# 2.2. Базові веб-технології

## Сучасний Веб - це постійно зростаюча кількість сторінок і веб-додатків, що пов'язані між собою посиланнями. Він сповнений відеороликів, фотографій і інтерактивного контенту. Однак, взаємодія веб-технологій, завдяки яким все це так злагоджено працює, залишається прихованою від очей звичайного користувача.

## Веб-технології стрімко розвиваються і розробники мають широкі можливості створювати веб-вміст нового покоління. Сьогоднішній Інтернет є результатом безперервних зусиль відкритої веб-спільноти, яка розробляє новітні технології і домагається їх підтримки всіма браузерами.

## Основи Веб

Все, що бачить користувач на сайті від шрифту, фону, випадного меню, слайдеру створено за допомогою базової трійки засобів – мови HTML, таблиць стилів CSS і сценаріїв JavaScript.

HTML формує структуру сторінки (скелет), CSS визначає її зовнішній вигляд, а JS надає динаміку. Якщо пояснити наочними образами, то HTML - це тіло, CSS – одяг, JS - поведінка.

### HTML – мова розмітки тексту

Мова гіпертекстової розмітки HTML (HyperText Markup Language) є основним будівельним засобом для веб-сторінок, використовується для створення та візуального представлення веб-сторінок. Визначає структуру і описує зміст веб-сторінки в структурованій формі.

HTML-сторінка є звичайним текстовим документом, в якому використано спеціальні оператори – **теги** (*tag*) чи інша назва **дескриптори** (*descriptor*) для позначення, в якому вигляді буде виведено текстовий чи інший елемент у вікні браузера. Прикладами таких операторів є <img>, <title>, <p>, <div>, <table> тощо.

HTML дозволяє формувати на сторінці сайту текстові блоки, додавати до них зображення, організовувати таблиці, додавати до дизайну сайту звуковий супровід, організовувати гіперпосилання з переходом до інших розділів сайту або ресурсів Інтернету і компонувати всі ці елементи між собою. За допомогою HTML можна створити як статичний так і динамічний сайт. Сторінки, які створено лише засобами HTML мають розширення .html.

**Гіперпосилання** (Hyperlink) — це базовий функціональний елемент HTML-документу, який реалізовує зв'язок певного об'єкту веб-сторінки з іншим об’єктом. Для гіперпосилання може використовуватися як фрагмент тексту, так і графічний об'єкт, а сам зв'язок можна встановлювати як між об’єктами одного сайту, так і між об'єктами, що розміщені на різних сайтах Інтернету.

HTML є мовою, що лише інтерпретується, тому, для виконання коду, його не потрібно компілювати. Інтерпретатор мови втілено в браузер і він «компілює» код безпосередньо під час відкривання документа. Якщо в коді сторінки виявлено помилку, інтерпретатор, зазвичай, не видає відповідного попередження, а просто ігнорує помилку, що може зіпсувати зовнішній вигляд завантаженої сторінки. Для запобігання цього розробникам слід бути уважними під час складання HTML-коду і ретельно тестувати результати своєї роботи.

### CSS – каскадна таблиця стилів

CSS (Cascading Style Sheets) — це технологія опису зовнішнього вигляду документа, що створено засобами HTML.

CSS використовується для привласнення певних особливостей для елементів HTML-сторінки: колір, шрифт, розташування на сторінці тощо. До появи CSS оформлення елементів вказувалося безпосередньо в HTML-коді сторінки. Проте, з появою CSS стало можливим принципове розділення структури і опису документа. За рахунок такого розподілу стало можливим легке застосування єдиного стилю оформлення для кількох сторінок сайту, а також швидка зміна цього оформлення.

#### Переваги CSS:

* Застосування кількох варіантів дизайну сторінки для різних пристроїв перегляду, наприклад, для комп’ютера, планшету чи телефону.
* Зменшення часу завантаження сторінок сайту за рахунок перенесення правил опису даних до окремого CSS-файлу. В цьому випадку браузер завантажує лише структуру документа і дані, що містяться на сторінці. CSS-файл з правилами опису цих даних завантажується браузером лише один раз і зберігається в кеші браузера.
* Простота подальшої зміни дизайну. Не потрібно виправляти кожну сторінку, достатньо лише змінити кілька правил у CSS-файлі.
* Додаткові можливості оформлення. Наприклад, за допомогою CSS-правил можна застосувати обтікання певного блоку текстом або зробити так, щоб меню фіксовано знаходилося в певному місці при перегортанні сторінки.

### JavaScript – мова сценаріїв

JavaScript – це фрагменти програмного коду (скрипти), що надають динаміки для певних елементів сторінки. Програмний код на JavaScript вписується безпосередньо в текст HTML-сторінки і інтерпретується браузером в міру завантаження цього документа. За допомогою JavaScript можна динамічно реагувати на події, які пов'язані з діями відвідувача або змінами стану сторінки чи вікна.

Важливою особливістю JavaScript є об'єктна орієнтованість. Програмісту є доступними численні об'єкти, такі, як документи, гіперпосилання, форми, фрейми тощо. Об'єкти характеризуються описовою інформацією (властивостями) і можливими діями (методами).

