Today's goals

- Getting help to those who need it
- Questions raised by the assignment, including getting the right content into each view, colors, and floating text around an image.
- Begin formal introduction to Javascript.

Opportunities for help

- It is easy for us to spend more time with people who know the material than those that don’t.
- Labs sections will be focused on reviewing and enriching material we have covered.
- When possible, ask elementary questions on Piazza — other students are eager to help!
- I’ll do some quick polls on material to review more.
- But— We don’t do labs or drop-ins on Thursday.

Adding and deleting content

- The “read more” and “read less” buttons add and remove content.
- The media queries do too, e.g. the mobile view does not include this image, the tablet one does.

Media query example

```css
p.button {
  background-color: blue;
}

@media (max-width: 500px) {
  p.button {
    background-color: pink;
  }
}
```

Idea: change class names

- Don’t change display property to none in Javascript. Instead, change the class name and let css handle the display property.
- To swap a class name of an element in Javascript:
  ```javascript
  let element = document.getElementById("sketch");
  element.classList.remove("lessContent");
  element.classList.add("moreContent");
  ```
- An element can have lots of class names; so they go in this classlist.
The css

- The class names tell you if you are in the view with more content or less content.
  /* phone properties */
  /* use both id and class */
  display: none;
}
#sketch.moreContent {
  display: block;
}

Computer Color

- Can specify colors like black, white, red, paleturquoise.
- Computer colors are defined as mixtures of red, green and blue.
- Think of this as three spotlights; the more lights that hit a point, the brighter.

Specifying a color

- R, G and B are numbers between 0 and 255.
- You can write these in base 10:
  rgb(230, 102, 230)
- Or as percents:
  rgb(90%, 40%, 90%)
- Or in hexadecimal (base 16 numbers, 0-9 and A-F)
  #e666e6

Hexadecimal numbers

- Base 16 (usual numbers are base 10)
- Digits are 0...9,a,b,c,d,e,f - (0-15) (it's like a deck of cards; the face cards are higher than the numbers)
- So #10 is 16, #11 is 17, etc.
- Don't really need to convert hex to read colors. For instance, what is:
  #9900ff  #ff0000ff  #ffffff

Hexadecimal numbers

- Base 16 (usual numbers are base 10)
- Digits are 0...9,a,b,c,d,e,f - (0-15)
- So #10 is 16, #11 is 17, etc.
- Don't need to convert to read colors. For instance, what is:
  #9900ff  #ff0000ff  #ffffff

Jamie's oranges

- background-color: rgb(249,172,120)
- Lots of red, less green, even less blue.
- Jamie wanted a darker orange, so…
Jamie’s oranges

- background-color: rgb(249,172,120);
- Lots of red, less green, even less blue.
- She wanted a darker orange, so…
  - background-color: rgb(201,82,61);

Floating text around an image

- Put image, followed by paragraph, together in a div (NOT a flexbox container).
- In css, give image: `float: left;`
- Use flexbox to position the outer div.
- New example on assignment page.

Formal intro to Javascript

- Today, data types.
- Experiment in the Javascript console in Chrome.

Strings

- Strings are text data.
- Can use either single or double quotes, or both:
  - "Have another banana", she said.”
- Use \n for newline and \t for tab
  - "line one \n \t line two"
- This will have no effect in an HTML paragraph, just in text printed by Javascript, for example…

console.log()

- console.log() is your "print" or "print"
- In the browser, this prints into the Javascript console.

Working with strings

- Use + for string concatenation
  - let code = "32";
  - let outStr = "Today’s code is: " + code;
  // outStr gets "Today’s code is: 32"
- Indexing
  - let str = "Tue, 25 Apr 2017 10:00 AM PDT"
  - str[0];
  // value = "T"
Strings are objects

- ...and they have a lot of methods.

```javascript
let dayOfWeek = "Tue,"
dayOfWeek.slice(1,3)
// value is "ue". Starts at 1, ends right before 3.
```

- Also substring method, which is very similar, and substr, which is a bit different. Pick one, learn it, use it.

Split

- Split divides its string into an array of substrings, by cutting out the split character or split substring you give it.

```javascript
"http://www.cs.ucdavis.edu".split("/");
// value is ["http:", "www.cs.ucdavis.edu"]
```

- Why is the empty string in the output array?

Numbers

- Only floating point, although may be written differently; there are no integers!
- Conversion is automatic!
- This can lead to some interesting behavior, ie:
  ```javascript
  let a = 5 * "2.0";
  // a = 10 -- the string became a number
  let b = 5 + "2.0";
  // b = "52.0" ... why?
  ```

Explicit conversion

- To prevent errors, best to explicitly convert:
  ```javascript
  let b = Number("2.0")+5;
  ```

```javascript
let m = Number("cow"); // m contains NaN
```
- The value NaN means "not a number"
- Can also convert explicitly to String
  ```javascript
  let m = String(3)+2 // m contains "32"
  ```

= vs ==

- 2+3 = 5 // tries to set value of a number
- nasty error message

```javascript
2+3 == 5 // the Boolean test equality operator
true
```

```javascript
2+3 == "5.0" // try it!
```
### Javascript tricky questions

| let a = (3.0 === 3); |
| let b = ("a" === 'a'); |

- What is in a? b?

- Both true.