

# Web Technology Overview

with a focus on JavaScript-based technologies


Lawrence Yao  
lawrence.yao@unswalumni.com

1

# Who am I?

Lawrence Yao

UNSW alumnus (PhD)  
Developer Analyst at YTML Consulting



2

# Tech go in and out of fashion




Here are some "hot" consumer tech from past to present.  
Did they live beyond their hype?




3

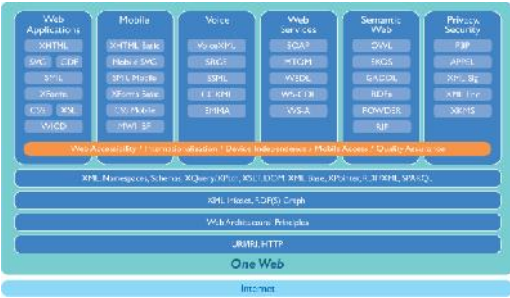
# Continuing education

Amazon Echo






4

# Web Technology Architecture

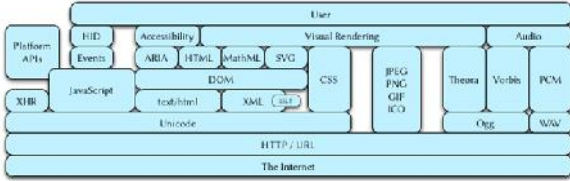


From: <http://www.w3.org/Consortium/techstack-desc.html>




5

# Web Browser Technology Architecture

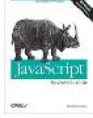


From: <https://hsivonen.fi/web-stack/>



6

## JavaScript



**JavaScript: The Definitive Guide, 6th ed.**  
By: [David Flanagan](#)  
Publisher: O'Reilly Media, Inc.  
Pub. Date: May 3, 2011

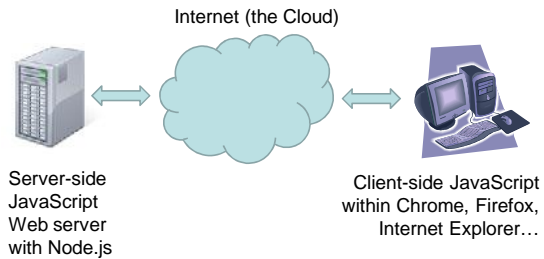
- Highly recommended!!!
- Older editions available through UNSW library
- Learn all about the JavaScript programming language



Slide 7

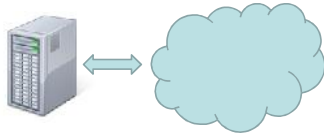
7

## JavaScript on client and server



Slide 8

8



## Server-side JavaScript



Slide 9

9

## <http://nodejs.org/>

- This simple web server written in Node responds with "Hello World" for every request.

```
1 var http = require('http');
2 http.createServer(function(req, res) {
3   res.writeHead(200, {
4     'Content-Type': 'text/plain'
5   });
6   res.end('Hello World\n');
7 }).listen(1337, '127.0.0.1');
8 console.log('Server running at http://127.0.0.1:1337/');
```



Slide 10

10

## Node.js

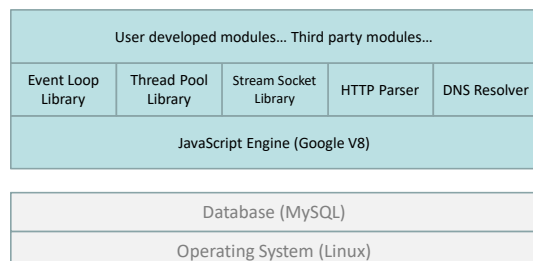
- How does it work? Read these tutorials
  - <http://www.nodebeginner.org/>
    - Highly recommended, but not entirely free
    - First section free, last updated June 2017
  - <http://code.tutsplus.com/tutorials/this-time-youll-learn-node-js-net-19448>
    - Last updated in 2011
- Official API
  - <http://nodejs.org/api/>



Slide 11

11

## Node.js architecture

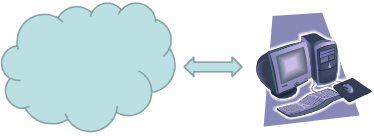


Based on slides from:  
<http://s3.amazonaws.com/four.livejournal/20091117/jsconf.pdf>