## Серверні мови програмування

Серверне програмування використовують для обробки дій користувача на динамічних складних проектах, таких як пошукові системи, електронна пошта, форуми, Інтернет-магазини тощо.

В цих випадках браузер приймає від відвідувача дані і надсилає їх до веб-серверу, який:

* Приймає запит на завантаження файлів (веб-сторінки, таблиці стилів, графічні зображення, фільми, звуки, архіви, виконувані файли тощо).
* Здійснює пошук цих файлів на дисках серверного комп'ютера.
* Скеровує завдання до відповідних програм, що виконують додаткові дії над файлами.
* Формує результати обробки серверних програм у HTML-код.
* Надсилає до браузера сформовану веб-сторінку в HTML-коді.

Для виконання завдань веб-сервера використовуються спеціальні програми, що працюють разом з веб-сервером на тому ж серверному комп'ютері. Вони називаються серверними програмами, не мають інтерфейсу користувача і "спілкуються" лише з веб-сервером, приймають від нього введені користувачем дані і повертають йому результат. Цим вони докорінно відрізняються від клієнтських програм, що працюють безпосередньо з користувачем. Є й багатофункціональні програми, що виконують функції не лише від веб-сервера, але і FTP чи поштового сервера.

Сторінки, що формуються серверними програмами називаються динамічними, на відміну від статичних сторінок у форматі .html.

#### Серверні програми поділяються:

1. **Розширення веб-сервера (додатки формату ISAPI, модулі розширення Apache тощо).** Це вбудовані в веб-сервер серверні програми. Вперше запропоновано фірмою Microsoft для їх сервера Microsoft Internet Information Server (інтерфейс ISAPI) і розробниками популярного веб-сервера Apache.
2. **Активні серверні сторінки (ASP, РНР тощо).** Це звичайні статичні веб-сторінки, що збережені як файли, які, окрім звичайного HTML-коду, містять команди, що обробляються або самим веб-сервером або його розширеннями.
3. **Серверні сценарії, написані мовою, що інтерпретується (PНР, Python, Java, Ruby).** Сценарії оброблення даних при різних діях користувача на сайті.

**Розширення веб-сервера** - різновид серверних програм, що представляють бібліотеки, в яких реалізовано логіку серверної програми. Такі бібліотеки вбудовуються в програму веб-сервера і працюють як її невід'ємна частина.

**Активні серверні сторінки** - це звичайні веб-сторінки, що містять особливі серверні сценарії (програмний код в HTML-коді) і виконуються самим веб-сервером або його розширеннями. Перевагами активних серверних сторінок є легкість та швидкість написання, і простота відлагодження.

**Серверні сценарії** подібні до активних серверних сторінок, проте, представляють собою "чистий" програмний код, без HTML-фрагментів. Інтерпретатор коду представлено як розширення веб-сервера. Сценарії, зазвичай, пишуться мовами програмування РНР, Java, Perl, Python, Ruby. Фактично писати сценарії можна будь-якою мовою програмування, для якої є інтерпретатор.

Серверні сценарії споживають значно більше системних ресурсів, ніж активні сторінки, бо для обробки кожного набору даних користувача запускається власна копія інтерпретатора, а інтерпретатор, в свою чергу, витрачає багато ресурсів на обробку сценарію. І все ж, незважаючи на це, сценарії - найпопулярніший спосіб створення серверних програм.

## Ajax - технологія взаємодії з сервером без перевантаження сторінок

 В стандартному веб-застосуванні обробкою всієї інформації займається сервер, браузер відповідає лише за взаємодію з користувачем, передачу запитів і виведення отриманих даних у форматі HTML.

В Ajax-застосуванні між користувачем і сервером з'являється ще один посередник – програмний механізм (рушій) Ajax (Asynchronous Javascript And XML). Він визначає, які запити можна обробити з боку клієнта, а які необхідно виконувати на сервері. Змінюється поведінка сервера: якщо раніше на кожен запит сервер видавав нову сторінку, то тепер він надсилає лише ті дані, які потрібні клієнту, а рушій Ajax в браузері формує з них HTML-код.

Асинхронність виявляється в тому, що не кожен запит користувача скеровується до сервера, а також не кожна реакція сервера обумовлена запитом користувача. Значну частину запитів формує рушій Ajax, причому є можливим, щоб він передбачав запити користувача.

Змінюється якісне навантаження на сервер - якщо раніше запитів було мало, але кожен з них вимагав значних ресурсів (серверу потрібно витягнути інформацію з бази даних, сформувати з неї веб-сторінку і відправити до браузера), то тепер завдання сервера спрощується (формувати веб-сторінки не потрібно, та і об'єм даних, що передаються є значно меншим), хоча запитів доводиться обробляти більше.

Створювати додаток на Ajax є трудомістким і складним завданням. Потрібно написати і відлагодити на JavaScript рушій з десятків тисяч рядків коду, а також реалізувати серверну частину.

## Розвиток та версії HTML



Рис.1. Розвиток та версії HTML

1991р. – представлення мови гіпертекстової розмітки (HTML, HyperText Markup Language), призначеної для розмітки и оформлення документів World Wide Web. Автор Тімоті Джон Бернерс-Лі.

