Web Development for Dinosaurs

An Introduction to Modern Web Development

Who Am I?

John Cleaver

Development Team Lead at Factivity, Inc.

Factivity, Inc.

Factivity is a world leader in touch-based Manufacturing Execution Systems.

Agenda

- Web History Timeline

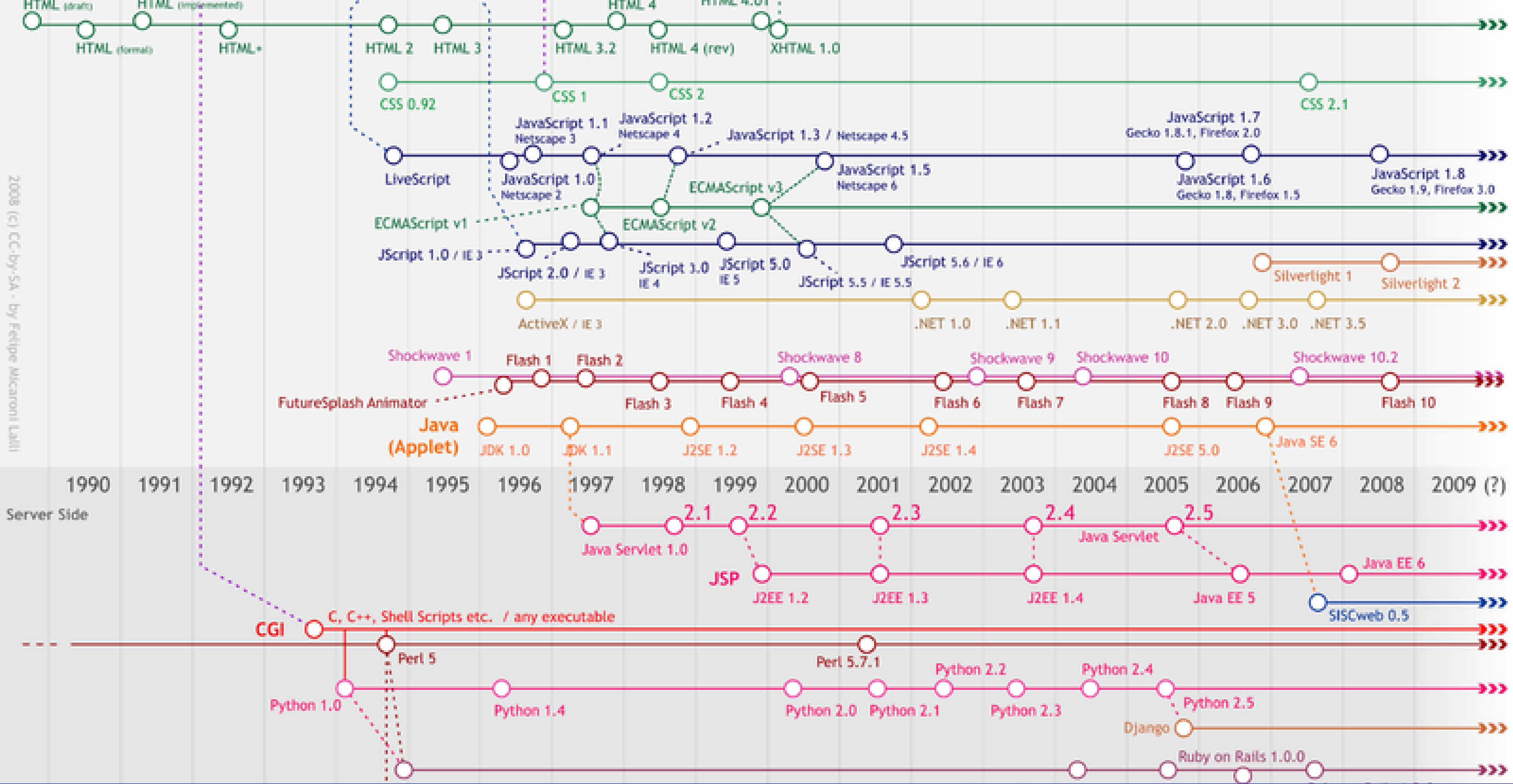
Agenda

- Web History Timeline
- Modern Web Application Tools

Agenda

- Web History Timeline
- Modern Web Application Tools
- Application Demo

A Timeline of Web Development


HTML (draft)
HTML (implemented)
HTML 4
HTML 4.01
HTML (formal)
HTML+
HTML 2
HTML 3
HTML 3.2
HTML 4 (rev)
XHTML 1.0
CSS 0.92
CSS 1
CSS 2
CSS 2.1
JavaScript 1.1 Netscape 3
JavaScript 1.2 Netscape 4
JavaScript 1.3 / Netscape 4.5
JavaScript 1.7 Gecko 1.8.1, Firefox 2.0
LiveScript
JavaScript 1.0 Netscape 2
JavaScript 1.5 Netscape 6
JavaScript 1.6 Gecko 1.8, Firefox 1.5
JavaScript 1.8 Gecko 1.9, Firefox 3.0
ECMAScript v1
ECMAScript v2
ECMAScript v3
JScript 1.0 / IE 3
JScript 2.0 / IE 3
JScript 3.0 IE 4
JScript 5.0 IE 5
JScript 5.5 / IE 5.5
JScript 5.6 / IE 6
Silverlight 1
Silverlight 2
ActiveX / IE 3
.NET 1.0
.NET 1.1
.NET 2.0
.NET 3.0
.NET 3.5
Shockwave 1
Shockwave 8
Shockwave 9
Shockwave 10
Shockwave 10.2
Flash 1
Flash 2
FutureSplash Animator
Flash 3
Flash 4
Flash 5
Flash 6
Flash 7
Flash 8
Flash 9
Flash 10
Java (Applet)
JDK 1.0
JDK 1.1
J2SE 1.2
J2SE 1.3
J2SE 1.4
J2SE 5.0
Java SE 6
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009 (?)
Server Side
2.1
2.2
2.3
2.4
2.5
Java Servlet 1.0
Java Servlet
JSP
J2EE 1.2
J2EE 1.3
J2EE 1.4
Java EE 5
Java EE 6
SISCweb 0.5
CGI
C, C++, Shell Scripts etc. / any executable
Perl 5
Perl 5.7.1
Python 1.0
Python 1.4
Python 2.0
Python 2.1
Python 2.2
Python 2.3
Python 2.4
Python 2.5
Django
Ruby on Rails 1.0.0
2008 (c) CC-by-SA - by Felipe Micaroni Lalli

Era	Page Types	Language	Styling & Layout	Apps
1990-1994	Static	HTML	HTML Attributes	None
1994-2005	Dynamic	PHP, Perl, JSP, ASP, C via CGI	CSS, Some JS	Flash, Shockwave, ActiveX, Java Applets
2005-2010	Dynamic	Ruby / Rails, Python / Django, PHP / Cake, ASP.NET / MVC	CSS, Jquery, AJAX	Flash, Shockwave, Java Applets, Silverlight
2010-2018	Dynamic, WebApps	+ Javascript / Meteor, Express, Angular, React, Vue	CSS Preprocessors, JQuery, AJAX	HTML5 + JS

How did you get in to web development?

Modern Front End Web Developement

Why get in to Web Dev Now?

Why get in to Web Dev Now?

- Javascript is maturing and stabilizing

Why get in to Web Dev Now?

- Javascript is maturing and stabilizing
- Language Specification Changes are slowing

