

Раздел

Качество кода

Навигация по уроку

Babel



Примеры в учебнике

Комментарии

Поделиться



Редактировать на GitHub

 → [Язык программирования JavaScript](#)  
→ [Качество кода](#) 26-го августа 2019

## Полифилы

JavaScript – динамично развивающийся язык программирования. Регулярно появляются предложения о добавлении в JS новых возможностей, они анализируются, и, если предложения одобряются, то описания новых возможностей языка переносятся в черновик <https://tc39.github.io/ecma262/>, а затем публикуются в [спецификации](#).

Разработчики JavaScript-движков сами решают, какие предложения реализовывать в первую очередь. Они могут заранее добавить в браузеры поддержку функций, которые всё ещё находятся в черновике, и отложить разработку функций, которые уже перенесены в спецификацию, потому что они менее интересны разработчикам или более сложные в реализации.

Таким образом, довольно часто реализуется только часть стандарта.

Можно проверить текущее состояние поддержки различных возможностей JavaScript на странице <https://kangax.github.io/compat-table/es6/> (нам ещё предстоит изучить многое из этого списка).

## Babel

Когда мы используем современные возможности JavaScript, некоторые движки могут не поддерживать их. Как было сказано выше, не везде реализованы все функции.

И тут приходит на помощь Babel.

**Babel** – это [транспилер](#). Он переписывает современный JavaScript-код в предыдущий стандарт.

На самом деле, есть две части Babel:

1. Во-первых, транспилер, который переписывает код. Разработчик запускает Babel на своём компьютере. Он переписывает код в старый стандарт. И после этого код отправляется на сайт. Современные сборщики проектов, такие как [webpack](#) или [brunch](#), предоставляют возможность запускать транспилер автоматически после каждого изменения кода, что позволяет экономить время.
2. Во-вторых, полифил.

Новые возможности языка могут включать встроенные функции и синтаксические конструкции. Транспилер переписывает код, преобразовывая новые синтаксические конструкции в старые. Но что касается новых встроенных функций, нам нужно их как-то реализовать. JavaScript является высокодинамичным языком, скрипты могут добавлять/изменять любые функции, чтобы они вели себя в соответствии с современным стандартом.

Термин «полифил» означает, что скрипт «заполняет» пробелы и добавляет современные функции.

Два интересных хранилища полифилов:

- [core.js](#) поддерживает много функций, можно подключать только нужные.
- [polyfill.io](#) – сервис, который автоматически создаёт скрипт с полифилом в зависимости от необходимых функций и браузера пользователя.

Таким образом, чтобы современные функции поддерживались в старых движках, нам надо установить транспилер и добавить полифил.

## Примеры в учебнике

Большинство примеров можно запустить «на месте», как этот:

```
1 alert('Нажмите кнопку "Play" в крайнем правом углу')
```

Раздел

[Качество кода](#)

Навигация по уроку

Babel

Примеры в учебнике

Комментарии

Поделиться



[Редактировать на GitHub](#)



Примеры, в которых используются современные возможности JS, будут работать, если ваш браузер их поддерживает.

Google Chrome обычно поддерживает современные функции, можно запускать новейшие примеры без каких-либо транспилеров, но и другие современные браузеры тоже хорошо работают.

Проводим [курсы по JavaScript и фреймворкам](#). ✕

## Комментарии

перед тем как писать...

