

12. Інструменти веб-розробника для створення веб-додатків

Веб-додатки стали необхідністю для розвитку бізнесу. Це найшвидший спосіб залучити потенційних користувачів та переконати їх у необхідності ваших товарів чи послуг. Але створення веб-додатків і сайтів іноді може бути дуже клопіткою справою. Для спрощення та оптимізації життєвого циклу веб-розробників без шкоди для продуктивності призначені інструменти веб-розробки.

Веб-розробка - це безперервний життєвий цикл, що містить багато етапів та ітерацій. Починаючи з вибору мови програмування, відповідного фреймворку і закінчуючи пошуком надійного веб-сервера для розміщення програми, все це вимагає обізнаності в їх функціональностях і можливостях.

Існує багато типів інструментів для веб-розробки, які підтримують та спрощують той чи інший аспект розробки. Деякі з найбільш популярних інструментів:

- **Конструктори.** Пропонують безкоштовні шаблони користувацького інтерфейсу і плагіни. Надають доступ до коду елементів, щоб вносити зміни.
- **Редактори і середовища коду.** Мають вбудовані функції, такі як підсвічування синтаксису, налагодження та автоматичне завершення коду, що спрощує процес написання та редагування коду.
- **Платформи для дизайну та інтерфейсу.** Розробник може створити інтерфейс сайту у зручний спосіб. Багато конструкторів є інтуїтивно зрозумілими і для створення дизайну не потрібно глибоко володіти навичками з кодування. Весь процес базується на візуальному редагуванні готових шаблонів сайтів, які можна адаптувати під власні потреби.
- **Зовнішні бібліотеки для верстання.** Містять попередньо написані фрагменти коду та шаблони, які можна використовувати у коді.
- **Препроцесори.**
- **Інструменти розробника у браузерях.**
- **Веб-сервер.** Встановлює з'єднання між сервером та клієнтськими браузерами та обробляє запити користувачів.
- **Системи бази даних.** Допомогають створювати, редагувати та підтримувати базу даних, формувати різні запити для доступу до баз даних.
- **Локальне оточення.** Середовище на локальному комп'ютері для тестування та запуску веб-додатків і рішень, не вимагаючи підключення до Інтернету або розміщення на веб-хостингу.
- **Інструменти автоматизованої збірки.** Інструменти збирання допоможуть перетворити код розробки на готовий до роботи код, який без проблем працює на будь-якому пристрої або платформі.
- **Інструменти для тестування програмного забезпечення.** Тестування програмного забезпечення є критичним і часто виснажливим етапом завершення продукту та підвищує його точність. Автоматизація тестування зменшує кількість помилок, час пошуку та виправлення багів та отримання задовільних результатів.
- **Системи контролю версій.** Надають центральну платформу для керування та спільного написання коду.

Це лише деякі з багатьох інструментів, доступних для веб-розробників. На сьогодні щодня розробляються нові веб-фреймворки та інструменти. Навіть існуючі інструменти отримують поновлення.

Інструменти для розробки фронтенду

Розробка фронтенду є важливою частиною веб-технологій, що фокусується на тому, що бачать користувачі. Фронтенд-розробка вимагає технічного розуміння клієнтських мов, таких як HTML, CSS і JavaScript, знання принципів дизайну, палітри кольорів і схем, а також налагодження та тестування коду.

Фронтенд веб-розробники відповідають за розробку доступного, привабливого UI-UX дизайну для веб-сторінок та додатків. Вони вирішують, як виглядатиме веб-сайт. Інструменти та фронтенд-фреймворки нададуть величезну допомогу, оскільки вони можуть автоматизувати та спростити багато з цих завдань. Інструменти фронтенду допомагають прискорити процес розробки веб-додатку, надаючи можливість перетягування елементів і різні вбудовані функції для створення більш привабливого макета дизайну. До фронтенд-інструментів відносяться CSS-фреймворки, бібліотеки JavaScript, інструменти UI-UX дизайнера, бібліотеки інтерфейсу користувача, інтегроване середовище розробки та інструменти перетягування HTML.

Середовища для кодингу

Visual Studio Code (<https://code.visualstudio.com/>)

Безкоштовний редактор коду з відкритим вихідним кодом. Це інтелектуальний редактор коду, що має велику кількість функцій, таких як завершення коду, рефакторинг коду, підсвічування синтаксису, вбудований Git та підтримка тестування. Редактор Visual Studio налаштовується і дозволяє налагоджувати код за допомогою точок зупинки, стеків викликів та інтерактивної консолі. Пропонує широкий вибір пакетів та безкоштовних розширень для додаткових функціональностей чи можливостей. Visual Studio Code працює як на JavaScript, так і на Python, його можна використовувати для бекенд-кодингу.



