

*Поліщук Марина,
здобувачка першого (бакалаврського) рівня вищої освіти
фізико-математичного факультету
Ярмоленко Тетяна,
асистент кафедри комп'ютерних наук та інформаційних технологій,
Житомирський державний університет імені Івана Франка,
м. Житомир, Україна*

ГРАФІЧНІ МОЖЛИВОСТІ JAVA SCRIPT

JavaScript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв вебсторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Технологія мови JavaScript дуже проста та зручна. Всі операції, які можливо виконувати програмою JavaScript, описують дії над добре відомими та зрозумілими об'єктами, які є елементами робочої області програми броузера Netscape Navigator і контейнерами мови HTML. Тобто існують об'єкти з множиною властивостей та множиною функцій над об'єктами, які ще називають методами. Крім атрибутів об'єкт та метод, існує ще і третій – подія. Події орієнтовані на роботу в Web, наприклад, завантаження сторінки в робочу область Navigator'a, вибір гіпертекстового зв'язку і т.д. [5]. Використовуючи параметр подія можна організувати перегляд динамічних об'єктів: управління багатовіконним інтерфейсом, рядок, який біжить і т.д.

Методи об'єктів та властивості об'єктів.

Кожний із класу об'єктів має функції управління об'єктами класу – метод. Найголовнішими методами є ті, які дозволяють переназначати занесення об'єктів. Робиться це звичайно по операції присвоєння. Всі типи операторів, котрі підтримують відомі мови програмування, реалізовані JavaScript (+, -, *, /, %, «, », +=, -=, ...).

Можливості Java Script

Можливості JavaScript контроль роботи браузера (виведення діалогових вікон, відкриття / закриття вікон браузера, керування режимами прокручування та розмірами вікон) взаємодія з вмістом документів (об'єкт Document - читання / зміна частин HTML-документа, об'єкт Form - читання / зміна вмісту його об'єктів Button, Checkbox, Hidden, Password, Radio, Reset, Select, Submit, Text і Textarea і т.д.) взаємодія з користувачем (можливість визначати обробники подій,

перевірка інформації, що вводиться, управління відсиланням вмісту форм) виконання довільних математичних обчислень + засоби роботи зі значеннями дати та часу.

JavaScript. Поняття, що використовуються Об'єкт – сукупність властивостей, методів, колекцій та подій, що надаються браузером в рамках об'єктної моделі Властивість – змінна в рамках об'єкта, що використовується для отримання значення або встановлення нового Метод – функція, що надається об'єктом для виконання будь-яких дій Колекція – впорядкований набір властивостей Подія – будь-яка дія користувача або момент роботи браузера (для реакції на події створюються обробники подій)[4].

Переваги і недоліки:

Javascript популярний не випадково, а завдяки своїм безперечним достоїнствам.

Незамінність для веб-розробки. Підтримка скриптів усіма популярними браузерами; повна інтеграція з версткою сторінок (HTML+CSS) та серверною частиною (backend).

Швидкість роботи та продуктивність. Javascript дає змогу частково обробляти веб-сторінки на комп'ютерах користувача без запитів до сервера. Це заощаджує час і трафік, знижує навантаження на сервер.

Потужна інфраструктура (екосистема). Перші 10 років цього не було і близько. Потім кількість готових рішень у відкритому доступі так зросла, що працювати з Javascript та його фреймворками стало задоволення.

Простота та раціональність застосування. Просте завдання можна вирішити за 5 хвилин, не треба виконувати зайву роботу. Для складних завдань є варіанти вирішення, можна підібрати найкраще, адаптувати.

Зручність інтерфейсів користувача. Заповнення форм, вибір дій, активація кнопок, перевірки введення, реагування на наведення/кліки миші тощо Це дає приголомшливий рівень юзабіліті.

Легкість освоєння. Можливо, це справа звички. Спочатку код може здатися складним, але до його синтаксису, логіки швидко звикаєш. Відображення дій відчутно додає ентузіазму [3].

Як і будь-яка мова програмування, Javascript має деякі недоліки (обмеження).

Немає можливості читання та завантаження файлів. Це обмеження функціональності за клієнта. Головна причина – міркування безпеки.

Не сувора типізація та вільне трактування. Мова ігнорує явні нестиківки. Наявна різна інтерпретація даних. Немає можливості раннього виявлення помилок. Усі недоліки виявляються вже етапі роботи.

Нема підтримки віддаленого доступу. Тому не можна використовувати мову для мережних програм. За це Javascript навіть не вважають повноцінною мовою програмування.

Доступність для зловмисників. У вільну скриптову мову найпростіше вбудувати фрагмент шкідливого коду, який може зашкодити користувачеві. Надія тільки на антивірус та фаєрвол.

Слід зазначити, що спільнота користувачів Javascript активно покращує мову, усуваючи багато недоліків. Вузьких місць стає дедалі менше. Браузери постійно вдосконалюють роботу із JS. Вихід HTML 5 дав новий поштовх розширення можливостей скриптів.

