JavaScript

the language of the web

story of JavaScript

haters gonna hate...

- created in 10 days (and it shows)
- renamed from Mocha to LiveScript to JavaScript (trying to be popular)
- created by by Brendan Eich (controversy)
- prototypal, NOT object oriented (get over it)

...but JavaScript is great

- your work can live on the web, so anyone can experience it
- so many tools in JS to make your life easier
- just be a responsible programmer and only use the good parts of the language, and don't try to pretend JS is OO (JavaScript, the Good Parts)

cool things

cool things

- http://www.baroque.me (web audio)
- http://labs.dinahmoe.com/plink/ (web audio)
- http://jsfiddle.net/juansrx/mt6jcwwt/ (p5js)
- http://jennz0r.github.io/wasteland/ (google maps api)
- http://www.patatap.com/ (web audio and svg)
- http://pablotheflamingo.com/ (threejs)
- http://www.creativeapplications.net/javascript-2/fuelband-fibers-visualizing-training-data-with-plask/ (plask, pex, webGL, node.js)

MAMP

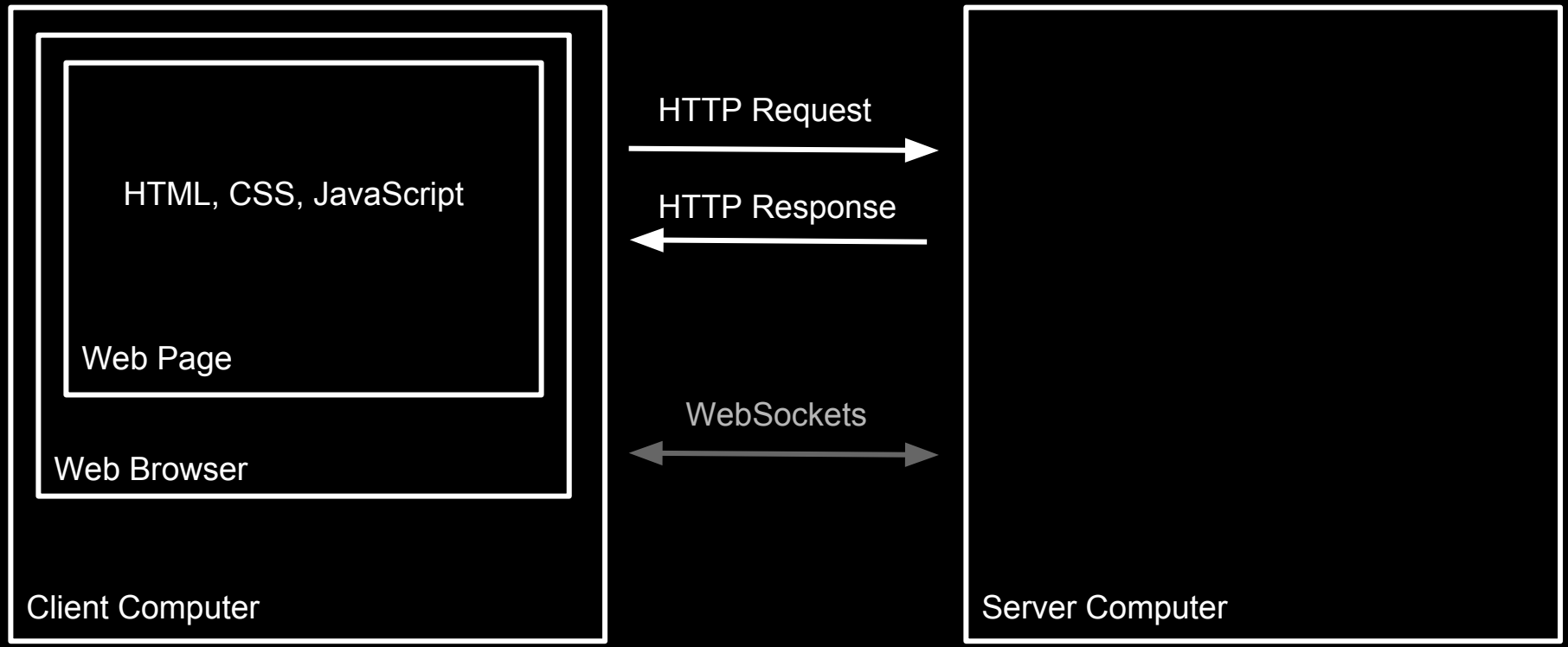
MAMP: Setting up our server

- Download and install MAMP/WAMP
  - http://www.mamp.info (OSX)
  - http://www.wampserver.com (Windows)
  - sudo apt-get install lamp-server (Ubuntu)
- Open and click “Start Servers”
- (why are we doing this? : for some features you can’t just open an HTML page locally)

MAMP: localhost / 127.0.0.1

- To access your web page, go to localhost:8888
- localhost is the “domain”
- 8888 is the “port”
- bookmark the htdocs folder
- (for most web pages, you don’t have to type the port because it uses port 80 which is the default)

Client & Server



p5js

p5.js

- Download p5.js from p5js.org
- put it in htdocs so MAMP can find it
- rename or copy the folder “empty-example” to “example”
- Make sure you do “Start Servers” in MAMP
- Go to localhost:8888/example in Chrome

P5 example

- image_weaver
- http://www.fastcodesign.com/3040714/see-every-instagram-you-took-last-year-woven-into-one-image#1

surprise! you're already writing JS!

how does p5js work?