Why get in to Web Dev Now?

- Javascript is maturing and stabilizing
- Language Specification Changes are slowing
- Modern JS looks a lot more like traditional programming

Why get in to Web Dev Now?

- Major Frameworks are > 5 years old and back by larger corporations like MS and Google
    - Even older "less cool" frameworks still exist with updates

Why get in to Web Dev Now?

- Major Frameworks are > 5 years old and back by larger corporations like MS and Google
    - Even older "less cool" frameworks still exist with updates
- HTML and CSS are still the best cross-platform UI toolkit

Why get in to Web Dev Now?

- Major Frameworks are > 5 years old and back by larger corporations like MS and Google
    - Even older "less cool" frameworks still exist with updates
- HTML and CSS are still the best cross-platform UI toolkit
- Write Once, Run Everywhere is actually somewhat feasible now

It is impossible to begin to learn that which one thinks one already knows.

Epictetus, Greek Philosopher

What does a Modern Web App Look Like?

- Components instead of full pages
- Distinct line between front end and backend
- Sometimes no backend at all

Modern JS Toolchain

Editor

- There are many good editors available for front end development
- There are plenty of plugins for older editors and IDEs as well

Feel free to use what you are familiar with or try something new.

Recommendations: MS Visual Studio Code, Jetbrains IntelliJ, Sublime Text 3

HTML5 and CSS3

- More semantic tags
- Local Storage API
- Canvas
- Flexbox layout
- Grid layout
- Smooth, native animations (with GPU acceleration)
- Service Workers
- WebSockets

CSS Preprocessor

- Allows for the use of more programming-like features
- Variables, functions, mixins
- Browser compatibility

Examples: Less (More CSS-like) or Sass (More Ruby-like)

I prefer Sass, but if you know CSS super well, Less is probably a better choice.

Front End Framework

- Optional, but makes your life easier
- Often deals with responsive layout, consistent styling, browser differences, and more
- Usually available in Less or Sass to make customizing and theming easier

Examples: Bootstrap, Foundation, Bulma

"Compiler" - Babel

- Babel is a Javascript transpiler
- It converts modern JS (ES6 and ES2015) into the more widely understood ES5
- It lets you use the latest language features before they are available in browsers
- Allows use of alternate languages that compile to JS like MS's Typescript

Why bother with ES6?