1993р. – версія HTML 1.2.

1994р. – заснування консорціуму W3C (.World Wide Web Consortium), що затверджує стандарти HTML.

1995р. – версія HTML2.0

1996р. – розроблення ієрархічних стильових специфікацій CSS (Cascading Style Sheets), що призначені для візуального оформлення HTML-документів.

1997р. – версія HTML3.2

1997р. – версія HTML4.0

1999р. – версія HTML4.1

2014р. – версія HTML5.0

## Версія HTML5

Інтернет зараз сильно відрізняється від того, що був у 1999 році, коли вийшло останнє велике оновлення HTML 4. Смартфони, планшети та інші мобільні пристрої - це нові виклики для інженерів і розробників програм. Глобалізація зробила стандартизацію Інтернет-технологій пріоритетним завданням, оскільки світове споживання Інтернету невпинно зростає, а популярні технології розвиваються швидше.



Рис.2. HTML5 та суміжні технології

HTML5 покликаний надати зручний і послідовний перехід до нових стандартів. HTML5 сильно спростив процес створення веб-додатків та прибрав багато проблем щодо кросбраузерності та мультиплатформості. Завдяки HTML5 веб-сторінки навчилися зберігати дані локально в браузерах користувачів, що дозволяє отримати контент швидше і безпечніше. Раніше, браузери використовували різні плагіни для програвання мультимедіа файлів, на тепер вбудована підтримка аудіо і відео усуває проблеми з сумісністю.

#### Переваги HTML5:

* Здебільшого сумісний з існуючими стандартами для HTML4. Нові засоби розмітки працюють згідно існуючих, нові API ґрунтуються на існуючих JavaScript / DOM, які розробники використовуються протягом багатьох років.
* Додає нові потужні засоби в HTML, які були раніше доступні в Веб тільки за допомогою технології плагинів, таких як Flash, або за допомогою складного коду JavaScript чи спеціальних прийомів.
* Краще підходить для написання динамічних додатків, ніж попередні версії HTML.
* Має чіткий алгоритм синтаксичного аналізу, тому всі браузери, які реалізують HTML5, створюють однакове дерево DOM з однією розміткою, що є величезним виграшем для сумісності.

### Властивості HTML5

Мова HTML5 містить багато нових властивостей, що робить HTML значно більш потужним і зручним для створення веб-додатків.

### Зміни щодо елементів мови

* **Застарілі елементи, які не вживаються**: isindex, noframes, acronym, applet, basefont, dir, font, frame, frameset, big, center, strike, tt.
* **Нововведені елементи**: summary, time, aside, audio, command, data, datalist, details, embed, wbr, figcaption, figure, footer, header, article, hgroup, bdi, canvas, keygen, mark, meter, nav, output, progress, rp, rt, ruby, section, source, track, video.
* **Семантичні елементи**: <nav>, <header>, <footer>, <aside>, <article>.
* **Нове призначення старих елементів:** <b> vs <strong>, <i> vs <em>, <small>, <s>, <del>
* **Нові види елементів управління:** dates and times, email, number, range, tel, url, search и т.д.
* **Заміна елементів:**
	+ <applet>, <object> замінено тегом <embed> (використовується для втілення об’єктів);
	+ <acronym> замінено тегом <abbr> (використовується для аббревіатур);
	+ <bgsound> замінено тегом <audio> (для втілення програвача на сторінку);
	+ <frame> замінено тегом <iframe>;
	+ <listing>, <xmp> замінено тегами <pre> и <code> (для уставляння лістингів програм та кодів);
	+ <plaintext> — замінено тегом <pre>

### Властивості форм

HTML5 надає стандартизований, простий спосіб реалізації таких властивостей, як вибір дати, повзунки і клієнтська перевірка – засіб перевірки заповнення полів форми з боку користувача.

Якщо користувач в полі, що призначене для введення електронної адреси зазначить номер телефону, то йому висвітиться попередження про помилку. Аналогічно, в полі для номера телефону можна водити лише цифри, інакше висвітиться помилка. Для такої клієнтської перевірки HTML5 використовує нові атрибути min, max, step, pattern, і required для введених даних.

Логічний атрибут required вказує браузеру, що форму можна відправити лише при заповненні даного поля, воно не може бути порожнім, а також містити допустимі типи значень.

Якщо обов'язкове поле є пустим, форма не відправиться. Браузер видасть користувачеві повідомлення з помилкою. Наприклад, «будь ласка, заповніть це поле» або «необхідно вказати значення».

### Власна підтримка відео та аудіо

Протягом багатьох років відео і аудіо в Веб втілювалося за допомогою Flash-технологій. Технологія Flash стала популярною на початку 21 століття, оскільки відкриті стандарти не надавали для різних браузерів сумісний механізм реалізації таких речей. Як наслідок, різні браузери реалізували різні способи виконання одних функцій (наприклад, < object> і <embed>), роблячи тим самим весь процес дійсно складним. Flash надавав високоякісний, легкий спосіб відтворення відео в різних браузерах.

HTML5 містить елементи <video> і <audio> для простої реалізації власних відео і аудіо плеєрів за допомогою відкритих стандартів, і також містить API, що дозволяє легко реалізувати індивідуальні елементи управління плеєром.