Sublime Text (<https://www.sublimetext.com/>)

Кросплатформне програмне забезпечення з відкритим вихідним кодом для редагування коду та розмітки. Підтримує багато мов програмування та розмітки, все можна налаштувати відповідно до потреб проекту. Має продуманий, швидкий і ефективний інтерфейс, широку колекцію компонентів та інтелектуальний інструмент редагування. Надає такі можливості, як одночасне редагування, швидка навігація за файлами, символами, рядками та змінними, а також багато поєднань клавіш для плавного та гнучкого проектування додатків.



Платформи для дизайну та реалізації інтерфейсу

Сервіси в цій сфері розділені на дві групи за основним функціоналом: до першої відносяться платформи, що дають можливість створити безпосередньо дизайн - Figma, Adobe XD, Sketch, InVision Studio – тут можна розробити прототип та дизайн-макет сайту. Друга група сервісів має розширений функціонал, дизайнер може створити сайт з нуля: від прототипу до реалізації — Pixli, Webflow і Tilda

Figma (<https://www.figma.com/>)

Векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи. Працює у двох форматах: у браузері та як клієнтський додаток на десктопі користувача. Зберігає онлайн-версії файлів, з якими працював користувач. Популярний інструмент у дизайнерів, завдяки своїм можливостям:



- Хмарне зберігання файлів. Всі зміни у Figma зберігаються автоматично.
- Фрейми. Це окреме полотно, всередині якого дизайнери можуть проектувати екрани додатків. В наявності кілька стандартних розмірів під різні пристрої: всі види iPhone та найпопулярніші смартфони з Android.
- Компоненти. Ці елементи дозволять стандартизувати все, що є у макеті. За допомогою компонентів можна задати загальний стиль макету та у разі потреби змінити його.
- Багатокористувацький режим редагування. Всі файли знаходяться в хмарі і команда може одночасно працювати з сервісом.
- Умовна безкоштовність. Figma можна скористатися безкоштовно, але з обмеженнями для командної роботи. Приватних файлів може бути скільки завгодно. Для командної роботи доступні лише три файли та один проект.
- Історія версій зберігається 30 днів. Тариф Professional не має обмежень для командної роботи.

InVision Cloud (<https://www.invisionapp.com/>)

Універсальна онлайн-дошка для спільної роботи. Сервіс простий у використанні, має унікальні інструменти веб-дизайну, такі як бібліотека елементів, сучасні іконки, дошка фотографій, навігаційні потоки, прототипи для сайтів та інші традиційні елементи та методи дизайну. Інструмент Prototype для настільних та мобільних програм дозволяє створювати клікабельні елементи, а також добре підходить для створення анімації. Можна вибрати професійний тарифний план, якщо потрібні кілька прототипів для проекту та додаткові функції.



Novi Builder (<https://novibuilder.com/>)

Інтерактивний конструктор HTML, який пропонує безкоштовні шаблони користувацького інтерфейсу і плагіни. Надає доступ до вихідного коду елементів і дозволяє розробникам безпосередньо вносити зміни до графічного дизайну сайту. Novi Builder поставляється як вбудована функція або додаток для популярних браузерів, таких як Google Chrome і Opera. Рекомендується для досвідчених розробників, які мають глибокі знання, розуміння та досвід у створенні веб-додатків і сайтів.



Creative Tim (<https://www.creative-tim.com/>)

Інструмент фронтенд-розробки на основі Bootstrap, який надає компоненти інтерфейсу і інструментальні панелі, побудовані на JavaScript-фреймворках - Vue, Angular і React. Він дозволяє розробляти як веб- та мобільні додатки за допомогою готових шаблонів, попередньо закодованих розділів, наборів елементів інтерфейсу та наявних інструментів. Це універсальне рішення для всіх потреб фронтенд-розробника.



Envato (<https://www.envato.com/>)

Інструмент фронтенд-розробки, який підтримує HTML, CSS та JavaScript. Підтримує багато веб-фреймворків, надаючи шаблони для JavaScript-бібліотеки Vuejs, Laravel, Angular та Bootstrap. У ньому є колекція з тисяч готових HTML5-шаблонів і компонентів, які розробники можуть використовувати для створення інтерфейсів користувача. Envato пропонує оптимізовані методи кодування CSS та JavaScript, які допоможуть заощадити час на кодування та покращити показники швидкості роботи сторінок.



Onepage (<https://onepage.io/uk>)

Сучасний конструктор сайтів з України, який пропонує багато простих у використанні тем. Цей інструмент можна використовувати для необмеженої кількості доменів та проектів. Він надає широкий спектр доповнень, плагінів та стокових фотографій. Можна розробити професійний веб-додаток, використовуючи пропоновані аудіо- та відеоматеріали. Ідеальний інструмент для розробки веб-додатків для комерційних цілей.



WordPress (<https://wordpress.org/>)

Онлайн-платформа для розміщення динамічних сайтів на сервері з PHP/MySQL. Програма може бути використана як CMS для створення комерційного сайту. Його можливості включають текстовий редактор WYSIWYG, спільну роботу з кількома авторами, пермалінки, оптимізовані для пошукових систем, статичні сторінки, реєстрацію/логін.