- Module includes
- Arrow Syntax (`=>`) makes some code more readable
    - (Esp. if you are familiar with functional code or lambdas in Java and Python)
- More data structures
- You'll see a lot of example code using it

"Linker" - Webpack

- Webpack packages your source into a real application
- Combines all asset files into 1
- "Minifies"

Package Manager - NPM / Yarn

- The JS world leans more Unix-y, so you'll use the package manager a lot
- Lots of free, open-source packages out there
- Both install packages, manage dependencies, and run utility scripts
- Yarn is a newer one that solves some problems with NPM, but is compatible with NPM

Javascript Framework

- Helps to break down code into re-usable components (think User Controls)
- Helps managing state
- Makes building dyanmic sites easier

Examples: React (Facebook), Angular (Google), Vue (Alibaba)

You don't need to start with everything all at once!

Old School JS

Old School JS

```html
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Example</title>
  <script src="index.js"></script>
</head>
<body>
  <h1>Hello from HTML!</h1>
</body>
</html>
```

Old School JS

```
// index.js
console.log("Hello from JavaScript!");
```

Adding a JS Library

Adding a JS Library

```html
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Example</title>
  <link rel="stylesheet" href="index.css">
  <script src="moment.min.js"></script>
  <script src="index.js"></script>
</head>
<body>
  <h1>Hello from HTML!</h1>
</body>
</html>
```

Adding a JS Library

```
// index.js
console.log("Hello from JavaScript!");
console.log(moment().startOf('day').fromNow());
```

Using NPM

Using NPM

Create a package.json

```
npm init
```

Using NPM

Install packages

```
npm install moment --save
```

Using NPM

Add it to your HTML

```html
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Example</title>
  <script src="node_modules/moment/min/moment.min.js"></script>
  <script src="index.js"></script>
</head>
<body>
  <h1>Hello from HTML!</h1>
</body>
</html>
```

Using NPM

Hand it to a Coworker

```
git clone sweet-project
cd sweet-project
npm install
```

Bundling with Webpack

Bundling with Webpack

- JS did not includeany notion of include / using / import
- JS Modules had to be loaded into a global variable
  - Naming conflicts, global-state problems, etc.
- Nodejs introduced the idea that JS could include other JS files

```
var moment = require('moment');
```

Bundling with Webpack

How do we use that with client-side JS?

- A module bundler can look at include statements and build a JS file that has all the required code
- Webpack is the most popular today
  - Use by React, Vue, etc

Bundling with Webpack

Installing

```
npm install webpack webpack-cli --save-dev
```

Installs both webpack, a command line interface for webpack, and saves those as Development dependencies.

Bundling with Webpack

Webpack it up

```
./node_modules/.bin/webpack index.js --mode=development
```

```
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>JavaScript Example</title>
  <script src="dist/main.js"></script>
</head>
<body>
  <h1>Hello from HTML!</h1>
</body>
</html>
```

Bundling with Webpack

```
// webpack.config.js
module.exports = {
  mode: 'development',
  entry: './index.js',
  output: {
    filename: 'main.js',
    publicPath: 'dist'
  }
};
```

Transpiling

Transpiling

Installing

```
npm install @babel/core @babel/preset-env babel-loader --save-dev
```

Transpiling

```
// webpack.config.js
module.exports = {
...
  },
  module: {
    rules: [
      {
        test: /\.js$/,
        exclude: /node_modules/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }
      }
    ]
  }
};
```

Transpiling

What does this buy us?

Transpiling

What does this buy us?

Newer Features!

Transpiling

What does this buy us?

Newer Features!

Format Strings

```
var name = "Bob", time = "today";
console.log(`Hello ${name}, how are you ${time}?`);
```

Transpiling

What does this buy us?

Newer Features!

Format Strings

```
var name = "Bob", time = "today";
console.log(`Hello ${name}, how are you ${time}?`);
```

Newer Imports

```
import moment from 'moment';
```

NPM Scripts

NPM Scripts

NPM can also be told to be a task runner (think Ant)

```
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "build": "webpack --progress --mode=production",
    "watch": "webpack --progress --watch"
  },
```

```
npm run build
```

NPM Scripts

Built-in Web Server

```
"server": "webpack-dev-server --open"
```

```
npm run server
```

Demo: Vue CLI and Vue Projects

Further Reading

- How I learned to Stop Worrying and Love the Javascript EcoSystem
- Modern Frontend Developer in 2018
- Vuejs Guide
- Bootstrap Docs
- Sass Docs

Wrap Up

- Web Development Timeline

Wrap Up

- Web Development Timeline
- Modern Development Technologies

Wrap Up

- Web Development Timeline
- Modern Development Technologies
- Code Examples

John Cleaver

johnc@factivity.com

Don't forget to fill out the surveys in the conference app!