

**Instructor
Notes:**

Add instructor notes
here.



**Instructor
Notes:**

Add instructor notes
here.

Lesson Objectives



At the end of this module you will be able to:

- Explain the importance of ES6
- ES6 compatibility with modern browsers
- Run ES6 code in incompatible browsers using Transpilers like Babel, Traceur and TypeScript



Instructor Notes:

Add instructor notes
here.

Introduction



JavaScript is a superset of
ECMAScript scripting language.

ECMAScript forms the language
base for JavaScript, JScript and
ActionScript.

ES6 gives a vast makeover to
JavaScript by adding new syntaxes
and APIs to write complex
applications and libraries that are
easier to debug and maintain.

ECMAScript 2015

**Instructor
Notes:**

Add instructor notes
here.

Why ECMAScript 6 ?



JavaScript is extremely powerful and flexible, but it is often criticized for having unnecessary redundancy.

JavaScript developers often use abstractions such as CoffeeScript and Typescript, which provide easier syntax, powerful features, and compile to JavaScript.

ES6 was introduced to improve JavaScript and ensure that developers no longer needed to use abstractions or other techniques to write quality code, which leads to a lengthy process.

ECMAScript is a scripting language standardized by Ecma International in the ECMA-262 specification and ISO/IEC 16262. Scripting languages such as JavaScript, JScript and ActionScript are supersets of ECMAScript. Although JavaScript, JScript, and ActionScript have more capabilities than ECMAScript, by defining more objects and methods, the core features of these languages are the same as ECMAScript.

ECMAScript 6 is the sixth version and the seventh edition of the ECMAScript language. In short, it is also called "ES6".

JavaScript was lacking behind some other programming languages when compared to various syntactic forms such as declaring constant variables, declaring block scoped variables, extracting data from arrays, shorter syntax for declaring functions and so on. ES6 adds up a lot of new syntax-based features to JavaScript, which helps the developers to write less and do more. ES6 also prevents programmers from using various hacks for achieving various goals, which have negative performance impact and made code harder to read. In this chapter, we will look at the new syntactic features, introduced by ES6.

Instructor Notes:

Add instructor notes here.

ES6 compatibility table

kangax.github.io/compat-table/es6/														
ECMAScript 5 6 next intl non-standard compatibility table														
Sort by Engine types Show obsolete platforms Show unstable platforms														
v8 SpiderMonkey JavaScriptCore Chakra Carakan KJS Other														
Minor difference (1 point) Small feature (2 points) Medium feature (4 points) Large feature (8 points)														
Compilers/polyfills Desktop browsers														
91% 58% 74% 42% 60% 16% 15% 60% 79% 63% 85% 93% 10% 21% 53% 11%														
Feature name Current browser Traceur Babel + core-js Closure Type-Script + core-js es6-shim IE 11 Edge 12 Edge 13 FF 38 ESR FF 45 ESR CH 50, OP 37 SF 6.1, SF 7 SF 7.1, SF 8 SF 9 KQ 4.14														
Optimisation														
proper tail calls (tail call optimisation)														
Syntax														
default function parameters														
rest parameters														
spread (...) operator														
object literal extensions														
for...of loops														
octal and binary literals														
template literals														
RegExp "v" and "v" flags														
destructuring declarations														
destructuring assignment														
destructuring parameters														

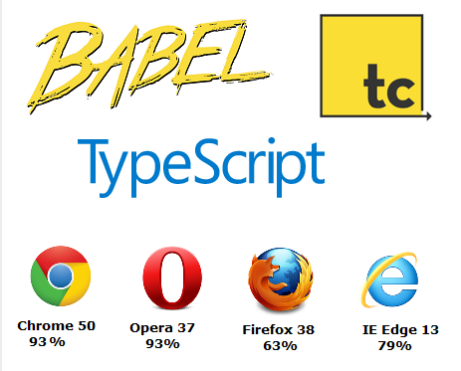
**Instructor
Notes:**

Add instructor notes
here.

Running ECMAScript 6 in an Incompatible Browser



ES6 on development phase can embed the compiler like Babel, TypeScript, Traceur in the webpages which compiles the ES6 to simple browser supportable JavaScript code. On production phase, It is recommended to use node compiler to compile and embed the compiled JS in web pages using the node packages of mentioned compilers.



On production site compiling ES6 to browser supportable JS on every page load can be a resource & time consuming task and can effect the site performance.

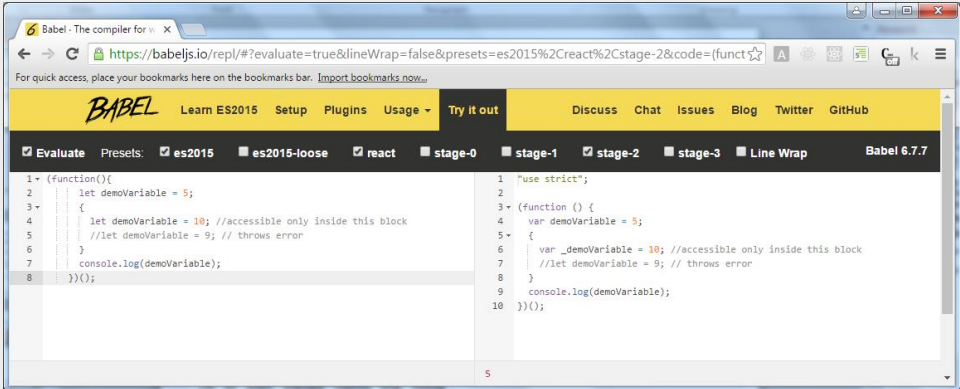
We can use ES6 polyfills but Polyfill is not available for all ES6 feature.