Бібліотеки для CSS і JS

Bootstrap (<https://getbootstrap.com/>)

Популярна бібліотека HTML, CSS та JavaScript з відкритим вихідним кодом. Вона дозволяє створювати адаптивні мобільні та веб-додатки в Інтернеті. Bootstrap - це великий набір інструментів інтерфейсу користувача, що надає доступ до багатьох готових до використання компонентів. Bootstrap простий у налаштуванні, і досвідчений розробник може створити функціональний макет із адаптивним дизайном за короткий час. Є підтримка інших інструментів розробки сайтів, таких як Sass та JQuery, що розширює список пропонованих функцій.



Materialize (<https://materializecss.com/>)

Сучасна адаптивна CSS-платформа, що містить набір інструментів для верстки і готові компоненти, наприклад: сітка, каруселі, акордеон, адаптивне меню. Загалом є всі необхідні компоненти, виконані в красивому та сучасному дизайні для верстки сайтів.



CSS стилі підтримують принципи популярного напрямку у веб-дизайні – Material Design, розробленого компанією Google. Material Design – це інтерфейс мобільних та веб-додатків, що базується на стандартах дизайну Google. Основні його принципи:

- графічність (в основі лежать поліграфічний дизайн, канони друкарні, побудови ієрархій);
- метафоричність (текстури, тіні, баланс світла взяті із реального світу);
- наочність (всі іконки інтуїтивно зрозумілі);

динамічність (застосовуються послідовні перетворення та переходи). Material Design надає легку взаємодію з сайтом та наголошує на важливості контенту. Часто матеріал дизайн роблять у стилі мінімалізму, з мінімальним набором елементів дизайну. Буває так, що іконки та логотип – єдині нефункціональні елементи на сторінці.

Material UI (<https://mui.com/>)

React компонент для швидкої і легкої веб-розробки. Розробники використовують для створення програм бібліотеку компонентів Material UI. Її можна застосовувати при розробці додатків на Android та iOS, у тому числі для кросплатформових рішень на фреймворку Flutter та веб-додатків.



CSS та HTML-препроцесори

Зараз фронтенд-розробники все частіше використовують препроцесори. Це допоміжні технології, які є обхідними шляхами для обмежень CSS і надають додаткові функціональні можливості (наприклад, міксини), які можуть допомогти з масштабованістю та ефективністю.

SASS (<https://sass-lang.com/>)

Скриптова мова попередньої обробки, що компілюється в коди CSS і допоможе розширити CSS-функціональність. Веб-розробники можуть легко додавати змінні, множинне і багаторівневе успадкування та вкладеність. Розширені можливості, такі як керуючі директиви для бібліотек, автоматичне форматування виведення та зовнішні бібліотеки роблять Sass ідеальним для роботи з CSS. Надає доступ до багатьох корисних функцій для роботи з кольором, шрифтом та іншими компонентами інтерфейсу користувача. Дозволяє легко ділитися дизайном всередині та між проектами.



LESS (<https://lesscss.org/>)

LESS найстаріший у списку препроцесорів. Це надійний препроцесор і досить легкий для вивчення, оскільки синтаксис дуже схожий на звичайний CSS. Він має ті ж функції, що й у SASS, тільки під іншою назвою, включаючи різні функції: кросбраузерність, міксини, швидкість компіляції.



Stylus (<https://stylus-lang.com/>)

Універсальний препроцесор: двокрапки, крапки з комою і коми не обов'язкові. Щоб створити блоки коду не потрібні фігурні дужки, замість цього використовуються відступи. Користувачка база Stylus менша, ніж у LESS і SASS.



PostCSS (<https://postcss.org/>)

Інструмент, що автоматизує рутинні операції з CSS за допомогою розширень, написаних на JavaScript. Використано при розробці Вікіпедії, Facebook та GitHub. Один із найчастіше завантажуваних з npm інструментів для роботи з CSS. На відміну від Sass і LESS, PostCSS не є мовою шаблонів, що компілюються в CSS. PostCSS — платформа, де можна створити різні інструменти роботи CSS, зокрема можна створити і мову шаблонів, таку як SASS та LESS.



Всі корисні функції надаються розширеннями – невеликими програмами, які працюють з деревом об'єктів. Після того, як ядро перетворює CSS-рядок на дерево об'єктів, розширення по черзі аналізують і змінюють це дерево. Далі ядро PostCSS генерує новий CSS-рядок по дереву, яке було змінено розширеннями.

HTML препроцесори і шаблонізатори

Pug/Jade (<https://pugjs.org/>)

Швидкість у роботі - це один із найважливіших факторів. З'являються різні інструменти-прискорювачі, один із них: шаблонізатор Pug/jade – сучасний прискорювач для HTML, написаний на JavaScript для Node.js.



Pug відрізняється синтаксисом від HTML і дозволяє не робити типових помилок, які можна зустріти у звичайному HTML. При цьому Pug значно прискорить верстку, в тому числі за рахунок багаторазового використання одних й тих самих частин коду. Правити такий код надалі буде набагато простіше.