Slide 12

12



## Client-side JavaScript

Slide 13




13

## HTML DOM


- To see the Document Object Model (DOM)
  - Open Web browser
  - Go to any Web page
  - Press F12 on keyboard to bring up the “Web Developer Tools” (works for Firefox and IE)
  - (IE10) Don’t forget to refresh DOM
  - Type JavaScript directly in the Console
- DOM can be changed without reloading entire Web page
  - Manually: Web Developer Tools
  - Programmatically: JavaScript

Slide 14



14


## Cascading Style Sheets



**HTML5 and CSS3**  
Elizabeth Castro and Bruce Hyslop  
7th ed. Berkeley, Calif. : Peachpit, c2012  
Checked out at Main Library Level 3 (006.74/5)

- HTML: content; CSS: formatting
- Colour, margin, position, animation, etc...
- Specified in-line or in a separate .css file
- CSS can be changed without reloading
  - Manually: Web Developer Tools
  - Programmatically: JavaScript

Slide 15



15


## CSS selectors

```

1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1 class="special">A Special Class of Heading</h1>
5     <p id="para">A paragraph identified by the string 'para'.</p>
6     <h1 class="special">Another Special Heading</h1>
7     <p id="para2" class="special">ID has to be unique
8       within the DOM.</p>
9   </body>

```

Slide 16



16

## JavaScript

- Guide: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- W3C DOM specification: <http://www.w3.org/DOM/DOMTR>

```

1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h1>A Heading 1</h1>
5     <p id="para">A paragraph of content</p>
6   </body>
7 </html>

```

```

1 var domElem = document.getElementById("para");
2 domElem.setAttribute('style', 'color: red');

```

Slide 17




17

## Web browser

- Provides an environment to run JavaScript
  - The JavaScript code changes the DOM/CSS
  - DOM/CSS changes >> dynamic Web page
- Ties things together
  - Reads HTML and CSS, creates DOM, applies CSS, executes JavaScript, interprets result as the final Web page

Slide 18



18

## Ajax

- Asynchronous JavaScript and XML
- Partially update a Web page in reaction to some event
  - No need to refresh entire Web page
  - Smoother user experience (less disruptive)
- Relies on Web browser functionality
  - Exposed as XMLHttpRequest (XHR) API
- Usually send/receive JSON in background
  - JSON data format: <http://json.org/>



Slide 19

19

## Front-End Frameworks

- A compilation of complementary display elements that works like a theme
  - Focus on responsive design
- Bootstrap: <http://getbootstrap.com/>
  - A popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web
- Foundation: <http://foundation.zurb.com/>
- Demo Grid system: <http://getbootstrap.com/examples/grid/>



Slide 20

20

## jQuery JavaScript library

- W3C DOM API is tedious and verbose
- jQuery is succinct and convenient
- Defines a global variable \$ (or *jQuery*)

```
1 <!DOCTYPE html>
2 <html>
3   <body>
```

```
1 var domElem = document.getElementById("para");
2 domElem.setAttribute('style', 'color: red');
```

```
1 $("#para").css("color", "red");
```



Slide 21

21

## jQuery

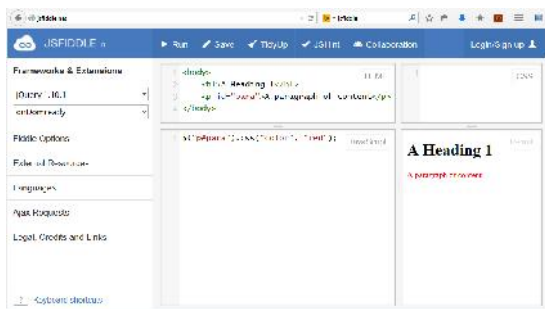
- API Doc: <http://api.jquery.com/>
- Cook book: <http://stackoverflow.com/questions/tagged/jquery>
- Let's play
  - JSFiddle <http://jsfiddle.net/>
  - Don't forget to View Source on the Result frame



Slide 22

22

## JSFiddle



Slide 23

23

## SVG

- Scalable Vector Graphics (SVG)
  - Vector graphics in XML
  - Specification: <http://www.w3.org/TR/SVG/Overview.html>

```
1 <svg width="600" height="400">
2   <g id="layer1">
3     <rect
4       style="fill:#f00000;stroke:#0000ff;stroke-width:12.5"
5       id="rect3774"
6       width="275.0"
7       height="120.0"
8       x="20"
9       y="20" />
10  </g>
11 </svg>
```



