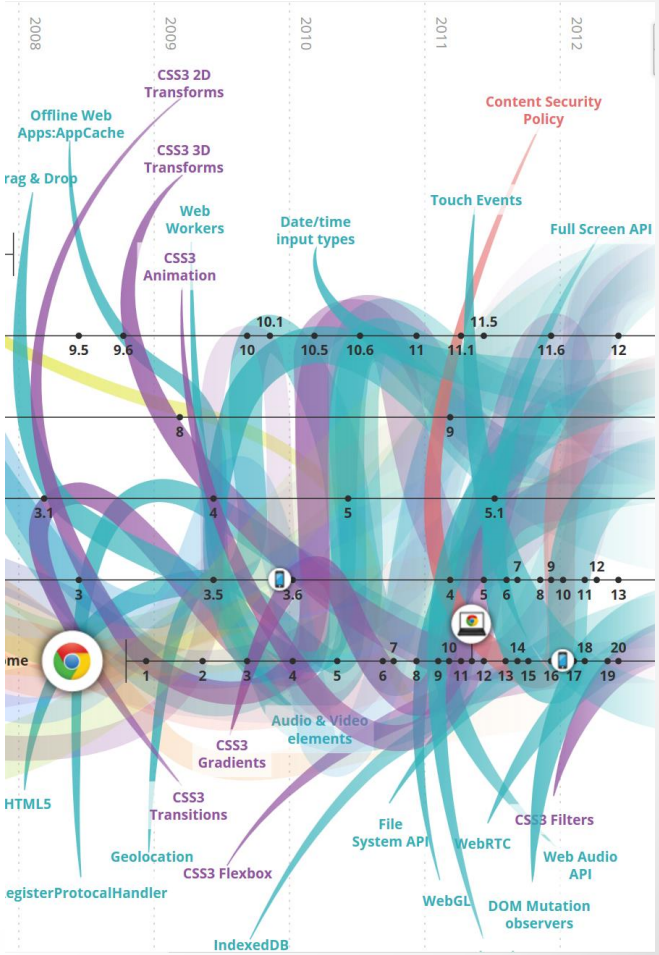
CIT 382
Web Dev II

Week 1

Phil Colbert

Web Applications

- This course focuses on expanding your working knowledge of and experience with creating web applications
- In order to understand what we will be working on, let's first examine The Evolution of the Web



Evolution of the Web

- Why do we care about the evolution of the web?
  - Historical context
  - Technological context
  - Appreciation of what's new and what's old
- On the Evolution of the Web website, hover over or select any of the ribbons to trace the history of the ribbon topic
  - Select the orange ribbon (Browsers & Technologies)
  - Select The Growth of the Internet at the top of the browser

JavaScript

- As you know, JavaScript is the primary programming language for this course
- JavaScript turned 20 years old in 2015
- Prior to this course, you've been exposed to or worked with a number of JavaScript frameworks and libraries, including but not limited to
  - jQuery
  - Node.js
  - Express
- You should review these technologies as needed as we proceed through the course, in addition to any review included during lecture or lab

Meteor

- We will be initially working with Meteor (MeteorJS)
- What is Meteor?
  - Meteor is a full-stack JavaScript platform for developing modern web and mobile applications. Meteor includes a key set of technologies for building connected-client reactive applications, a build tool, and a curated set of packages from the Node.js and general JavaScript community.
    - Meteor allows you to develop in **one language**, JavaScript, in all environments: application server, web browser, and mobile device.
    - Meteor uses **data on the wire**, meaning the server sends data, not HTML, and the client renders it.
    - Meteor **embraces the ecosystem**, bringing the best parts of the extremely active JavaScript community to you in a careful and considered way.
    - Meteor provides **full stack reactivity**, allowing your UI to seamlessly reflect the true state of the world with minimal development effort.