Haml (<https://haml.info/>)

Мова розмітки, яка дозволяє чисто і просто описати HTML веб-документа без використання вбудованого коду. Haml діє як заміна вбудованих систем шаблонізації, таких як PHP, ASP, ERB та мови шаблонізації, що використовуються в більшості



Ruby on Rails додатках. Haml допомагає уникнути прямої верстки HTML-шаблону, оскільки він сам по собі є описом HTML з деяким кодом, що генерує динамічний контент.

Можливості: підтримка пробілів, відформатована розмітка, виконання css інструкцій, можлива інтеграція Ruby коду

Інструменти розробника у браузерях

Панель інструментів розробника знаходиться у всіх популярних браузерах. Synchronb дозволяють редагувати HTML та CSS у режимі реального часу та налагоджувати JavaScript, переглядати аналіз продуктивності веб-сайту. Вкладка Network допоможе оптимізувати потоки завантаження, а тимчасова шкала дасть глибше розуміння того, що браузер робить у будь-який момент часу.

Щоб викликати цю панель можна застосувати кілька варіантів

1. Клавіатура. Windows – F12 (Ctrl + Shift + J), Mac OS (Cmd + Option + J)
2. Панель меню
 - Firefox. Відкрити меню_>> Інструменти розробки або Інструменти > Веб-розробка > Інструменти розробки
 - Chrome. Додаткові інструменти > Інструменти розробника
 - Safari. Розробка > Показати Web Inspector. Якщо ви не бачите меню "Розробка", зайдіть у Safari > Установки > Додатково, і перевірте, чи варто галочка навпроти "Показати меню розробки" .
 - Opera. Меню > Розробка > Інструменти розробника. Якщо ви не бачите меню "Розробка", увімкніть його, перейшовши до Меню > Інші інструменти > Показати меню розробника.

3. Контекстне меню. Натисніть правою кнопкою миші на будь-якій ділянці веб-сторінки (Ctrl-клік для Mac), з'явиться контекстне меню, в якому потрібно вибрати пункт Дослідити Елемент.

Інструменти для веб-розробки бекенда

Щоб забезпечити безперебійну роботу веб-сайту, потрібні правильні бекенд-фреймворки, відповідне середовище кодування, ідеальний хостинг та безперебійна система управління даними. Інструменти бекенда допомагають керувати базами даних, формувати запити та забезпечувати безперешкодне отримання даних при запиті користувача. Управління сесіями, HTTP-запити та контроль версій – все це підтримується різними інструментами бекенду.

Веб-сервери та бази даних

Apache (<https://httpd.apache.org/>)

Безкоштовне програмне забезпечення веб-сервера з відкритим вихідним кодом, яке встановлює з'єднання між клієнтською та серверною сторонами та доставляє файли між ними. Apache пропонує можливість налаштування, за допомогою модулів, які адміністратори можуть включати та вимикати за необхідності. Apache не підходить для сайтів з високою відвідуваністю, і якщо створюється сайт, який має контролювати та керувати великим навантаженням, рекомендується використовувати NGINX через його швидкість та оптимізовану продуктивність. Однак можливості налаштування та зручність використання Apache роблять його доречним рішенням для більшості розробників.



NGINX (<https://nginx.org/>)

Програмне забезпечення з відкритим вихідним кодом для обслуговування сайтів і додатків. Може використовуватися як зворотний проксі, балансувальник навантаження, HTTP-кеш, а також медіаплеєр і стрімер. Є надзвичайно швидким і ефективним і впевнено конкурує з Apache. NGINX може працювати як проксі-сервер для електронної пошти.



MySQL (<https://www.mysql.com/>)

Популярна реляційна база даних з відкритим вихідним кодом. Є частиною стека веб-розробки LAMP (Linux, Apache, MySQL, PHP/Python). MySQL безкоштовна, проста в налаштуванні та вивченні, швидко масштабується і є найкращим вибором для більшості розробників бекендів як система управління базами даних. Добре підходить для роботи з великими базами даних.



MongoDB (<https://www.mongodb.com/>)

База даних NoSQL, яка може зберігати та повертати величезні обсяги неструктурованих даних. Для цього вона сортує дані до документів, а не таблиці. Дозволяє створювати складніші структури, будувати ієрархії, визначати відносини та зберігати масиви. MongoDB може використовувати такі дані, як демографічні дані, історія відвідувань сайту та поведінка користувачів для створення індивідуального підходу до кожного користувача.



Redis (<https://redis.io/>)

Інструмент із відкритим вихідним кодом, який кешує дані у базі даних NoSQL. Підтримує більшість мов програмування, включаючи JavaScript, Java, Python, PHP тощо. Допомагає в роботі з великими базами даних та зменшити кількість запитів до бази даних, що підвищує швидкість роботи веб-додатку. Redis використовують для отримання мінімального часу відгуку на кешовані дані такі організації, як Twitter, Pinterest, Craigslist і GitHub.