24

## SVG in JSFiddle

```

1 <body>
2 <div><rect></div>
3 <svg width="640" height="480">
4 <g id="layer1">
5 <rect
6   id="rect3274"
7   width="275.0"
8   height="120.0"
9   x="33"
10  y="120" />
11 </g>
12 </svg>
13 </body>

```

Slide 25

25

## Web graphics

- Great graphics = HTML + CSS + SVG
  - View in any browser, always sharp, quick to load
  - But static!!
- Great visualisation needs interactivity
- Great visualisations = HTML + CSS + SVG + JavaScript (D3.js)

Slide 26

26

## Data-Driven Documents

- D3.js JavaScript library: <http://d3js.org/>

Data → JavaScript → Great Visualisation

+ HTML + CSS + SVG

Slide 27

27

## D3.js

- How does D3 work? Find out here <https://github.com/mbostock/d3/wiki/Tutorials>
- In particular:
  - Three Little Circles <http://bost.ocks.org/mike/circles/>
  - D3 Tutorials – Scott Murray <http://alignedleft.com/tutorials/d3/>
  - Example gallery <https://github.com/mbostock/d3/wiki/Gallery>

Slide 28

28

## Web Browser Technology Architecture

From: <https://hsivonen.fi/web-stack/>

Slide 29

29

## Putting Everything Together

- People use **Web apps** (Web applications):
  - Client-side: HTML + CSS + SVG + JavaScript + Ajax (D3.js, jQuery, Bootstrap)
  - Server-side: Node.js
- Common styles / behaviour
  - [https://youtu.be/\\_2EDDPUMnk0](https://youtu.be/_2EDDPUMnk0)

Slide 30

30

## Common Web app styles

- Traditional
  - Every click loads another (entire) Web page
- Single-Page Application
  - Clicks loads data (via Ajax) not Web page
  - Use data to refresh areas within Web page
  - Smoother user experience (no page loads)
- Mixed
  - Combination of above

Slide 31



31

## Some hints on starting your project

Slide 32



32

## Node.js tooling

- Node.js on (recent versions of) Linux
  - Install using system package manager
    - Debian/Ubuntu: nodejs, npm
  - Use command line and a text editor
  - Integrated Development Environments
    - Eclipse (search “eclipse node.js”)
    - NetBeans (search “netbeans node.js”)

Slide 33



33

## Node.js tutorials (Linux)

- <http://www.nodeclipse.org/ubuntu/linux/java/nodejs/2015/2015/07/09/Starting-with-Java-and-Node.js-development-on-Ubuntu-Linux.html>
- Netbeans Nodejs Plugin:  
<https://github.com/timboudreau/nb-nodejs>
- Command line:  
<http://blog.modulus.io/absolute-beginners-guide-to-nodejs>

Slide 34



34

## Node.js for Windows users

- Option 1: Switch to Linux?
- Option 2: Use Eclipse or NetBeans?
- Option 3: Use Node.js Tools for Visual Studio
  - Visual Studio 2013 Community Edition
  - Node.js Tools for Visual Studio
  - Node.js for Windows (MSI 64-bit)

Slide 35



35

## Node.js tutorials (Windows)

- <https://channel9.msdn.com/Blogs/Seth-Juarez/Nodejs-Tools-for-Visual-Studio>
  - Includes a demo on deployment to Azure (if you want to use it)

Slide 36



36

## Tutorial on Express framework

- Express via the command line  
<https://codeforgeek.com/2014/10/express-complete-tutorial-part-1/>
- Deploy node.js apps to Heroku (free hosting service – conditions apply)  
<https://devcenter.heroku.com/articles/getting-started-with-nodejs#introduction>

Slide 37



37

## Some UI examples

- Examples:
  - <http://www.nytimes.com/newsgraphics/2013/09/13/fashion-week-editors-picks/>
  - <http://www.nytimes.com/interactive/2014/02/11/sports/sochi-2014-interactive-stories.html>
  - <http://onedrive.live.com/>

Slide 38



38

Thank You

Questions?

Email me: [lawrence.yao@unswalumni.com](mailto:lawrence.yao@unswalumni.com)

Remember  
When learning coding, be brave

**Acknowledgments**  
Lecture notes from Kenny Sabir

Slide 39



39