Programming
web design

MICHAEL BERNSTEIN
CS 247
Today: how do I make it?

- All designers need a medium. Napkin sketches aren’t enough.
- This week: core concepts for implementing designs on the web
  - Grids
  - Responsive design
  - UI components
  - Iconography and fonts
Technologies I assume you know

- HTML
- Javascript
- CSS
- No back-end experience assumed

- If you are taking or have taken CS 142, you’ll be fine.
Grids

I know there's a 'grits' pun in here somewhere.
Architecture of a grid

- Grid columns
Architecture of a grid

- Grid columns
- Rows indicate where items get vertically aligned
Architecture of a grid

- Grid columns
- Rows indicate where items get vertically aligned
- Inside rows, the divs span multiple columns
Architecture of a grid

- Nest grids inside of grids in order to set up layouts within a div
Implementing grids: Bootstrap

- Twitter’s front-end web development framework
- Makes sane layout and styles easy to write
- Comes with predefined styles that you apply using their classes
Add Bootstrap to your HTML

1. CSS in `<head>`, before your own CSS

```html
<head>
    <title>HCI Project Portfolio</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Bootstrap -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <link rel="stylesheet" href="css/bootstrap-theme.min.css">
    <link rel="stylesheet" href="css/introHCI.css">
</head>
```

2. JS just above `</body>` at the end of the document

```html
<script src="https://code.jquery.com/jquery.js"></script>
<script src="js/bootstrap.min.js"></script>
```
Add Bootstrap styles

- Put the entire `body` inside a `<div class="container">`
- Containers add margins
Bootstrap Grid

- Bootstrap gives you 12 columns

<table>
<thead>
<tr>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
<th>.col-md-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>.col-md-8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.col-md-4</td>
<td>.col-md-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>.col-md-6</td>
<td>.col-md-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Bootstrap Grid

- Rows go in the `container`, and each div is tagged with `col-md-X` where X is the # columns to span out of 12
Four images to a row, spaced evenly

Example

```html
<div class="container">
  <div class="row">
    <div class="col-md-3">
      <img src="/hci-square.png" class="img img-responsive">
    </div>
    <div class="col-md-3">
      <img src="/hci-square.png" class="img img-responsive">
    </div>
    <div class="col-md-3">
      <img src="/hci-square.png" class="img img-responsive">
    </div>
    <div class="col-md-3">
      <img src="/hci-square.png" class="img img-responsive">
    </div>
  </div>
</div>
```
Example 2

Four images to a row, spaced at 2col-2col-4col-4col

```html
<div class="container">
  <div class="row">
    <div class="col-md-2">
      <img src="./hci-square.png" class="img img-responsive">
    </div>
    <div class="col-md-2">
      <img src="./hci-square.png" class="img img-responsive">
    </div>
    <div class="col-md-4">
      <img src="./hci-square.png" class="img img-responsive">
    </div>
    <div class="col-md-4">
      <img src="./hci-square.png" class="img img-responsive">
    </div>
  </div>
</div>
```
Example 2

Four images to a row, spaced
Nesting grids

- Nest by creating a new `<div class="row">` inside of a div
- You now have a new set of 12 columns within that div

```html
<div class="container">
  <div class="row">
    <div class="col-md-6">
      <p>Hello, world!</p>
    </div>
    <div class="col-md-3">
      <div class="row">
        <div class="col-md-8">
          <p>Internal grid</p>
        </div>
        <div class="col-md-4">
          <p>Internal grid 2</p>
        </div>
      </div>
    </div>
    <div class="col-md-3">
      <p>Goodbye, world!</p>
    </div>
  </div>
</div>
```
Enter the grid

Get the files at http://hci.st/247grid
Hamlet: Tomorrow and tomorrow and tomorrow creeps in this petty pace from day to day. To the last syllable of recorded time.
Hamlet

Tomorrow and tomorrow...

Hamlet

Tomorrow and tomorrow...
Responsive design

Totally responsive to your needs, just like Stanford Dining.
One HTML page, multiple views

- How should my page look on…
  - A laptop with a wide window?
  - A laptop?
  - A tablet?
  - A phone?
Intuition: use the grid

- In responsive design, we reflow the grid at different viewport widths
CSS media queries

- CSS rules that ask how wide the browser viewport is

/* Extra small devices (phones, less than 768px) */
@media (max-width: 768px) { #logo { color: yellow; } }

/* Small devices (tablets, 768px and up) */
@media (min-width: 768px) { #logo { color: red; } }

/* Medium devices (desktops, 992px and up) */
@media (min-width: 992px) { #logo { color: blue; } }

/* Large devices (large desktops, 1200px and up) */
@media (min-width: 1200px) { #logo { color: green; } }
Responsiveness in Bootstrap

- Bootstrap has wrapped all these media queries into CSS classes for grid layouts
- We’ve been using the default, \texttt{md}
- **Examples**: \texttt{col-md-4, col-lg-6, col-sm-3, col-xs-2}
  - \texttt{xs} extra small (mobile)
  - \texttt{sm} small (tablet)
  - \texttt{md} medium (desktops)
  - \texttt{lg} large (wide desktops)
Responsiveness in Bootstrap

- Bootstrap automatically applies the appropriate grid class when the viewport size changes.
- By default, all `md` items are 12 columns (full width) at `xs` and `sm` sizes.
- To create different behavior, we apply multiple classes to a div element:
  ```html
  <div class="col-md-3 col-xs-6">
  ```
Example

```html
<div class="row">
  <div class="col-md-3 col-xs-6">
    <img src="./hci-square.png" class="img img-responsive">
  </div>
  <div class="col-md-3 col-xs-6">
    <img src="./hci-square.png" class="img img-responsive">
  </div>
  <div class="col-md-3 col-xs-6">
    <img src="./hci-square.png" class="img img-responsive">
  </div>
  <div class="col-md-3 col-xs-6">
    <img src="./hci-square.png" class="img img-responsive">
  </div>
</div>
```

Four to a row at desktop (md) width, and two to a row at mobile (xs) width.
Example at xs screen width

class="col-md-3"

HCl

class="col-md-3 col-xs-6"

HCl  HCl
HCl  HCl

col-md-3 col-xs-6  col-md-3 col-xs-6

col-md-3 col-xs-6  col-md-3 col-xs-6

Try it

Edit your previous page so that the three columns of text near the top lay out with one two-column row and one full-width row when you're on a mobile phone.

To test, drag your browser window and make it thin.
<div class="row">
    <div class="col-md-4 col-xs-6">
        <p>col-md-4 col-md-4 col-md-4 ...</p>
    </div>
    <div class="col-md-4 col-xs-6">
        <p>col-md-4 col-md-4 col-md-4 ...</p>
    </div>
    <div class="col-md-4 col-xs-12">
        <p>col-md-4 col-md-4 col-md-4 ...</p>
    </div>
</div>
All that said...

- Responsive design is part of the material in this class

...but it is not required for P2.