

## ECS 162 WEB PROGRAMMING

4/10

### 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

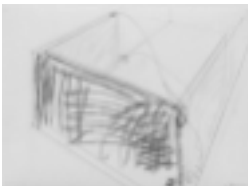
- It is easy for us to spend more time with people who know the material than those that don't.
- Labs sections will be focused on reviewing and enriching material we have covered.
- When possible, ask elementary questions on Piazza – other students are eager to help!
- I'll do some quick polls on material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

### Media query example

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

### Adding and deleting content

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



Say it has id “sketch”

### 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:  

```
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 */
#sketch.lessContent { /* 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 256.
- 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 ?  
#aaaaaa ?  
#eeee22 ?

## 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 really need to convert to read colors. For instance, what is:  
#9900ff ? Bright bluish-purple  
#aaaaaa ? Dark gray  
#eeee55 ? Pale yellow

## Jamie's oranges

background-color: rgb(249,172,120)

- Lots of red, less green, even less blue.
- She 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

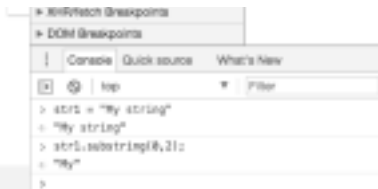
- Put image, followed by paragraph, together in a div (NOT a flexbox container).
- In css, give image:  
float: left;
- Use flexbox to position the outer div.
- New example on assignment page.



This exhibition coincides with a major Neuman retrospective, an organized by The Museum of Modern Art (MoMA) and the Stedelijk Museum, the artist's first in over twenty years. While the retrospective provides an opportunity to survey the full breadth of Neuman's work, the present exhibition at the Ronald M. Starbuck Museum focuses on a single, exceptional artwork: a large-scale participatory environment from 1970-71. Between 1969 and 1976, Neuman designed some forty rooms, rooms, and other architectural installations. The majority of these installations have been built for exhibition on at least one occasion, but Blue and Yellow Consider is one of a handful that have remained on paper. In the form of a drawing by the artist, although Neuman

## 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 "printf" 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.

```
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.

```
"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:

```
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:  
let b = Number("2.0")+5;

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

## = vs ==

```
2+3 = 5 // tries to set value of a number
```

- nasty error message

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

- true

```
2+3 == "5.0" // ?? try it!
```

## ===

- Do equality testing with type checking instead of type conversion with ===

```
2+3 === 5 // true
```

```
2+3 === "5.0" // false!
```

## Javascript tricky questions

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

- What is in a? b?

## Javascript tricky questions

```
a = (3.0 === 3);  
b = ("a" === 'a');
```

- What is in a? b?
- Both true.