Reactivity

- Meteor includes the concept of reactivity
- Reactivity is exactly what the word implies: listening for and reacting to change
- Reactivity is accomplished with the concept of reactive computations (also known as reactive contexts)
  - A reactive computation is simply a block of code (basically, the inside of a function) that will re-run *whenever a reactive data source inside it changes*.
    - Source: Reactivity Basics: Meteor's Magic Demystified

React vs React Native

- If you search online, you'll encounter articles discussing React Native
  - What are the main differences between ReactJS and Reactive-Native?
- We will be using ReactJS, the JavaScript framework library
  - ReactJS from Facebook
- React-Native is a mobile framework that compiles to native app components, allowing you to build native mobile applications (iOS, Android, and Windows) in JavaScript that allows you to use ReactJS to build your components, and implements ReactJS under the hood
  - React-Native from Facebook

Front-End Developer Handbook

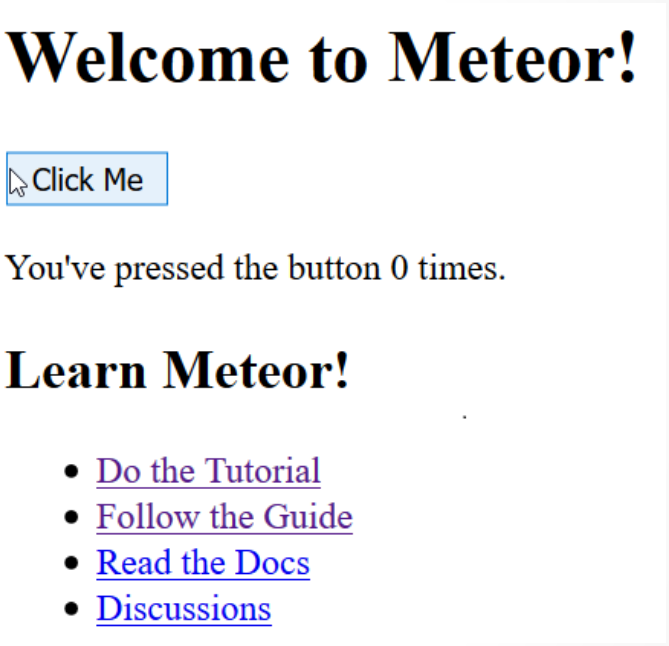
- Before heading into Meteor, we need to establish a technological and vocabulary framework
- Let's head over to the Front-End Developer Handbook web site
  - What Is a Front-End Developer?
  - What Technologies Employed by Front-End Developers
  - Front-End Dev Skills
  - Front-End Developers Develop For…
  - Generalist Myth
  - How Front-End Developers Are Made

Meteor

- Let's begin investigating Meteor
  - Integration list
  - Overview bullets
    - Ship more with less code
    - Build apps for any device
    - Integrate technologies you already use
    - Built with Meteor
  - Meteor Tutorial
    - Blaze
    - Angular
    - React
  - Meteor Guide
  - Meteor Documentation

First Meteor App

- During your Week 1 Lab you will create and run your first Meteor application (app)
- You will also begin the first steps of the official Meteor Tutorial



Full-Stack JavaScript Framework

- Remember that we are creating web applications using the client-server concept
  - The Meteor server component is a Node application that servers as both a web server and an API server for your application
  - The Meteor client components are HTML, CSS and JavaScript files that work as entry points for consumers
- Meteor is labeled a full-stack JavaScript framework, meaning that in addition to client-side and server-side code, Meteor includes common code that runs on both client and server

Initial Folders and Files

- Initially, Meteor separates your application components using folders
  - client: contains client files
  - server: contains server files
- Each folder contains the relevant files

```
client/main.js          # a JavaScript entry point loaded on the client
client/main.html        # an HTML file that defines view templates
client/main.css         # a CSS file to define your app's styles
server/main.js          # a JavaScript entry point loaded on the server
package.json            # a control file for installing NPM packages
.meteor                 # internal Meteor files
.gitignore              # a control file for git
```

Additional Folders and Files

- In addition to client and server folders and files, all other application code will be placed in a folder called imports
- Code contained within the imports folder must be explicitly included in your other code files
- Meteor limits code in the server folder to only run on the server, and all code in the client folder is only available to clients
- The client/main.js and server/main.js files serve as entry points for configuration and startup code