Instructor
Notes:

Add instructor notes
here.

Compiling ES6 using Babel online compiler

Babel Transpiler



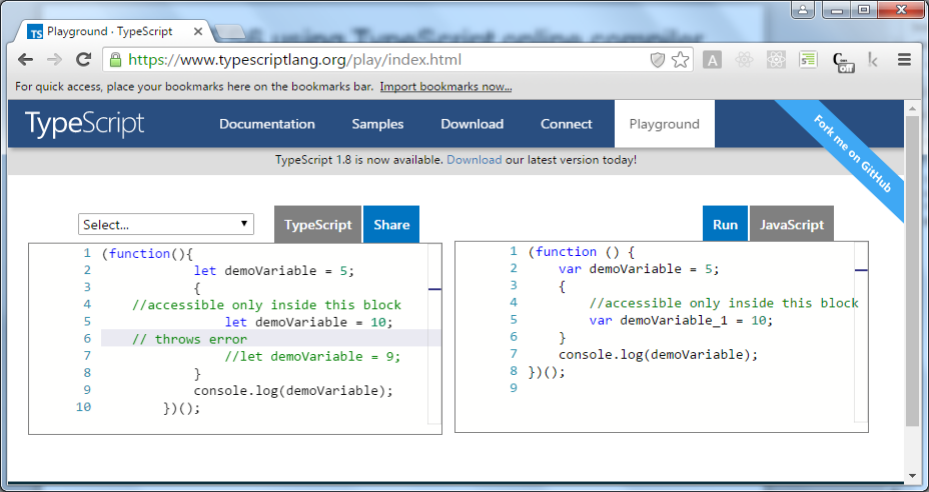
<https://babeljs.io/repl/>

Instructor
Notes:

Add instructor notes
here.

Compiling ES6 using TypeScript online compiler

TypeScript Transpiler



The screenshot shows the TypeScript Playground interface. On the left, the TypeScript source code is displayed:

```
1 (function(){  
2     let demoVariable = 5;  
3     {  
4         //accessible only inside this block  
5         let demoVariable = 10;  
6         // throws error  
7         //let demoVariable = 9;  
8     }  
9     console.log(demoVariable);  
10 })();
```

 On the right, the compiled JavaScript code is shown:

```
1 (function () {  
2     var demoVariable = 5;  
3     {  
4         //accessible only inside this block  
5         var demoVariable_1 = 10;  
6     }  
7     console.log(demoVariable);  
8 })();
```

 The interface includes a 'Run' button and a 'JavaScript' tab. A blue banner on the right says 'Pork me on GitHub'. The URL bar shows 'https://www.typescriptlang.org/play/index.html'.

<https://www.typescriptlang.org/play/index.html>

Instructor
Notes:

Add instructor notes
here.

Compiling ES6 using Traceur online compiler

Traceur Transpiler

Source

```
1 let demoVariable = 5;
2 {
3   //accessible only inside this block
4   let demoVariable = 10;
5 }
6 console.log(demoVariable);
7
```

Options

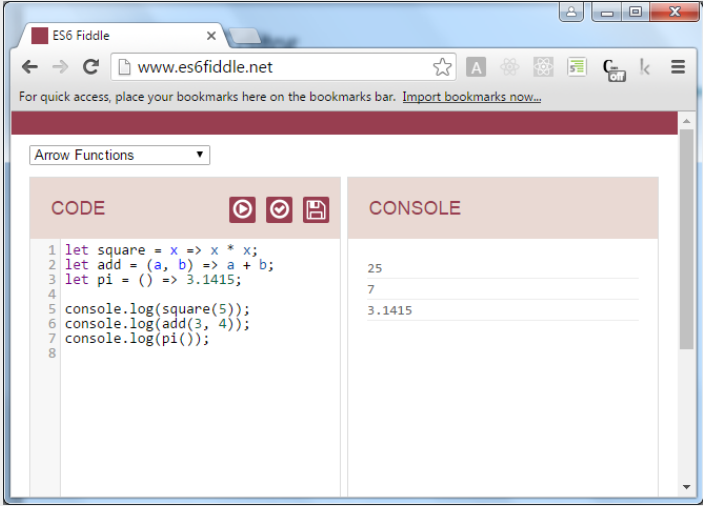
```
1 $traceurRuntime.ModuleStore.getAnonymousModule(function() {
2   "use strict";
3   var demoVariable = 5;
4   {
5     var demoVariable$__0 = 10;
6   }
7   console.log(demoVariable);
8   return {};
9 });
10 /** sourceURL=traceured.js
11
```

[https://google.github.io/traceur-compiler/demo/repl.html#](https://google.github.io/traceur-compiler/demo/repl.html#let%20demoVariable%20)

Instructor
Notes:

Add instructor notes
here.

Online ES6 Editor – es6fiddle.net



<http://www.es6fiddle.net/>

Instructor Notes:

Add instructor notes here.

Summary



➤ECMAScript 2015 is the sixth version and the seventh edition of the ECMAScript language. In short, it is also called "ES6".

➤ES6 adds up a lot of new syntax-based features to JavaScript, which helps the developers to write less and do more.

➤Google Chrome and Opera supports most of the ES6 features.

➤In the production environment, It is recommended to use node compiler like babel, traceur and TypeScript to compile and embed the compiled JS.

