - JavaScript використовує процеси автоматизації сайтів, тому розробникам не доводиться окремо редагувати кожен процес, який відображається на сторінці. Він використовується, зокрема, для створення, додавання та керування динамічним контентом веб-сайту. Після обробки всіх процесів HTML та CSS JavaScript запускає оновлення в реальному часі, поки відвідувач переглядає сторінку з інтерактивним контентом.
* XML (або "розширювана мова розмітки", від англ Extensible Markup Language) дозволяє передавати дані, розміщені на сторінці, в браузери, які її переглядають. Це важливо для певних систем, які найчастіше не вміють взаємодіяти з даними, відформатованими іншою системою. У цьому випадку XML дозволяє уникнути несумісності при збереженні даних у текстовому форматі між тегами XML. Таким чином, XML пропонує засоби зберігання, переміщення та обміну даними, які не залежать від однієї програмної або апаратної системи. Оскільки XML застаріває, поступаючись новим форматам даних, можна запускати AJAX з JSON, який є більш коротким та зручним для читання людьми.

На сьогоднішній день AJAX не так часто обговорюється у фронтенд-спільноті через те, що асинхронний рендеринг веб-сторінки входить до стандартної комплектації всіх основних фронтенд-фреймворків, які ми обговорювали вище.

### Як працюють асинхронні поновлення.

Припустимо, користувач натискає кнопку. Клієнт у фоновому режимі надсилає запит серверу у форматі XML/JSON, доки користувач продовжує переглядати сторінку. Серверна сторона отримує дані від JavaScript, при необхідності звертається до бази даних та обробляє дані. Дані у форматі XML/JSON відправляються назад на вихідну сторінку на стороні клієнта, яка здійснила запис. Коллбек, що виконується JavaScript, збирає дані та оновлює компонент веб-сторінки, який потребує змін.

### REST та GraphQL

REST розшифровується як «передача репрезентативного стану» (англ. Representational State Transfer). По суті, це спрощений архітектурний стиль, який використовується для обміну повідомленнями між клієнтом (фронтенд) та сервером (бекенд). Служби RESTful та API – це веб-служби, що відповідають архітектурі REST.

Наприклад, розробник повинен розробити веб-додаток, який показує всіх друзів у соціальних мережах у певному порядку. Фронтенд може створювати запити до RESTful API Facebook для перегляду списку друзів та передачі цих даних назад. Будь-яка служба, що використовує RESTful API, схожа на цей загальний процес, різниця лише в тому, що дані вилучаються та повертаються. REST – це простий набір посібників та практик, які встановлюють правила взаємодії з веб-службою.

Основна ідея REST полягає в тому, що сервер байдуже, що відбувається з клієнтом. REST-повідомлення містять усю необхідну інформацію для сервера, щоб повернути необхідні дані та забути про цю операцію. Дізнатися більше про різні формати обміну повідомленнями та більш старий стандарт SOAP можна в цій статті.

GraphQL представляє сучасну версію обміну повідомленнями. Це мова запитів до баз даних із клієнтських додатків. GraphQL визначає стороні сервера, як відображати дані клієнту, і обслуговує різні типи клієнтів та його потреби у даних. Щоб дізнатися більше про цю тему, читайте статтю про можливості ядра GraphQL.

## Більше практик, які використовуються у фронтенд-розробці

Отже, ми окреслили базові складові фронтенд-розробки. Однак є ще багато термінів та понять, у яких фронтенд-розробник має розбиратися.

### Чуйний дизайн

Оскільки все Більше людей користуються інтернетом з мобільних пристроїв, а не з настільних комп'ютерів, обов'язковою характеристикою веб-додатків стала чуйність дизайну. Чуйний дизайн означає, що макет програми (включаючи функціональність та контент) підлаштовується під розмір екрану та вигляд пристрою.

Наприклад, коли сайт відвідується з настільного комп'ютера з великим монітором, дизайн включає кілька колонок, важку графіку та інтерфейс користувача, розроблений з урахуванням використання миші та клавіатури. На мобільних пристроях контент того ж сайту відображається в одну колонку, сайт адаптований для сенсорної взаємодії, але містить самі базові дані. Щоб дізнатися про чуйність більше, читайте цю статтю.

### Доступність та інклюзивність

Говорячи про доступність, ми маємо на увазі, що сайт повинен бути доступний якомога більшій кількості людей з різними видами порушень, таких як порушення зору, когнітивних функцій, слуху або рухливості. Цей термін також охоплює питання права, стандарти відповідності, різні мобільні пристрої та різні типи мережевих підключень. Докладніше на цю тему можна прочитати у статті «Як створювати інтерфейси, які принесуть користь усім: тестування доступності та принципи інклюзивного дизайну».

## Висновки

* HTML (Hypertext Markup Language) – це мова розмітки для створення веб-сайтів. Зазвичай, він застосовується для структурування веб-документа.
* CSS (Cascading Style Sheets) — це мова таблиць стилів, яка використовується для стилізації HTML-елементів на веб-сторінці. Область дії CSS включає дизайн, макет і варіанти відображення для різних пристроїв і розмірів екрану.
* DOM (Document Object Model) – це програмний інтерфейс для HTML- та XML-документів. Він містить інформацію про сторінку, щоб програми могли змінювати структуру, стиль та вміст документа.
* JavaScript – сценарна мова. У фронтенді він використовується для надання веб-сторінкам динаміки. Крім того, він надає фулстек-технології з бібліотеками та фреймворками для написання скриптів як на стороні клієнта, так і сервері.
* AJAX. AJAX (Asynchronous JavaScript and XML) — це комплекс методів для створення веб-сайтів і веб-додатків з контентом, що динамічно завантажується, без створення нового запису даних.

Звичайно, є ще багато інших технологій, навичок та загальних знань, які мають освоїти новачка. Тим не менш, ми сподіваємося, що ця стаття проклала стежку в ліс фронтенд-розробки, щоб мандрівник не збився зі шляху.

## Джерела інформації

1. Фронтенд-розробка: ключові технології і поняття  
   <https://habr.com/ru/company/otus/blog/674748/>