

ПРЕИМУЩЕСТВА И РАЗЛИЧИЯ TYPESCRIPT И JAVASCRIPT: ЧТО ВЫБРАТЬ ДЛЯ НАЧИНАЮЩИХ

*А.Г. Буймистров, И.И. Гламаздин,
Д.В. Горбачев
БИП – Университет права и социально-
информационных технологий,
Белорусский государственный университет
информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Один из первых вопросов, который стоит перед любым желающим изучить язык программирования высокого уровня, какой выбрать путь и с чего начать. Попробуем прояснить ответ на поставленный вопрос путем сравнения языков Typescript и JavaScript.

JavaScript – это язык программирования, изначально разработанный для динамического HTML в веб-браузерах. Он использовался для встраивания «логики» веб-страницы, чтобы можно было оценивать взаимодействие с пользователем, изменять или генерировать контент. JavaScript, наряду с HTML и CSS, являются стандартными инструментами веб-разработки, хотя JavaScript также используется на серверах или в микроконтроллерах [1].

Первоначально JavaScript был опубликован Netscape в 1995 году как LiveScript и переименован в JavaScript из-за сотрудничества с Sun Microsystems (тогда разработчиком Java). Хотя Java и JavaScript имеют разные базовые концепции и разрабатывались независимо друг от друга, изменение названия было направлено на то, чтобы сделать JavaScript более популярным за счет использования уже зарекомендовавшего себя бренда «Java».

JavaScript – это язык сценариев, оснащенный стандартизированным языковым ядром ECMAScript, который при необходимости может использоваться для объектно-о-

риентированного, процедурного или функционального программирования. Ядро языка ECMAScript описывает динамический, типизированный, объектно-ориентированный, но бесклассовый язык сценариев. В общем и целом использование API – это безопасный и проверенный способ подключения веб-сервисов за считанные секунды. Он позволяет приложению расширить функциональность за счет извлечения информации из других приложений. Но вас может беспокоить вопрос о том, как API отправляет запрос, загружает данные и передает их в определенном формате. Ответ на этот вопрос заключается в том, что это зависит от того, как вы построили API для своего приложения. В этой статье мы обсудим два распространенных API для соединения двух приложений для передачи данных [2].

Typescript – это язык программирования, разработанный Microsoft на основе ECMAScript – стандарта JavaScript. Typescript считается надмножеством JavaScript. Это означает, что любой код JavaScript также является допустимым кодом Typescript, и что в Typescript также можно использовать общие библиотеки JavaScript, такие как Angular JS или Ionic. Язык сценариев со строгой типизацией использует для описания данных такие понятия, как ориентация объекта и вывод типа, а также может обрабатывать общие типы данных.

Microsoft хотела исправить недостатки языка сценариев JavaScript в области разработки крупномасштабных приложений. В поисках более масштабной альтернативы JS Microsoft инициировала разработку Typescript, основанного на JavaScript.

Разработчики, в том числе главный разработчик Андерс Хейлсберг, стремились разработать Typescript таким образом, чтобы не подвергать опасности совместимость и независимость JavaScript от платформы. Вот как Typescript был оснащен расширенным компилятором JavaScript, который может переводить код в JavaScript, чтобы Typescript также мог работать в браузере.

Чтобы получить четкое представление о преимуществах и недостатках двух языков программирования, необходимо провести прямое сравнение их основных возможностей и характеристик.

JavaScript – это легкий язык сценариев, который может вычислять, проверять и изменять данные, чтобы сделать веб-страницы интерактивными. Он также может добавлять динамический текст в документы HTML и CSS. Некоторые разработчики из Netscape решили разработать язык для модификации

HTML-компонентов браузера (если вы не слышали об этом, в то время это была настоящая сенсация). Так был разработан JavaScript. Пожалуйста, помните, что JavaScript и Java не имеют ничего общего друг с другом. Внедрение «Java» в JavaScript было просто стратегией брендинга.

В перспективе JavaScript будет продолжать развиваться и приобретать новые функции. Инновации на основе JavaScript, такие как библиотеки React и React Native, а также другие популярные фреймворки, показали, насколько полезными они могут быть при веб-разработке на стороне сервера и на стороне клиента.

В последние годы JavaScript был дополнительно адаптирован для работы в серверной среде (Node.js). Бэкэнд-разработчики знакомы с фронтендом точно так же, как разработчик фронтенда знаком с бэкэнд-логикой. Он существует уже долгое время и будет продолжать существовать, а большое сообщество разработчиков открытого исходного кода является чрезвычайно ценным ресурсом.

Многие фреймворки и библиотеки разрабатываются на основе JavaScript, что упрощает реализацию общих функций.

Это динамический язык, целью которого является повышение производительности труда программиста, позволяя создавать функции с меньшим количеством кода. Это один из самых простых языков программирования для изучения.

Typescript обладает всеми функциями объектно-ориентированного языка программирования, включая классы, интерфейсы, наследование, модули и так далее. Мы можем писать код на Typescript как для клиентской, так и для серверной разработки. Все элементы JavaScript поддерживаются Typescript. Это позволяет разработчикам использовать Typescript с существующим кодом JavaScript. Здесь мы можем легко использовать все фреймворки, инструменты и библиотеки JavaScript.

Код Typescript не выполняется сразу в браузере. Программы Typescript всегда начинаются с JavaScript и заканчиваются на JavaScript. Поэтому нам нужно только понимать JavaScript, чтобы использовать его в Typescript. Для выполнения код Typescript компилируется и преобразуется в эквивалент JavaScript. Это называется транспилингом. Браузеры могут читать код TypeScript и отображать вывод с кодом JavaScript.

TypeScript можно использовать как JavaScript для изменения DOM и добавления или удаления элементов. Typescript включает в себя мощный компилятор JavaScript, который

может преобразовывать код в JavaScript, чтобы Typescript также можно было выполнять в браузере.

В конечном итоге выбор между Typescript и JavaScript зависит от доступных вам ресурсов и самого проекта. В рамках больших коммерческих проектов предпочтителен Typescript, так как он вам позволит писать меньшее количество тестов и создаст контекст, в котором вы не будете вынуждены тестировать JavaScript код в рантайме, чтобы словить какую-нибудь глупую ошибку при выборке данных пользователя, у которого поле «телефон» будет являться необязательным и так далее. В рамках небольших проектов или лендингов, где JavaScript будет нужен для написания анимаций, отсылки письма через SMTP-провайдер – использование Typescript не будет хорошим решением. В таком случае предпочтительнее выбрать JavaScript.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Learn. javascript [Electronic resource]. – Mode of access: <https://learn.javascript.ru>. – Date of access: 06.03.2023.
2. O'Reilly [Electronic resource]. – Mode of access: <https://www.oreilly.com/library/view/pro-javascript-html/8781445279571/>. – 07.03.2023.
3. Typescript lang [Electronic resource]. – Mode of access: <https://www.typescriptlang.org/en-us/learntocode>. – Date of access: 24.02.2023.
4. Javatpoint Community [Electronic resource]. – Mode of access: <https://www.javatpoint.com/typescript-tutorial>. – Date of access: 30.02.2023.