Responsive Web Design

Mendel Rosenblum
Web App Challenges: Screen real estate

- Do we need to build N versions of each web application?
Responsive Web Design

- **Content is like water!**
  - The web app should flow into and fill whatever device you have.

- **Possible with CSS extensions:**
  - Add grid layout system with relative (e.g. 50%) rather than absolute (e.g. 50pt) measures
    - Specify element packing into columns and rows
  - Add `@media` rules based on screen sizes
    - Switch layout based on screen size
  - Made images support relative sizes
    - Autoscale image and videos to fit in screen region
      ```
      img { width: 100%; height: auto; }
      video { width: 100%; height: auto; }
      ```
Example of Responsive Web Layout

<table>
<thead>
<tr>
<th>Menu #1 - 25%</th>
<th>Menu #2 - 25%</th>
<th>Menu #3 - 25%</th>
<th>Menu #4 - 25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nav #1 - 25%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nav #2 - 25%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nav #3 - 25%</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

View component - 75%

Footer - 100%
CSS Breakpoints

CSS Rules:

```css
@media only screen and (min-width: 768px) {
    /* tablets and desktop layout */
}

@media only screen and (max-width: 767px) {
    /* phones */
}

@media only screen and (max-width: 767px) and (orientation: portrait) {
    /* portrait phones */
}
```
Responsive implementation

- Build components to operate at different screen sizes and densities
  - Use relative rather than absolute
  - Specify sizes in device independent units

- Use CSS breakpoints to control layout and functionality
  - Layout alternatives
  - App functionality conditional on available screen real estate

- Mobile first popular
  - Expand a good mobile design to use more real estate