### API малювання на полотні

Елемент <canvas> і відповідний API дозволяють визначити на сторінці область для малювання, і використовувати команди JavaScript для малювання ліній, фігур і тексту, імпорту та маніпуляцій з зображеннями і відео, експорту в різні формати зображень, і багатьох інших речей. Canvas задає прямокутну область, що контролюється розробником і призначена для використання сценаріїв візуалізації двовимірних 2D-форм і растрових зображень.

Елемент <canvas> ідеально підходить для створення візуальних матеріалів, які покращують користувацькі інтерфейси, діаграми, фотоальбоми, схеми, графіки, анімовані матеріали і вбудовані графічні додатки.

### Веб-сховище Web Storage

До HTML5 єдиним способом локального зберігання даних було використання механізму файлів cookies. Файли cookies призначені для зберігання невеликих обсягів даних, але модель JavaScript для роботи з ними незручна.

В HTML5 запроваджено кращу альтернативу для файлів cookies, яка дозволяє легко і просто зберігати інформацію на комп'ютері відвідувача. Ця інформація може зберігатися на клієнтському комп'ютері необмежений час, не надсилається до веб-серверу (якщо тільки розробник не зробить це сам), може бути великого обсягу і для роботи з нею потрібно лише пара простих об'єктів JavaScript.

Ця можливість називається веб-сховищем (Web Storage) і добре підходить для застосування автономного режиму роботи сайтів, оскільки дозволяє створювати автономні програми, які можуть зберігати всю необхідну їм інформацію, навіть за відсутності під’єднання до Інтернету.

Функціональність веб-сховища HTML5 дозволяє веб-сторінці зберігати дані на комп'ютері відвідувача. Ця інформація може бути короткочасною, яка видаляється після закривання браузера, або довготривалою, яка залишається доступною при наступному відвідуванні веб-сторінки.

### Автономні веб-додатки

HTML5 надає можливості для веб-додатків виконуватися в автономному режимі. Кеші додатків дозволяють зберігати копії всіх ресурсів, необхідних для локального виконання веб-додатків, а бази даних Web SQL - локальну копію даних веб-програми. Спільно вони спроможні використовувати додаток, за відсутності з'єднання з Інтернетом. Згодом, коли мережа стане доступною зміни синхронізуються з основною версією на сервері.

### Web workers

Загальною проблемою веб-додатків є зменшення продуктивності, якщо потрібно обробити багато даних. В HTML4 всі JavaScript скрипти на сторінці могли виконуватися тільки по черзі. В деяких випадках користувачі повинні були чекати завершення виконання скрипта, щоб продовжити взаємодію з веб-сторінками.

Завдяки технології Web Worker в HTML5 стало можливим створення та запуск скриптів, які будуть виконуватися у фоновому режимі і не впливати на швидкість завантаження основної сторінки. Потік Worker може виконувати завдання без втручання в користувацький інтерфейс.

### Веб-сокети WebSocket

Протокол http, за яким передаються дані, є синхронним, тобто побудованим за моделлю «запит - відповідь». WebSocket - це серйозне розширення HTTP5, яке дозволяє додаткам підтримувати багатокористувацьку взаємодію в режимі реального часу. Завдяки цьому клієнт і сервер стають рівноправними учасниками обміну даними. Кожен при цьому працює сам по собі, і за потребою надсилає дані до іншого. Інша сторона відповідає в міру необхідності, і тільки тоді, коли ця необхідність виникне.

WebSocket - протокол повнодуплексного двохбічного зв'язку поверх TCP-з'єднання, що призначений для обміну повідомленнями між браузером і веб-сервером в режимі реального часу. WebSockets встановлює одне TCP-з'єднання і підтверджує, що сервер може спілкуватися з WebSocket, роблячи спеціальні перевірки. Після цього сервер і клієнт можуть надсилати текстові повідомлення через встановлене з'єднання за необхідності, в результаті чого зв'язок стає швидше. З'єднання постійно тримається відкритим, поки порт не буде закрито. При цьому в веб-сокетах немає обмежень на кількість з'єднань.

Така спроможність істотно покращує ефективність веб-додатків, оскільки дані можуть безперервно передаватися між клієнтом та сервером без опитування сервера чи немає доступних оновлень, зайвих HTTP заголовків та постійного перезавантаження сторінки.

### Геолокація

Специфікація геолокаціі (<https://w3c.github.io/geolocation-api/>) визначає API, який дозволяє веб-додатку легко отримувати доступ до даних у будь-якому місці розташування, яке стало доступним, наприклад, за допомогою засобів GPS пристроїв. Це дозволяє надавати додатку різних корисних властивостей, що пов'язані з місцем розташування, наприклад, розмістити контент, який більше підходить для місця розташування.

#### Максимальна підтримка мобільних пристроїв

Швидке поширення мобільних пристроїв змусило скорегувати HTML стандарти. HTML5 спрощує мобільну підтримку та враховує особливості смартфонів і планшетів.

## Чи буде версія HTML6?

З 2014 року очікується випуск стандарту HTML6 з новими функціями і можливостями для кращої сумісності і зручності використання.