Стеки для розгортання та розміщення веб-додатків.

LAMP (Linux, Apache, MySQL, PHP)

Стек LAMP є, ймовірно, найпоширенішим стеком, що використовується для розміщення веб-сайтів та веб-додатків, особливо сайтів малого та середнього розміру. Кожен з компонентів стеку LAMP має відкритий вихідний код і безкоштовний у використанні, що сприяло їхній популярності.



- Linux - операційна система забезпечує основу, де можуть працювати всі інші компоненти стека.
- Apache - це програмне забезпечення веб-сервера, яке він отримує та обробляє всі запити відвідувачів. Якщо клієнт запитує index.html або JPEG, Apache шукає файл на сервері та надсилає його назад до браузера відвідувача.
- PHP – це мова програмування для створення динамічних веб-сторінок. Може також означати мову Perl або Python, хоча вони не настільки поширені.
- MySQL – це база даних, що використовується для зберігання даних сайту.

LEMP (Linux, Nginx, MySQL, PHP)

Програмний стек LEMP може використовуватися для обслуговування динамічних веб-сторінок та веб-застосунків. Він є досить популярним серед сайтів WordPress. Оскільки всі компоненти з відкритим вихідним кодом, стек LEMP можна використовувати безкоштовно.



- Linux - операційна система забезпечує основу, де можуть працювати всі інші компоненти стека.
- Nginx (engine-x) - це програмне забезпечення веб-сервера, яке він отримує та обробляє всі запити відвідувачів. Nginx здатний обробляти величезний трафік, що робить LEMP найкращим вибором для хостингових компаній
- PHP – це мова програмування для створення динамічних веб-сторінок. Може також означати мову Perl або Python, хоча вони не настільки поширені.
- MySQL – це база даних, що використовується для зберігання даних сайту. Може також означати базу даних MariaDB - відгалуження від системи управління базами даних MySQL, що використовується для бази даних та PHP для завантаження динамічних веб-сторінок у браузері.

MEAN (MongoDB, Express, Angular, Node.js) (<https://meanjs.org/>)

Стек MEAN є доволі популярним і ним користуються частіше, ніж стеком LAMP. Разом програми використовуються для створення потужних веб-додатків.



- MongoDB – це база даних NoSQL,
- Express.js – це веб-фреймворк, що працює на Node.js,
- Angular – це платформа розробки на стороні клієнта,
- Node.js – середовище виконання.

Переваги стеку MEAN

- Єдина мова використовується у всьому додатку
- Краща продуктивність швидше, ніж у стека LAMP
- Підтримка та часте застосування використання архітектури MVC
- Популярність JavaScript

XAMPP (кросплатформовий, Apache, MariaDB, PHP або Perl) (<https://www.apachefriends.org/>)

Один з найкращих дистрибутивів Apache, який допомагає розробникам створювати локальний веб-сервер для тестування та розгортання. Стек розроблено, щоб в простий спосіб встановити та запустити локальний сервер. Оскільки стек є кросплатформним, його можна використовувати на Windows, MacOS або Linux.



Він отримав багато доповнень, таких як Drupal, Joomla, WordPress, Magento, SuiteCRM тощо, щоб розпочати проект за лічені хвилини.

WAMP (Windows, Apache, MySQL та PHP) (<https://www.wampserver.com/>)

Єдина різниця між WAMP та LAMP - операційна система, що стоїть за цими стеками. Таким чином, все вищезгадане для LAMP також виконується WAMP, але тільки в системі Windows.

AMPPS (Apache, MySQL, MongoDB, PHP, Python, Softaculous) (<https://ampps.com/>)

AMPPS - це комбінація стека LAMP (Linux), WAMP (Windows) та MAMP (Macintosh) від Softaculous, що складається з Apache, MySQL, MongoDB, PHP (також Perl та Python).



Стек поставляється з великою кількістю попередньо встановленого програмного забезпечення та програм з відкритим вихідним кодом. Забезпечує безпечне середовище для управління базами даних, управління доменами, управління програмами для полегшення розробки. У цей стек входить понад 425 програм.

Laragon (<https://laragon.org/>)

Laragon забезпечує швидкий та простий спосіб створення ізольованої платформи розробки для Windows. Платформа поставляється з встановленими популярними програмами, такими як Node.js, PHP, Apache, Composer і MySQL. Вона підтримує PHP, NodeJS, Python, Java, Ruby та багато інших мов. Надзвичайно легкий та ефективний, Laragon є відмінним інструментом для розробки та управління сучасними програмами.



Інструменти автоматизованої збірки

Під час створення проекту необхідно формувати пакети, які працюють швидко, керують залежностями, автоматизують завдання, завантажують зовнішні модулі тощо. Інструменти, які дозволяють перетворити код розробки на робочий код, називаються інструментами збирання. Інструменти збирання допоможуть перетворити код розробки на готовий до роботи код, який без проблем працює на будь-якому пристрої або платформі.

