

ECS 189H WEB PROGRAMMING

4/10

Assn2, Layout

- We'll hone our CSS skills by making a reasonably professional-looking Web page
- No functionality, buttons will not work, etc.
- Should be responsive, ie. stuff moves when window size changes, pieces don't get cut off!
- We'll use flexbox and media queries, which we will cover on Wds.
- Code to Danielle's specs.
- Due Friday night.

Designing for Web

With Dani Macedo

How cool is this?!

Who did it better?

<http://www.thecapitol.pn/>

<https://smartsite.ucdavis.edu/portal>

<https://www.ucdavis.edu/>

What's important!

- Design with audience in mind
 - Statistics tell that users often leave a website due to bad design
- Potential factors of bad design:
 - Colors (emotions, disabilities, etc.)
 - Bad structure
 - Unclear navigation
- User testing

Good practice for web (code)

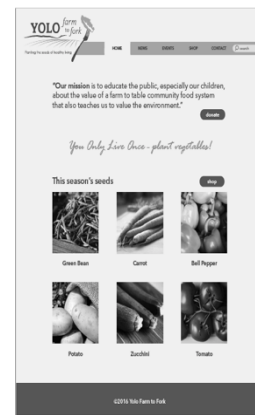
- 1 CSS sheet for all pages
- Make your code legible and understandable
 - Semantic markup
 - Appropriate indentation
 - Comments
- Hand coding instead of copy + paste
- If you're stuck on ONE thing for 15+ minutes, return to it later!

Good practice for web (design)

- "Above the fold"
- Be mindful of the screen size you use
- Hierarchy
 - What will the user want to find first?
- "Form follows function"
 - Good design should implement and amplify the functionality of the site
 - Bad design can distract from it

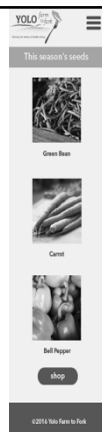
Your Goal

Desktop size



Mobile size

- Keep the six are still there... display 1 column of 6 rows, or 3 rows of 2 columns
- Hamburger menu



Flexbox

- We know page elements are block (eg. div) or inline (eg. text). This is the "display" property of the element (can always change in CSS).
- Third type of display:
display: flex;
- Inherited! So children of flex boxes should be flex boxes. But many elements (eg. div) are explicitly block, so they need to be made explicitly flex.

Few details in the HTML head

```
<meta name="viewport" content="width=device-width">
```

- Tell browser content should fill viewport. You'll notice problems in the Chrome emulator if this is missing.

```
<link rel="stylesheet" type="text/css" href="reset.css">
```

- Reset css parameters to very simple defaults so no browser-specific surprises.

Flex boxes can stack sideways

- Decide whether to keep the default vertical stacking or start stacking left-to-right (or even right-to-left).

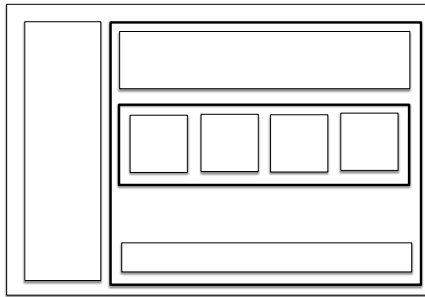
```
flex-direction: row; /* default for flexboxes! */
```

or

```
flex-direction: column;
```

- Defines the main axis; the perpendicular direction becomes the cross axis.

Switching creates interest, focus



Box width and height

- Can be auto (sized just big enough in the main direction to contain their contents, fill space in cross direction)
- Can be fixed (using the width and/or height properties)
- Can stretch to fill up empty space in parent box (this is the "flex" part!)

```
flex-grow: 1;
```

Spacing things using growth

- We can give different boxes different growth factors to allocate more free space to some than to others.
- A box with factor 2 gets two shares of the extra space, a box with factor 3 gets three shares, etc.