**Виділені бібліотеки HTML6.** Великі бібліотеки створюють надмірне навантаження, і для цього в HTML6 створюється кешувальна версія бібліотек для простоти використання і відомостей про інші бібліотек.

**Свобода зміни розміру зображення.** Для кращого перегляду в стандарті HTML6 може бути введений новий тег <srcset>, що дозволить змінювати розмір і налаштовувати зображення відповідно до розміру вікна і дозволом пристрої.

**Експрес-теги.** <logo> </logo> для логотипу на сторінці, <sidebar> </sidebar»,« navigation> </navigation> - деякі спеціальні теги.

**Анотації до зображень і відео.** У HTML5 можна анотувати слова, але не зображення і відео, але в стандарті HTML6 можуть бути включені анотації для відео і зображень.

**Покращення аутентифікації.** Хоча HTML5 був хороший з точки зору безпеки і аутентифікації, для більшої зручності ключі безпеки можна зберігати за межами сайту, щоб запобігти доступу людей. Тому він також може бути прийнятий WHATWG.

**Краща контактна інформація.** Іноді, щоб витягнути контактний номер, електронну пошту, потрібно його скопіювати і вставити. У стандарт HTML6 може бути включена поліпшена система контактної інформації.

**Підключення мов.** У HTML6 буде реалізовано підключення ефективних мов, щоб розробники могли створювати унікальний дизайн за час, а браузер міг легко реалізувати цю функцію за кілька кроків.

**Підключення препроцесорів.** У версії HTML5 препроцесори присутні, але в HTML5 цей процес буде оптимізований і відкриє нове вікно для розробників.

**Інтеграція камери.** Зазвичай, камери і мікрофони використовуються для спілкування з людьми. HTML6 допоможе отримати доступ до медіа, розташованих на пристрої користувача, і мати кращий контроль над мікрофоном і камерою.

**Сильніші мікроформати.** У HTML є деякі теги, такі як нижній колонтитул, абзац, заголовок, але варто додати стандартні розділи веб-сторінки, такі як адреса або номер телефону. Це підвищить якість веб-сайту та швидкість обробки інформації пошуковими системами.

## Розвиток та рівні CSS

Каскадні таблиці стилів не мають версій в традиційному розумінні; натомість є рівні. Кожен рівень CSS заснований на попередньому, вдосконалюючи визначення і додаючи функції. Набір функцій кожного вищого рівня є надбудовою будь-якого нижчого рівня, а поведінка, що дозволена для даної функції на більш високому рівні, є підмножиною того, що дозволено на більш низьких рівнях. Таким чином, властивість, що відповідає більш високому рівню CSS, також відповідає всім нижчим рівням.

Існують три рівні CSS, які визначаються наявністю завершеною редакції структури. Перехід від одного рівня до іншого, в основному, супроводжувався деякими видозмінами в структурі і в правилах стильового оформлення, технологічними доповненнями, а також спробами систематизувати застосування CSS.

Базова ілюстрація для розуміння логіки (рис. 3):



Рис.3. Рівні CSS та модулі

**CSS 1.** Перший рівень структури стильових шаблонів, остаточно затверджений 11 січня 1999 року.

**CSS 2.** Другий рівень стильових конструкцій, початок обговорення якого датується травнем 1998 року.

**CSS 3.** Третій рівень стильового оформлення електронних документів, прийнятий до обговорення 23 травня 2001 року. CSS 3 позиціонується розробниками як єдиної системи подання стилів в електронному документі, заснованої на використанні спеціальних модулів. З цього рівня модулі вирівнюються незалежно один від одного.

Робоча група вирішила більше не підтримувати тільки одну Специфікацію для всієї мови. Замість цього є специфікація для кожного модуля, яка розглядає CSS2.1 (або CSS2.2) в якості основного. Ось чому можна знайти: CSS Flexible Box Layout Module Level 1, CSS Grid Layout Module Level 1, Selectors Level 3, Selectors Level 4 тощо.

Ілюстрація для розуміння логіки CSS Level 3 (рис.4).



Рис.4. Рівні специфікацій CSS

Підводячи підсумок, можна сказати, що CSS3 (або CSS Level 3) - це набір різних специфікацій (кожна з яких пов'язана з модулем) плюс «Специфікація CSS2.2» в якості основної.

Не варто думати, що CSS3 - це самостійна мова, це тільки один рівень CSS, який визначає, як повинні бути організовані специфікації.

**CSS 4.** Немає CSS рівня 4. Незалежні модулі можуть досягати рівня 4 або вище, але CSS мова більше не має рівнів. (Термін «рівень CSS 3» використовується тільки для того, щоб відрізнити його від попередніх монолітних версій.) CSS - це просто CSS, і він містить різні модулі, які підвищуються незалежно один від одного.

## Сучасні засоби веб-технологій

### Бібліотеки

Бібліотека - це структурований набір корисного функціоналу. Стандартна бібліотека повинна містити функції для роботи з рядками, даними, DOM-елементами, подіями, cookie, анімацією, запитами, і багатьох інших. Кожна функція повертає значення до додатку,що зробив запит, яке може в подальшому використовувати їх в залежності від логіки розробника. Це можна порівняти з підбором запчастин для автомобіля: ви вільні у виборі будь-якого варіанту, з яким машина буде працювати, але логіка цілком на вашій совісті.