Розробники фронтенду в основному працюють з наступними типами інструментів збирання:

- Менеджери пакетів.
- Виконавці завдань.
- Завантажувачі модулів.
- Збирачі модулів.

Всі ці інструменти запускаються в командному рядку, тому вони не мають графічного інтерфейсу користувача.

Менеджери пакетів

NPM (Node Package Manager) (<https://www.npmjs.com/>)

Менеджер пакетів Node для JavaScript, що допомагає розробникам NodeJS виявляти пакети компонентів, що багаторазово використовуються, і збирати їх новими потужними способами. Його можна використовувати як утиліту командного рядка для взаємодії з репозиторієм, який допомагає працювати з пакетом. Менеджер проекту може використовувати подібний робочий процес для управління публічним та приватним кодом серед команди. Він надзвичайно простий у використанні та виявляється потужним інструментом управління проектами для середніх та великих команд.



При установці Node.js у свою систему, npm встановлюється автоматично, і можна отримати доступ до нього з інтерфейсу командного рядка. За допомогою npm можна встановити будь-який пакет Node.js за допомогою однієї команди. Ключові особливості:

- Простий процес встановлення.
- Крос-платформне програмне забезпечення (Windows, Linux, macOS, SmartOS та ін.).
- Сотні тисяч пакетів.
- Ефективне керування залежностями за допомогою файлу package.json.
- Декілька варіантів конфігурації (через командний рядок).
- Велика документація та корисна спільнота.

YARN (<https://yarnpkg.com/>)

Інтерфейсний менеджер пакетів, який можна використовувати як альтернативу npm. Метою створення Yarn є вирішення низки проблем, з якими зіткнулися розробники при використанні npm, а саме:



- Установка пакетів була досить швидкої і послідовної.
- Проблеми з безпекою, оскільки npm дозволяє пакетам запускати код під час встановлення.

Оскільки Yarn сам є пакетом Node.js, необхідно встановити Node.js, перш ніж використовувати Yarn. Yarn це новий клієнт командного рядка, що завантажує модулі з реєстру npm. У самому реєстрі нічого не змінюється - можна завантажувати та публікувати модулі так само, як і раніше.

Yarn прискорює процес складання, кешуючи кожен пакет, тому не потрібно завантажувати залежності кілька разів. Щоб ще більше скоротити час збирання він виконує паралельні операції.

Ключові особливості:

- Крос-платформний інструмент (Windows, Linux, macOS) , який можна встановити як звичайний пакет Node.js. Сумісний з усіма пакетами Node.js.
- Величезна екосистема із тисячами плагінів. Якісна кодова база з використанням найкращих практик Node.js. Легка розуміння документація.
- Швидкий час збирання. Використовує потоки вузлів для прискорення операцій.
- Додаткова безпека за рахунок використання контрольних сум для перевірки цілісності пакетів. Flat режим, щоб уникнути дублювання.
- Автономний режим.

Виконавці завдань

JS-системи управління завданнями (Task Manager) для автоматизації рутинних завдань, що виникають при веб-розробці як з боку фронтенда так й бекенда. До ядра виконавця можна долучити додаткові модулі і плагіни. Для системи створюються певні завдання, що описані мовою Javascript, система виконує ці завдання у потрібній послідовності.

Системи виконання завдань відрізняється від менеджерів пакетів, їх не можна використовувати для керування залежностями. Вони потрібні у випадку, якщо потрібно виконувати одні й ті самі завдання під час кожного процесу складання.

Типові завдання систем управління завданнями

- **Мініфікація коду.** Код, який пише розробник зазвичай є добре структурований, зрозумілий і добре сприймається. Але на робочому сервері, код в такому вигляді буде довше завантажуватися через надлишковість інформації у вигляді відступів, коментарів тощо. Gulr дозволяє усунути все зайве з коду, підготувати його для того, щоб це можна було викласти на робочий сервер.
- **Об'єднання коду з різних файлів на один.** Під час роботи за протоколом http кожен запит до файлу - це додатковий час завантаження сторінки. Також, писати програму значно простіше, якщо її код розбитий на модулі та незалежні частини. Для збільшення швидкості завантаження документа варто об'єднувати код із CSS, Javascript файлів в один файл. Завантажити один файл простіше і швидше, ніж завантажити кілька файлів.
- **Робота з CSS препроцесорами.** Системи виконання завдань надають можливість використовувати препроцесори і компіляція у кінцевий CSS-код відбувається в швидкий і зручний спосіб.
- **Підтримка нових стандартів Javascript.** Якщо відвідувач сайту користується старими браузером, там нові стандарти не будуть працювати. За допомогою Gulr ви можете вирішити це завдання.

Grunt (<https://gruntjs.com/>)

Менеджер завдань JavaScript, написаний поверх NodeJS. Автоматизує більшість повторюваних завдань веб-розробки. Після встановлення та налаштування завдання за допомогою файлу Grunt, такі завдання, як мініфікація, компіляція, модульне тестування та літинг, стають надзвичайно простими. Grunt має великий репозиторій плагінів, що забезпечує відмінне середовище кодування для розробників. Grunt підтримує фронтенд веб-інструменти, такі як Sass або RequireJS, для виконання попередньо налаштованих завдань.



