

Developing Javascript with Class

 @mikegeyser

Current JS	ECMAScript 6
Class:	
1. <code>var Vehicle = function () {</code>	<code>class Vehicle {</code>
2. <code> // Constructor</code>	<code> constructor() {</code>
3. <code>};</code>	<code>}</code>
4. <code></code>	<code>}</code>
Inheritance:	
1. <code>var Car = function () {};</code>	<code>class Car extends Vehicle {</code>
2. <code>Car.prototype = new app.Vehicle();</code>	<code>}</code>
Overriding:	
1. <code>var Car = function () {};</code>	<code>class Car extends Vehicle {</code>
2. <code>Car.prototype = new app.Vehicle();</code>	<code> drive() {</code>
3. <code> app.Vehicle.prototype.drive.apply(this, []);</code>	<code> super();</code>
4. <code>}</code>	<code>}</code>
5. <code></code>	<code>}</code>
Namespacing / modules:	
1. <code>(function (app) {</code>	<code>export class Car {}</code>
2. <code> var car = function () {};</code>	
3. <code> app.Car = car;</code>	
4. <code>})(window.app = window.app {});</code>	

Using Google's Traceur Compiler:

(at runtime):

```
1. <script src="traceur.js"></script>
2. <script src="bootstrap.js"></script>
3. <script type="module">
4.   export class Vehicle { }
5. </script>
```

(at 'compile' time using node):

```
$ npm install traceur
$ traceur --out output/vehicles.js --script src/vehicles.js
```

Full source code: <https://github.com/mikegeyser/developing-javascript-with-class>

Google Traceur Compiler: <https://github.com/google/traceur-compiler/>