Бібліотека, зазвичай, скорочує час розробки приблизно на 20%, дозволяючи не турбуватися про дрібниці, втім, слід врахувати певні особливості:

* Баг в реалізації бібліотеки може викликати складності в його знаходженні та способи усунення.
* Немає гарантії, що команда розробників оперативно випустить оновлення або виправлення недоліків.
* Оновлення може змінити API, що може спричинити значні зміни у коді.

Більшість популярних бібліотек поширюються як Open Source. Популярних комерційних бібліотек досить мало, крім складних вузькоспеціалізованих напрямів.

### Фреймворки

Фреймворк - це каркас додатку. Він зобов'язує розробника вибудовувати архітектуру програми відповідно до деякої логіки. Фреймворк, зазвичай, надає функціонал щодо подій, сховищ і зв'язування даних. За аналогією з машиною, фреймворк надає готові рами, корпус і двигун. Можете додавати, видаляти або замінювати деякі деталі, при цьому припускаючи, що автомобіль залишиться в робочому стані.

Фреймворк знаходиться на більш високому рівні абстракції у порівнянні з бібліотекою і дозволить без зайвих зусиль розробляти близько 80% застосування. Але слід врахувати:

* Останні 20% можуть викликати чималі труднощі через обмеження, що накладаються фреймворком.
* Оновлення фреймворку може бути пов'язане зі складнощами підтримки попередніх версій.
* Основний код і концепції рідко задовольняють розробників в своєму первісному вигляді. Вони завжди знайдуть «кращий» спосіб зробити що-небудь.

### Інструменти

Інструмент - це допоміжний засіб розробки, але він не є невід'ємною частиною проекту. Інструменти містять системи збирання, компілятори, транспайлери, механізми розгортання та інше.

Інструменти спрощують процес розробки. Наприклад, використання препроцесору Sass замість чистого CSS, оскільки він надає можливість використовувати цикли, функції, локальні змінні і багато іншого. Браузери не розуміють Sass/SCSS синтаксис, тому код переводиться в CSS.

### Open Source

Open Source – це програмне забезпечення, яке постачається для кінцевого користувача з відкритим вихідним кодом. Тобто програму можна доопрацювати під власні завдання без порушення авторських прав розробників вихідного ПЗ. Рішення розповсюджується під ліцензіями GNU/Linux, MIT та інші.

Саме поняття виникло як альтернатива пропрієтарного ПЗ, коли комерційні компанії закривали доступ до вихідного коду.

Програмне забезпечення, яке розповсюджується як Open Source продукт, має ряд особливостей:

* Програмні продукти постачаються абсолютно безкоштовно до кінцевого споживача.
* Розробляються різними фахівцями, які його використовують. Вони уважно відстежують код програми, оперативно усуваючи помилки чи вразливості, які можуть спричинити критичні наслідки.
* Більшість продуктів Open Source сумісні з різними сімействами операційних систем, тобто вони кросплатформні.
* Спільнота розробників вільного програмного забезпечення відкрита для зворотного зв'язку від користувачів. Кожен з них може зробити пропозицію щодо покращення або додавання нової функції.
* Оновлення Open Source рішень відбувається значно частіше, ніж комерційні продукти. Кінцевий користувач отримує актуальні виправлення миттєво, як тільки вони фіксуються.
* Активний розвиток вільного ПЗ пожвавлює конкуренцію серед комерційних організацій, що сприяє підвищенню якості програм.
* Як комерційну основу розробники використовують у своїх рішеннях опцію добровільних пожертв. Тобто, користувачі можуть за власним бажанням підтримати програмістів фінансовими коштами шляхом переведення на електронний гаманець.
* Більшість Open Source рішень створюються під операційні системи сімейства Unix/Linux, що зменшує ризик зараження комп'ютера чи сервера вірусом. Шкідливе ПЗ переважно пишеться під Windows.

Для розміщення відкритих проектів з контролем версій популярною є онлайн платформа Github з використанням Git. Git — розподілена система контроля версій, яка надає можливість розробникам відслідковувати зміни в файлах и колективно працювати над одним проектом. Хоча Github більше відомий як платформа для розробки open source проектів, ресурс також надає можливість використання приватних репозиторіїв.

### CSS-фреймворк Bootstrap

Bootstrap є найпопулярнішим фреймворком, що вільно поширюється через офіційний сайт getbootstrap.com. Фреймворк містить готові стилі і скрипти, для застосування яких достатньо прописати необхідні класи і атрибути html-елементів. За допомогою фреймворку легко розробляти мобільні проекти, завдяки сітці Bootstrap, яка дозволяє легко адаптувати будь-який сайт і добре відображати його на будь-яких пристроях.

Файли Bootstrap з готовим написаним кодом під’єднуються до сайту в елементі head, після чого стають доступними його можливості.

Bootstrap ідеально підходить при роботі в команді. Верстка на Bootstrap при належному розумінні відбувається в 3-5 разів швидше, а однаковість коду дозволить іншому розробнику вносити правки. У разі верстки без фреймворку, то тут в кожного розробника може бути свій стиль і іншій людині доведеться витратити час на вивчення його коду.