Grunt - це пакет Node.js, тому його можна встановити за допомогою npm, Yarn або іншого менеджера пакетів Node.js. Grunt зберігає власні залежності, необхідні для виконання визначених завдань, у файлі package.json. Можна визначити завдання в Gruntfile, який запускається під час кожного процесу складання та автоматично виконує кожну визначену задачу.

Ключові особливості:

- Крос-платформний інструмент командного рядка, що працює у будь-якій операційній системі.
- Простий процес налаштування.
- Величезна екосистема із сотнями плагінів для додавання інструментів зовнішнього інтерфейсу (таких як Sass, Jade, JSHint, Handlebars, RequireJS та інші), які виконують попередньо налаштовані завдання.
- Асинхронні задачі.
- Велика документація. Широко використовується.

Gulp (<https://gulpjs.com/>)

Автоматизований інструмент для виконання завдань, а також найсильніший конкурент Grunt. Gulp використовують для автоматизації повторюваних інтерфейсних завдань, таких як попередня обробка CSS, автоматичне додавання префіксів, оптимізація зображень тощо. Це також пакет Node.js, який можна встановити за допомогою менеджерів пакетів npm і Yarn. Можна визначити завдання в Gulpfile і налаштувати залежності, пов'язані з завданнями у файлі package.json.



На відміну від Grunt, Gulp використовує ефективнішу техніку автоматизації, яка дозволяє скоротити час збирання. Grunt використовує тимчасові файли для обробки завдань, Gulp виконує операції у пам'яті без запису у тимчасові файли. Ці операції в пам'яті називаються потоками вузлів, які можуть заощадити багато часу, особливо якщо потрібно обробляти кілька завдань при кожній збірці.

Завантажувачі і збиральники модулів

Browserify (<https://browserify.org/>)

Завантажувач модулів Node.js, який дозволяє об'єднувати клієнтські залежності та завантажувати їх як один JavaScript в браузер користувача. Менеджери пакетів, такі як npm та Yarn, завантажують модулі на стороні сервера за допомогою функції



Node.js `require()`, призначеної для завантаження модулів. Browserify надає метод `require()` на стороні клієнта, що може призвести до величезного збільшення продуктивності.

При використанні Browserify браузер користувача повинен завантажити лише один статичний файл JavaScript, що містить всі залежності від яких залежить проект. Можна додати свій зв'язаний JavaScript як тег `<script>` на сторінку. Оскільки Browserify є модулем Node.js та реалізацією CommonJS API (аналогічно `npm`), його можна використовувати тільки для завантаження модулів Node.js, але не для інших типів файлів JavaScript (або інших).

Ключові особливості:

- Поєднує всі залежності Node.js в один файл.
- Прискорює модульні програми, які використовують кілька модулів Node.js.
- Дозволяє зовнішні вимоги (можна вимагати модулі з інших елементів `<script>`).
- Дозволяє за необхідності розділяти пакети.

Webpack (<https://webpack.js.org/>)

Розширений збирач модулів, який дозволяє об'єднати всі залежності та завантажити їх як статичні ресурси у браузер. Webpack є модулем Node.js, встановити його можна за допомогою диспетчера пакетів `npm` або `Yarn`.



Browserify об'єднує лише модулі Node.js, Webpack може обробляти будь-які файли інтерфейсу, такі як `.html`, `.css`, `.js`, `.scss`, зображення та інші ресурси. Webpack аналізує проект та будує граф залежностей. На основі графіка залежностей він поєднує файли та модулі в один або кілька статичних файлів, які можна додати на свою HTML-сторінку.

Ключові особливості:

- Декілька варіантів конфігурації.
- Поділ коду на дрібніші частини, які можуть завантажуватися асинхронно.
- Усунення мертвого коду.
- Заміна гарячого модуля.
- Підтримка вихідних карток.
- Величезна екосистема з багатим інтерфейсом плагінів.

Крім поширених рішень, на ринку також з'являються нові інструменти, такі як менеджер пакетів `npm` (альтернатива `npm` та `Yarn`), збирач модулів `Parcel` (альтернатива `Webpack`) та накопичувальний модуль пакетування `ES` (за аналогією з `Browserify`).

Системи контролю версій

Системи контролю версій важливі, тому що допомагають відстежувати зміни на сайті, щоб легко повернутись до попередніх версій, якщо щось піде не так.

Git (<https://git-scm.com/>)

Безкоштовна розподілена система контролю версій із відкритим вихідним кодом, призначена для швидкої та ефективної обробки будь-яких проектів, від малих до великих. Git простий у вивченні та має блискавичну продуктивність. Він перевершує такі інструменти



SCM, як Subversion, CVS, Perforce і ClearCase, завдяки таким функціям, як локальне оточення, зручні робочі області та численні робочі процеси.