- pretty much, it’s Processing in JavaScript
- p5js will put a <canvas> element in index.html. We can draw to the canvas and to the whole page
- sketch.js is like our Processing script

differences in P5

- a few syntax changes from Processing
- https://github.com/lmccart/p5.js/wiki/Processing-transition

JavaScript Basics

Variables

- variables are like lockers for storing things
- unlike in Processing, they don’t need a type in JavaScript

```
var myThing = something;
```

Arrays

- if a variable is a locker, then an array is like many lockers all in a row, which can make it easier to find things

```
var fruits = [“apple”, “banana”, “pear”];
// fruits[0] is “apple”, fruits[1] is “banana”, etc
```

changing HTML from p5js

- change_text

Objects

```
var fruits = {
    apple: {
        delicious: 3,
        honeycrisp: 4
    },
    banana: 3,
    pear: 1
};
// fruits.apple is 2, fruits.banana is ‘cat’, etc
```

built-in objects

JavaScript gives you some great objects automatically that have tons of helpful functions attached to them

console

document

window

Functions

- setup() and draw() are functions
- we can also write our own functions
- ex. write a function that sums the two numbers you give it (arguments, returning a value)

The Human Wasteland (poop)

- http://jennz0r.github.io/wasteland/
- How does this work?

API

- Application Programmer Interface
- lets you as for data from some other service
- ex. Google Maps API

AJAX

```
$.ajax({
    type: 'GET', // dear server, please get me poop
    url: "http://poopserver.com/poop_please",
    success: function(data) {
        console.log('success! poop:', data);
    }
});
```

The Human Wasteland (poop)

- make a basic html page where you want to show this map
- include a title, a description, and give credit to the original with a link

fin

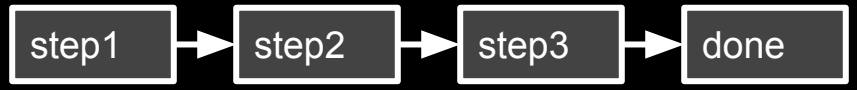
super secrete supplementary materials (i.e. what we will probably do on Thursday and/or Saturday)

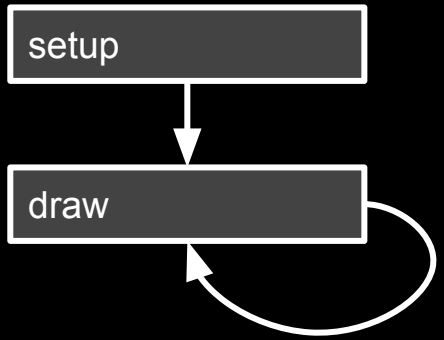
Logo Turtle

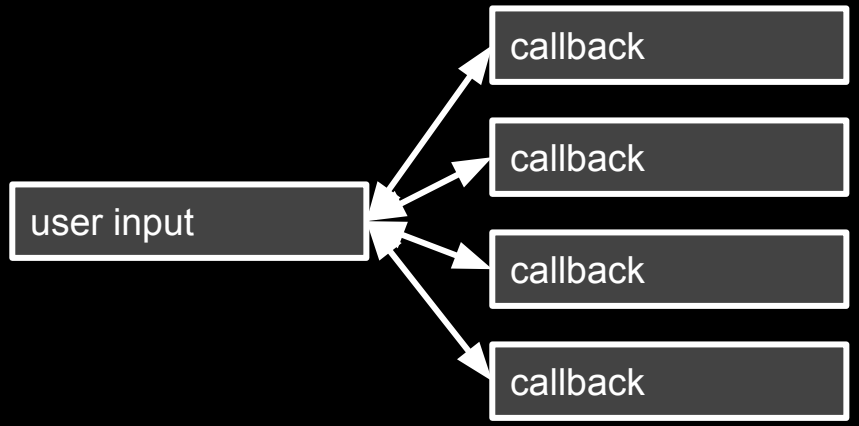
~

Click Beetles

Logo: linear programming







p5 / Processing: runtime loop

User Interfaces: callbacks

sounds

http://labs.dinahmoe.com/plink/

example_sound

three.js

drawing in 3D with webGL made easier in JS

http://cabbi.bo/

further reading

- JavaScript, the Good Parts (I can email you a copy)
- understanding this
- apply, call, bind (more on this)

equality comparison

just use === and !==, *really*

== and != were a mistake

```
'' == '0' // false
0 == '' //
true 0 == '0' // true
false == 'false' // false
false == '0' // true
false == undefined // false
false == null // false
null == undefined // true
' \t\r\n ' == 0 // true
```

The lack of transitivity is alarming. My advice is to never use the evil twins. Instead, always use `===` and `!==`. All of the comparisons just shown produce `false` with the `===` operator.

--JavaScript, the Good Parts

Rock Paper Scissors

example_rps

heat map

https://jennz0r.github.io/wasteland/

example_heatmap

Literals

- literals are constant, which means they cannot be modified
- numbers: 1, 2.5, -3000
- strings: “Hello World!”
- boolean: true or false
- NaN
- null
- undefined

Functions II

- JavaScript functions have special “this” and “arguments” variables
- “this” is whoever invoked the function*** (http://javascriptissexy.com/understand-javascripts-this-with-clarity-and-master-it/)
- “arguments” is whatever get passed in
- functions are objects, objects are functions
- ex. write a function that sums all the arguments you give it (arbitrarily many)