Ladies Who Code talk

ES6 overview (collection of examples: http://codepen.io/collection/nNqpxp/)

- Holy trinity of front-end: HTML, CSS and Javascript
  - We've had new capabilities with HTML5 and CSS3, about time we had something new for Javascript
  - HTML5 = 2014 release, CSS3 = 2012 release
  - ES5 = 2009 release!
- ES6 release due month
  - very little of the specification should change now
  - designed with maximum compatibility in mind
  - pretty good browser support already but check http://kangax.github.io/compat-table/es6 for full list
  - Transpilers (e.g. Traceur) for new syntax compatibility
  - Shims for objects and functions compatibility
  - can enable in Chrome by going to chrome://flags and "enable experimental javascript features"
  - BEST OF ALL: existing code will still work
- let variables:
  - in the beginning, variables were declared using var and they were global
  - variable declarations (not initialisations) were hoisted to the top of the scope - as the scope was global, this meant they were moved to the top of the script/function
  - this meant that variables could be used before they had been declared, causing all sorts of fun
  - let variables are basically vars, but block-scoped
- const variables:
  - also block-scoped
  - but cannot be reinitialised
  - fails with a runtime error if program tries to change a constant's value

- variable improvements:
  - variables can be declared closed to where they're needed, instead of at the top
  - no need for nested functions to control variable scope
  - constants tell programmers and runtime about what the variable is meant to do
  - both let and const will make code easier to read and allow for small optimisations
- for-of loop:
  - old method: declare new variable to count index, while the index is less than the length of the array, increment index
  - for-of loop: simply, for every item in this array
  - works for data and strings
  - can get rid of the array variable completely
  - better than forEach() because works with break, continue and return
  - better than for-in because loops over data (e.g. array values) and strings
  - more concise, also works with strings, sets and map objects
  - as seen in C++, Java, C# and Python
- sets:
  - ordered list of values without duplicates
  - usually for comparison rather than access (e.g. does the set contain x?)
  - as seen in Java, Ruby and Python
  - adding to the set is chainable,
  - can also delete individuals or clear all
- maps:
  - again, familiar to those using other languages
  - used to use regular objects as maps but means all keys are stringified
  - ES6 means key and value can be any type
  - store and retrieve data using getters and setters (also chainable)
  - as with sets, has(), clear() and delete() can be used
  - determine number of items in map with size