Як переваги фреймворку є присутність великої спільноти і наявність хорошої документації. Завдяки такій поширеності Bootstrap з'явилося багато шаблонів, де вже перероблено дизайн всіх основних елементів. На основі таких шаблонів можна робити сайти, лише незначно щось змінюючи.

#### Недоліки Bootstrap

1. Коду в бібліотеці написано на всі випадки життя, це більше, ніж потрібно при розробці з нуля.
2. Шаблонний дизайн. Ця проблема легко вирішується, оскільки вона є буде в тому випадку, якщо використовувати тільки готові компоненти фреймворку і нічого не стилізувати.

#### Компоненти фреймворка

Bootstrap є всеосяжним фреймворком, в нього закладено багато компонентів, все, що може знадобитися при розробці типових сайтів, наприклад, випадне меню, кнопки, таби, індикатори стану, хлібні крихти, списки, заголовки і т.д.

В фреймворку присутній іконковий шрифт, що містить понад 200 іконок, вставляти їх на веб-сторінки просто, опис є в офіційній документації.

### JS-фреймворки

JS-фреймворки - це інструменти для побудови динамічних додатків на Javascript. Розробники використовують JS-фреймворки там, де неможливо/складно/довго виконувати завдання звичайними засобами. У переважній більшості випадків, фреймворки використовуються для написання, так званих, Single Page Applications.

Односторінковий додаток (Single Page Application, SPA) - це веб-додаток чи веб-сайт, який вміщується на одній сторінці. В односторінковому додатку весь необхідний код - HTML, JavaScript, та CSS - завантажується разом із сторінкою або динамічно довантажується за потребою, зазвичай, у відповідь на дії користувача. Сторінка не оновлюється і не перескеровує користувача на іншу сторінку в процесі роботи з нею. Взаємодія з односторінковим додатком часто здійснюється через динамічний зв'язок з веб-сервером.

За допомогою JS-фреймворків можна розробляти як повноцінні сайти, так і функціональні модулі. Фреймворки надають чітку структуру програми та реалізуються з використанням так званих «патернів проектування». Найбільш поширені наступні патерни: MVC (Model-View-Controller), MVP (Model-View-Presenter) і MVVM (Model-View-ViewModel).

#### Переваги побудови програми на JS-фреймворку:

* Ефективність - проекти, які раніше розроблялися місяцями і мали сотні рядків коду, зараз можуть бути реалізовані значно швидше з добре структурованими готовими шаблонами і функціями. Коду стає помітно менше і він чистіше, що позитивно відбивається на швидкості розробки, а також підтримки та усунення помилок в коді програми.
* Безпека - кращі JavaScript фреймворки мають фірмову систему безпеки і підтримуються великою спільнотою.
* Витрати - більшість фреймворків є з відкритим кодом і безкоштовні. Оскільки вони допомагають програмістам швидше розробляти власні рішення, підсумкова ціна веб додатку буде нижчою. Наявність структури передбачає модульність додатку, а це дає можливість простіше працювати над додатком кільком розробникам одночасно.
* Можливість швидко створити мобільний або настільний кросплатформний додаток з веб-версії.

Додатків на js-фреймворках створено багато і цей сегмент швидко розвивається.

### Популярні JS-фреймворки

### Angular.js

Angular.js. Зараз це найбільш популярний JS фреймворк для розробки односторінкових додатків (SPA Single-Page-Applications) і він може похвалитися найбільшою спільнотою розробників. Серед головних переваг для створення стартапів і середніх компаній називають: швидке написання коду, швидке тестування будь-якій частині програми та двохстороння прив'язка даних (зміни в бекенді відразу ж відбиваються в користувацькому інтерфейсі).

Angular2 поставляється з великим списком функцій, які дозволять розробити всі, починаючи від веб- до декстопних і мобільних додатків. Фреймворк побудовано на TypeScript від Microsoft. Функції Angular2 мають архітектуру на основі компонентів, покращений DI (dependency injection - впровадження залежностей), ефективний сервіс логів, межкомпонентну взаємодію і багато іншого.

Обидва Angular є найкращим варіантом для корпоративних додатків або для середовищ програмування з високими стандартами до читання коду.

### React.js

Це більше бібліотека, ніж фреймворк і доводить свою ефективність всередині динамічних додатків з високим трафіком (пропускною спроможністю), як приклад Facebook та Instagram. Це самий швидкозростаючий JS-фреймворк, на сьогодні налічується близько 1000 авторів Github.

React є складним у вивченні, проте, він робить розробку додатків простою і легкою для розуміння. Він може ідеально підійти для складних, значних програмних рішень з високим ступенем навантаження.

### Vue.js

Vue 2.0 представлено в 2016 році, він увібрав найкраще з Ember, React і Angular, поклавши це все в зручну упаковку. Автори стверджують, що Vue2 є одним з найшвидших фреймворків.

Vue.js буде кращим вибором для швидкої розробки крос-платформних додатків. Він може стати міцною основою для високопродуктивних додатків в одну сторінку (SPAs) і вигідним рішенням для тих випадків, коли продуктивність є важливішою, ніж хороша організація коду або структура програми.

