Javascript Basics

Part 1

Covered key-points

- Survey over JavaScript history
- Language paradigm: dynamic, prototype-based, functional.
- Node.js, package management tools
- Popular build tools
- Popular frontend frameworks and libraries
- React JS and Angular JS. What’s the difference?
- Keeping it all together. React JS boilerplate project.

Survey over JavaScript history

ES1	1997	First edition
ES2	1998	Compliance with ISO/IEC 16262 standard
ES3	1999	Regular expressions, better string handling, new control statements, try/catch…
ES4	-	Abandoned
ES5.1	2011	Strict mode, getters, setters, JSON handling, Reflection…
ES6	2015	Classes, modules, arrow functions, spreads, generators, promises, Map, Set, Weak Map…
ES7	2016	Code isolation, exponentiation operator, Array.prototype.includes(), await ….
ES8		

Language paradigm

Dynamic features

Run-time evaluated and dynamically typed. A type is associated with each value, rather than just with each expression

Prototype-based features

Is almost entirely object-based. An object is an associative array, augmented with a prototype

Functional features

A function is first-class and considered to be an object. As such, a function may have properties and methods, such as .call() and .bind()

Browsers JavaScript compatibility table

ES5.1	99%	98%	100%	100%	98%
ES6	100%	97%	95%	97%	97%
ES7	94%	83%	55%	100%	83%

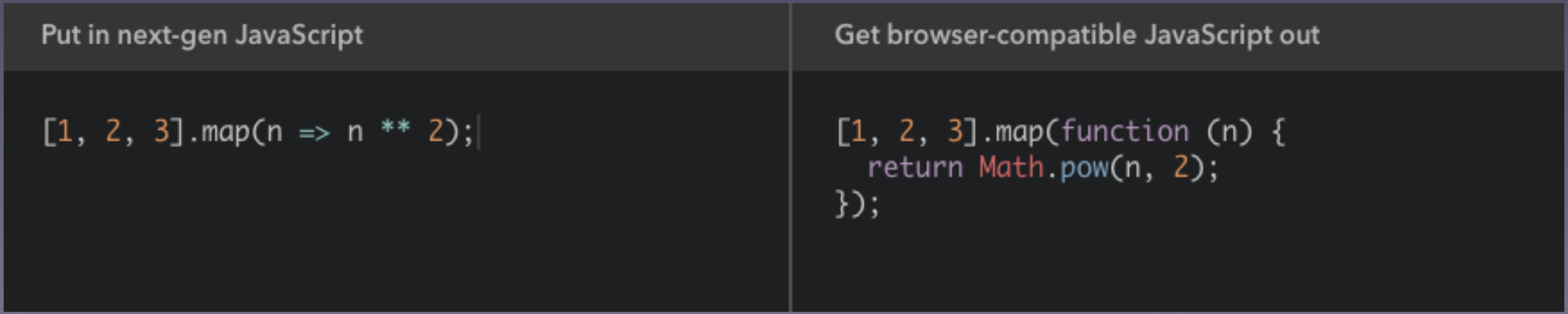
Data source: http://kangax.github.io/compat-table/

Use next generation JavaScript, today

BABEL

Downgrade (transpoiling) version of JavaScript

Babel has support for the latest version of JavaScript through syntax transformers. These plugins allow you to use new syntax, right now without waiting for browser support



Put in next-gen JavaScript

```
[1, 2, 3].map(n => n ** 2);
```

Get browser-compatible JavaScript out

```
[1, 2, 3].map(function (n) {
  return Math.pow(n, 2);
});
```

JavaScript for desktop & server usage

Node.js

V8 Engine

Runtime environment

Open-source JavaScript environment based on Google’s V8. Node.js has an event-driven architecture capable of asynchronous I/O that optimise throughput and scalability in Web applications

Package management tools

NPM

Yarn

Package managing

Node.js package managing command-line tools

Webpack

A module bundler

Webpack takes modules with dependencies and emits static assets representing those modules.