Example Directory Layout

```
imports/
  startup/
    client/
      index.js                # import client startup through a single index entry point
      routes.js               # set up all routes in the app
      useraccounts-configuration.js # configure login templates
    server/
      fixtures.js             # fill the DB with example data on startup
      index.js                # import server startup through a single index entry point

  api/
    lists/                    # a unit of domain logic
      server/
        publications.js       # all list-related publications
        publications.tests.js # tests for the list publications
      lists.js                # definition of the Lists collection
      lists.tests.js          # tests for the behavior of that collection
      methods.js              # methods related to lists
      methods.tests.js        # tests for those methods

  ui/
    components/               # all reusable components in the application
                              # can be split by domain if there are many
    layouts/                  # wrapper components for behaviour and visuals
    pages/                    # entry points for rendering used by the router

client/
  main.js                     # client entry point, imports all client code

server/
  main.js                     # server entry point, imports all server code
```

Source: [Meteor Application Structure]

Import and Export

- Although we typically refer to JavaScript as the programming language for this course, factually JavaScript as a language standard is known as ECMAScript, often abbreviated as ES
- Meteor supports the ES2015 standard, which includes a number of language enhancements, including the import and export statements
  - import: import functions, objects, or primitives that have been exported from an external module, another script, etc.
  - export: export functions, objects or primitives from a given file or module
- We will see import and export in action as we work through the Meteor tutorials

ES2015 – ES6

- ES2015, also known as ES6, includes a number of other enhancements used by Meteor
  - arrows (arrow functions): function shorthand
  - let and const: block-scoping constructs
- For a nice summary, refer the following article
  - ES6 features

First-App HTML Source

- Examine the web page source code sent to our browser when we load the Meteor first-app

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" class="__meteor-css__" href="/merged-stylesheets.css?hash=1a165863f95fc2ba4d6af1034a1380b15db31f19">
<title>first-app</title>

</head>
<body>



<script type="text/javascript">__meteor_runtime_config__ = JSON.parse(decodeURIComponent("%7B%22meteorRelease%22%3A%22METEOR%401.4.2.3%22%2C%22