При виборі професії web-програміста без JavaScript не обійтися. Знадобляться також знання з HTML та CSS, ці інструменти зазвичай працюють у зв'язці. Швидкому освоєнню мови сприяють:

- нескладність синтаксису та раціональність, інтуїтивна зрозумілість коду;
- простота та доступність інструментів (мінімум: текстовий редактор та браузер);
- наявність в інтернеті великої кількості матеріалів для навчання та практики.

JavaScript, наразі, є однією з найпопулярніших мов програмування в інтернеті [1]. В перші роки існування, більшість професійних програмістів скептично ставилися до мови, цільова аудиторія якої складалася з програмістів-аматорів [2]. Поява AJAX змінила ситуацію та звернула увагу професійної спільноти до мови, а її подальші модифікації за стандартами ES6+ внесли багато корисних можливостей, яких не вистачало для ефективного програмування. В результаті, були розроблені та покращені багато практик використання JavaScript (зокрема, тестування та налагодження), створені бібліотеки та фреймворки, поширилося використання JavaScript поза браузером.

Оскільки JavaScript є інтерпретованою мовою програмування, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій. Типізація вважається одною з ключових проблем JavaScript, тому восени 2012 року, компанія Microsoft презентувала мову програмування TypeScript, що компілюється в JavaScript та містить декілька важливих для програмістів доповнень, що полегшують розробку.

При розробці великих і нетривіальних вебзастосунків з використанням JavaScript, критично важливим є доступ до інструментів відлагодження. Оскільки браузер, від різних виробників, дещо відрізняються у поведінці JavaScript і реалізації Об'єктної моделі документа, необхідно мати відлагоджувач для кожного браузера, якщо вебзастосунок орієнтовано на нього.

На даний час Firefox, Opera, Google Chrome, Edge та Safari мають зневаджувачі для себе.

Також існують такі корисні інструменти, як:

ESLint [Архівовано 7 жовтня 2017 у Wayback Machine.] – перевірка якості коду, що сканує JavaScript-програму, шукаючи вади у коді;

Prettier [Архівовано 13 вересня 2017 у Wayback Machine.] – автоматичне форматування коду у коректний вигляд;

Babel [Архівовано 15 листопада 2016 у Wayback Machine.] – компілятор JavaScript-коду до старіших версій стандарту ECMAScript, який допомагає

розробникам використовувати найновіші можливості мови для оточення, що не встигло реалізувати останній стандарт.

Кожен блок сценарію інтерпретатор розбирає окремо. На вебсторінках, коли треба комбінувати блоки JavaScript та HTML, синтаксичні помилки знайти легше, якщо зберігати функції сценарію в окремому блоці коду, або (ще краще) використовувати багато малих, пов'язаних між собою .js файлів. Таким чином, синтаксична помилка не спричинятиме «падіння» всієї вебсторінки та дозволить сповістити користувача про проблему.

Мова Java Script являє собою мову сценаріїв для Web. На сьогодні підтримується броузерами Netscape та Microsoft Explorer. Java Script є підмножиною мови Java, але не є частиною мови Java. Корпорація Sun не підтримує мову Java Script. Через свою обмеженість JavaScript не може бути використана для програмування багатьох складних функцій. Вона призначена для програмістів та користувачів, вони хочуть використати та застосувати нові функціональні можливості мови HTML.

JavaScript – це мова програмування, що дозволяє зробити Web -сторінку інтерактивною, тобто такою що реагує на дії користувача.

Ядро JavaScript включає цілий ряд функцій, які дають наступні можливості:

- зберігати дані в змінних;
- активувати частину коду у відповідності з певними сценаріями, які здійснюються на сторінці сайту;
- створювати контент, який оновлюється автоматично;
- управляти мультимедійними можливостями (працювати з відео, анімувати зображення).

Усього кілька рядків коду скрипта роблять дивовижні речі. Саме тому мова настільки популярна серед розробників.

Список використаних джерел та літератури

1. Власій О. О. Комп'ютерна графіка. Обробка растрових зображень: навчально-методичний посібник. Івано-Франківськ: ДВНЗ «Прикарпатський національний університет імені Василя Стефаника», 2015. 72 с. URL: http://personal.pu.if.ua/depart/olesia.vlasii/resource/file/Vlasii_Dudka_Graph.pdf .

2. Abbott D., Djirdeh H., Accomazzo A., Shoemaker S. Fullstack React Native: create beautiful mobile apps with JavaScript and React Native. Kindle Edition, 2019. 690 p.

3. Boduch A., Derks R. React and React Native: a complete hands-on guide to modern web and mobile development with React.js. Packt Publishing, 2020. 526 p.

4. Duckett J. HTML and CSS: Design and Build Websites. Wiley, 2014. 512 p.

5. Goransson A. Efficient Android Threading: Asynchronous Processing Techniques for Android Applications.. O'Reilly Media. 2014; 280 p.