**Node.js**

Node.js – це серверна платформа з відкритим вихідним кодом, що створена на основі Google Chrome JavaScript Engine. Це одне з найбільш завантажених кросплатформових середовищ для виконання коду JavaScript. Node.js - це асинхронна, однопоточна, неблокуюча модель введення/виводу, яка робить її легкою та ефективною.

## JS-бібліотеки

### Бібліотека jQuery

JQuery - це кросплатформова JavaScript-бібліотека, яку встановлюють на 65% з 10 мільйонів найбільш відвідуваних сайтів в Інтернеті. JQuery - це безкоштовне програмне забезпечення з відкритим вихідним кодом. Бібліотеку було представлено в 2006 році Джоном Ресігом, яка спростила створення клієнтських рішень. У той час, це було особливо корисно через невідповідностей, які існували серед реалізацій JavaScript в різних браузерах.

JQuery - швидка, невелика і багатофункціональна бібліотека JavaScript, з простим у використанні API, який працює у всіх сучасних браузерах. Завдяки поєднанню універсальності і розширюваності, jQuery змінив спосіб, яким мільйони людей пишуть JavaScript.

З моменту свого створення jQuery перетворився в щось більше, ніж просто бібліотека JavaScript, яка пропонує можливість виконувати як прості, так і потужні операції в кросплатформовому режимі. Як доповнення до основної бібліотеки jQuery існують інші великі проекти, про які варто згадати.

#### jQuery UI https://jqueryui.com/

JQuery UI - це набір ефектів, віджетів і тем для користувацького інтерфейсу, вбудованих в бібліотеку jQuery. Якщо створюються інтерактивні веб-додатки jQuery UI - це ідеальний вибір. Бібліотеку презентовано в 2007 році, через рік після самої jQuery. Працює як додаткова бібліотека для jQuery, оскільки він підвищує кросплатформову сумісність бібліотеки, щоб допомогти створювати віджети, які можна використовувати на всіх веб-сайтах.

Багато з віджетів містять часто використовувані функціональні можливості, такі як ефекти, утиліти і взаємодії. Набір JQuery UI містить широкий спектр зворотних викликів, атрибутів і функцій, які дозволяють взаємодіяти з ними в повній мірі.

## Контрольні питання

1. Для чого призначена мова розмітки HTML?
2. Перелічити переваги, що надає використання стилів CSS
3. Які функції покладено на мову клієнтського програмування JavaScript?
4. Для чого використовують мови серверного програмування?
5. Перелічити переваги від використання мови HTML5.
6. Які функції покладено на різні рівні специфікацій CSS?
7. Який набір функціоналу може містити бібліотека?
8. Що собою представляє JS-фреймворк?
9. Перелічити популярні JS-фреймворки.
10. Назвати переваги від використання бібліотеки jQuery?

## Використані джерела

1. Web-технології <https://htmlweb.ru/html/>
2. Огляд мови розмітки HTML для початківців <https://timeweb.com/ru/community/articles/chto-takoe-html>
3. HTML-довідник [https://schoolsw3.com/html/](https://schoolsw3.com/html/index.php)
4. CSS-довідник [https://schoolsw3.com/css/](https://schoolsw3.com/css/index.php)
5. JS-довідник [https://schoolsw3.com/js/](https://schoolsw3.com/js/index.php)
6. HTML Book <https://html5book.ru/html-html5/>
7. Керівництво по оформленню коду HTML/CSS <https://proglib.io/p/obuchenie-veb-razrabotke-rukovodstvo-po-oformleniyu-koda-html-css-2020-12-06>
8. Мови програмування для серверної веб-розробки <https://andreyex.ru/programmirovanie/7-luchshih-yazykov-programmirovaniya-dlya-servernoj-veb-razrabotki/>
9. Семантична верстка <https://htmlacademy.ru/blog/boost/frontend/semantics>
10. Селектори CSS. Види, групування та специфічність <https://itchief.ru/html-and-css/selectors>
11. Інструменти та веб-ресурси для веб-розробки <https://techblog.sdstudio.top/blog/instrumenty-i-veb-resursy-dlia-veb-razrabotki-100-plus>
12. Кращі фреймворки CSS <https://proglib.io/p/9-luchshih-freymvorkov-css-aktualnyh-v-2021-godu-2021-03-10>
13. Кращі JS-фреймворки та тенденції веб-розробки [https://medium.com/nuances-of-programming/лучшие-javascript-фреймворки-и-тенденции-веб-разработки-в-2021-году-2a35e348a12](https://medium.com/nuances-of-programming/%D0%BB%D1%83%D1%87%D1%88%D0%B8%D0%B5-javascript-%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA%D0%B8-%D0%B8-%D1%82%D0%B5%D0%BD%D0%B4%D0%B5%D0%BD%D1%86%D0%B8%D0%B8-%D0%B2%D0%B5%D0%B1-%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8-%D0%B2-2021-%D0%B3%D0%BE%D0%B4%D1%83-2a35e348a12)
14. Керівництво по використанню бібліотеки jQuery <https://itchief.ru/javascript/jquery-introduction>