  <script type="text/javascript" src="/packages/underscore.js?hash=cde485f60699ff9aced3305f70189e39c665183c"></script>
  <script type="text/javascript" src="/packages/meteor.js?hash=e3f53db3be730057fed1a5f709ecd5fc7cae1229"></script>
  <script type="text/javascript" src="/packages/meteor-base.js?hash=a4d07a6b394e56bbe6ccc773c95e7cdb3434960d"></script>
  <script type="text/javascript" src="/packages/mobile-experience.js?hash=8ded3e69a3e367f321ab9a2b52e3ecdd2661a365"></script>
  <script type="text/javascript" src="/packages/modules-runtime.js?hash=637cb12b6fd3acf3ffbfcf0f2ee1cf3988a7f1fe"></script>
  <script type="text/javascript" src="/packages/modules.js?hash=954e49e6fa669579f04cdf1f3329b6e8b050f0e6"></script>
  <script type="text/javascript" src="/packages/es5-shim.js?hash=adc3c6270d5697523fe2a72e73428390b7eba83a"></script>
  <script type="text/javascript" src="/packages/promise.js?hash=33f3b940f94952cb3a44d500099467511da0b575"></script>
  <script type="text/javascript" src="/packages/ecmascript-runtime.js?hash=d0c1e87c070810a617a5853c970ab05284650255"></script>
  <script type="text/javascript" src="/packages/babel-compiler.js?hash=a9546d4e245cfe40b406e08d40bf106241f01683"></script>
  <script type="text/javascript" src="/packages/ecmascript.js?hash=370a8752194bcf73be7fffa3635715d0fbf7853d"></script>
  <script type="text/javascript" src="/packages/base64.js?hash=0053489bb30bb5c0e3545df151f83e41150344b0"></script>
  <script type="text/javascript" src="/packages/ejson.js?hash=0f17ced99d522d48cd8f8b2139167fd06babd969"></script>
  <script type="text/javascript" src="/packages/id-map.js?hash=c7aea8dfa2bf46ff2ae0aa6c6cf09e36abc61d07"></script>
  <script type="text/javascript" src="/packages/ordered-dict.js?hash=bacdd1852075630a01f7de783e5e8e8aa8541cdc"></script>
  <script type="text/javascript" src="/packages/tracker.js?hash=9f8a0cec09c662aad5a5e224447b2d4e88d011ef"></script>
  <script type="text/javascript" src="/packages/babel-runtime.js?hash=cd7adc00903a7005bb7fbad73ca0c3986be427d1"></script>
  <script type="text/javascript" src="/packages/random.js?hash=31dadb9d3427506dd30dc50f5c898837e69e739b"></script>
  <script type="text/javascript" src="/packages/mongo-id.js?hash=345d169d517353f8146292b4abd24061721f8b26"></script>
  <script type="text/javascript" src="/packages/diff-sequence.js?hash=15014d7b1e11c05111a386992e684ab1d3cc4158"></script>
  <script type="text/javascript" src="/packages/geojson-utils.js?hash=b204c7d4caf119e6883522fb87c6cce060724bf0"></script>
  <script type="text/javascript" src="/packages/minimongo.js?hash=66cc6ab213289f154f49d61566dba8ff9dfc33b2"></script>
  <script type="text/javascript" src="/packages/check.js?hash=63d7478b74cadc04d378bc2266ea8bd1bf6849d8"></script>
  <script type="text/javascript" src="/packages/retry.js?hash=1e409617b538ff3e2b0238b15e45b3380c51a224"></script>
  <script type="text/javascript" src="/packages/ddp-common.js?hash=d42359bcace6c66ac90e2782193494253ee68155"></script>
  <script type="text/javascript" src="/packages/reload.js?hash=628b069673bffbc7390ba84ece8809c8c88c2eed"></script>
  <script type="text/javascript" src="/packages/ddp-client.js?hash=bc32a166cd269e06a394f9418e0024d805bab379"></script>
  <script type="text/javascript" src="/packages/ddp.js?hash=25dc3f428447c81620c91c4245dbc6e4f7d32fb7"></script>
```

First-App HTML Source

```
60    <script type="text/javascript" src="/packages/spacebars.js?hash=ebf9381e7fc625d41acb0df14995b7614360858a"></script>
61    <script type="text/javascript" src="/packages/templating-compiler.js?hash=a71883cdec50e95ca135291415990753ed6d57fc"></script>
62    <script type="text/javascript" src="/packages/templating-runtime.js?hash=c18de19afda6e9f0db7faf3d4382a4c953cabe18"></script>
63    <script type="text/javascript" src="/packages/templating.js?hash=c2cf38de06efb47f67affb2dff9320e5eef33893"></script>
64    <script type="text/javascript" src="/packages/launch-screen.js?hash=2f56943306c7e900ed9f4d894b87f534ebffeaeb"></script>
65    <script type="text/javascript" src="/packages/ui.js?hash=039c55a98376abd03d9d8cd4100895861b897643"></script>
66    <script type="text/javascript" src="/packages/autoupdate.js?hash=1fd9cf3472adaa6887170d88ab5ea1ddabf695fa"></script>
67    <script type="text/javascript" src="/packages/global-imports.js?hash=297fad17a32be63bc48685db4bfdc6a296e2e8f9"></script>
68    <script type="text/javascript" src="/app/app.js?hash=decb3bad3ca03d88660961ae4f296bfdef4a22a1"></script>
69
70
71  </body>
72  </html>
```

Template Compilation

- Where is the source HTML?
- When building our app, the meteor templating system parses all of the client HTML files and generates a JavaScript file grouping the contents based on the HTML head tag, body tag and template tags
- When accessing our app entry web page, in addition to all of the related packages being loaded, the last entry is our actual template

```
<script type="text/javascript" src="/app/app.js?hash=decb3bad3ca03d88660961ae4f296bfdef4a22a1"></script>
```

Templates

- Meteor defaults to the use the Blaze templating system for the user interface (UI), or View
- Although templates work, the concept of a template is often used to view web page content as a whole, rather than as components or subcomponents, a key feature of React
- We will be learning React rather than Blaze