GitHub (<https://github.com/>)

Спеціалізована платформа для спільної роботи з відкритим вихідним кодом, що забезпечує контроль версій. Є найпотужнішим інструментом для професіоналів з великою бібліотекою знань. GitHub є спільнотою для розробників, де вони можуть обмінюватися інформацією та співпрацювати: ділитися та зберігати код, брати участь у проектах з відкритим вихідним кодом та відстежувати зміни. Його можна використовувати як для контролю версій власного проекту, так й для створення профілю як розробника, вносячи внесок у проекти інших.



Github Client (<https://desktop.github.com/>)

Безкоштовний GUI-клієнт Git з відкритим вихідним кодом. Github Client мінімізує витрати на створення програм на кількох платформах. Завдяки використанню програмного каркасу Electron розробники можуть швидко писати кросплатформні десктопні програми, використовуючи JavaScript, HTML та CSS. Можна додавати локальні репозиторії або створювати нові за короткий час. Це надає розробникам переваги уніфікованого кросплатформного досвіду. Github Client можна використовувати як редактор коду, інтегрований з Git.

Bitbucket (<https://bitbucket.org/>)

Веб-сервіс для хостингу проектів та їх спільної розробки, заснований на системі контролю версій Git. За призначенням і основним пропонованим функціям аналогічний до GitHub, від якого відрізняється з одного боку меншою базою користувача, а з іншого, має певні переваги в плані розміщення непублічних репозиторіїв — можливістю їх безкоштовного хостингу з обмеженням на розмір команди не більше п'яти осіб і меншою орендною платою при більшому розмірі команди, а також управління правами доступу на рівні окремих гілок проекту. Якщо основні переваги GitHub лежать у галузі соціалізації програмування, Bitbucket більше орієнтований на невеликі закриті команди розробників. Слоган сервісу "Bitbucket - це Git-рішення для професійних команд".



Валідація та тестування

Не менш важливими є інструменти валідації та тестування коду на етапі розробки. Літери – це інструменти для дослідження програмного коду з метою виявлення помилок програмування, вад, порушень стилю, а також сумнівних чи підозрілих виразів.

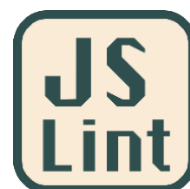
ESLint (<https://eslint.org/>)

ESLint — літер, який допомагає покращити якість коду та заздалегідь виправляє помилки самостійно, щоб уникнути їх появи під час виконання коду. Це допомагає уникнути важких проблем налагодження в майбутньому.



JSLint (<https://www.jshint.com/>)

Інструмент статичного аналізу коду, який використовують для розробки програмного забезпечення для перевірки, наскільки вихідний код JavaScript відповідає правилам кодування. Він надається в основному як веб-додаток на основі браузера, який доступний через домен. Є адаптація командного рядка.



Mocha (<https://mochajs.org/>)

Багатофункціональний тестовий фреймворк JavaScript, який працює на Node.js та у браузері, що робить асинхронне тестування простим та захоплюючим. Тести Mocha виконують послідовно, що забезпечує гнучку та точну звітність.



Jasmine (<https://jasmine.github.io/>)

Поведінкова платформа розробки для тестування коду JavaScript. Вона не залежить від будь-яких інших фреймворків JavaScript. Для неї не потрібний DOM. У Jasmine чистий, очевидний синтаксис, можна легко писати тести.



QUnit (<https://qunitjs.com/>)

Бібліотека JavaScript для тестування коду. Коли тестують код використовують певні функції, які надають QUnit. Потім QUnit їх запускає. QUnit може запускати тести у браузері або Node.js.



Jest (<https://jestjs.io/>)

Бібліотека JavaScript для створення, запуску та структурування тестів. Jest — пакет NPM. Можна встановити його у будь-який проект JavaScript. Це один із найпопулярніших тестових раннерів і його за замовченням обирають для проектів React.



Ava (<https://github.com/avajs>)

Запускник тестів для Node.js. Він дозволяє запускати JavaScript тести одночасно. Ava — запускник із простим синтаксисом, відсутністю неявних глобальних змінних, підтримкою промісів, асинхронних функцій та observable.

Висновок

Веб-розробка — це спільнота веб-розробників, кодерів і дизайнерів. Для задоволення всіх потреб розробляється багато фреймворків і інструментів веб-розробки. Тому непросто вибрати набір інструментів для проекту. Але зараз багато редакторів, препроцесорів, розширень та лінтерів, які полегшать роботу на будь-якому проекті.

Згадані вище інструменти розробки веб-додатків — найкращі як для досвідчених розробників, так й для початківців.

Джерела інформації

<https://clickfraud.ru/top-20-instrumentov-veb-razrabotki-dlya-sozdaniya-veb-prilozhenij/>

<https://sky.pro/media/instrumenty-dlya-veb-razrabotki/>

<https://senior.ua/articles/6-luchshih-instrumentov-sborki-dlya-frontendrazrabotki>