3/9: Tables, responsive design

Intro to CSS, continued

Tables

- From **section 17** ⬈ **(http://www.w3.org/TR/CSS21/tables.html)** of the spec
- Some things are specified with properties already covered
  - Horizontal alignment in a column: text-align
  - Vertical alignment in a row: vertical-align
- Borders
  - border-collapse
    - separate
      - Default
      - Different borders for different cells actually makes sense
      - Special handling for **empty cells** ⬈ **(http://www.w3.org/TR/CSS21/tables.html#empty-cells)** (if you ask for it)
    - collapse
      - Better for common borders between cells
      - Trying to have different borders around different cells leads to a confusing nightmare
      - Allows borders around column and row groups

Spanning rows and columns

- Not a part of CSS
- It's actually structural, part of XHTML
- `rowspan` and `colspan` attributes of `<th>` and `<td>` elements

Responsive Web design

- Web-browsing platforms are increasingly diverse
  - Originally just computers (or terminals)
  - Now laptops, smart phones, not-so-smart phones, tablets, laptops, …
  - Different interface styles (keyboard+mouse vs. touchscreen vs. voice input vs. ???)
- Prohibitive to design separately for each platform
- Result: responsive Web design
  - Term coined by **Ethan Marcotte** ⬈ **(http://alistapart.com/article/responsive-web-design)**
  - Idea is that Web page responds differently to different platforms
  - Usually includes:
    - Resizable content (especially images)
    - Flexible layout (adapts to different screen sizes)

- Media queries (adapts to different sizes, orientations, and resolutions)
        - Javascript to handle different interaction styles, pick up the finer points of display
    - We'll deal with the last part later
- **Examples** ⤷ **(https://www.invisionapp.com/inside-design/examples-responsive-web-design/)** , collected by Jes Kirkwood
- **Quick article/tutorial with good links** ⤷ **(http://coding.smashingmagazine.com/2011/01/12/guidelines-for-responsive-web-design/)** , by Vitaly Freidman

CSS and responsive Web design

Resizable elements

- First step: don't give things fixed sizes!
- Specify dimensions in something **other** than pixels
    - Exception: 1px
        - Still the thinnest possible border/margin/padding
    - Specify in %
        - Result: scales automatically!
    - Specify in font units for font-dependent things (example: indents)
        - ems, exs
            - 1 em is (in theory) the width of an "M" in the current font—
            - 1 ex is (in theory) the height of an "x" in the current font
            - Box, 1 em wide by 1 ex high: ⧠
- Tradeoffs:
    - Some cost in speed to resize things (especially images)
    - Scaling things can give strange-looking results
        - Especially making things larger

Scaling images

- Basic approaches:
    1. Do direct scaling on the image in CSS
        - We've seen `width` and `height` properties
        - `max-width` property keeps images from scaling larger
        - `min-width` property keeps images from scaling down too small
    2. Let the image crop
        - Good for images where the size of the individual elements is important
        - Background images do this
    3. Serve different images at different sizes/screen resolutions
        - Use media queries to figure out which to serve
- Often combine approaches

Media queries

- **[W3C spec](http://www.w3.org/TR/css3-mediaqueries/)** ⤷ **(http://www.w3.org/TR/css3-mediaqueries/)**
- Three places they show up:
  - `<link>` tags (`<link rel="stylesheet"` **`media="screen"`** `href="screen.css"/>`)
  - Stylesheets and `<style>` elements
  - Style data on individual elements
- Media types
  - screen, print, projection, speech
  - Define *types* of output devices
- Media queries
  - Allow querying *specific properties* of output devices
  - List is in the spec
  - Query returns a Boolean value
    - If true, apply the style
    - If false, skip it
  - Usual logical operators
    - and and not are simply spelled out
    - or is accomplished with a comma
    - No grouping facility
    - Parentheses are already used to delimit pieces of the query

Responsive layout