CS196

#8

Today

Intro to Web Dev

Javascript

Announcements

Homework

Project Meetings

StateFarm

Hackerspaces

Mobile Dev: Siebel 1105

Data Science: Siebel 0216

Web Dev: Siebel 1304

Office Hours

Office Hours will take place after Hackerspaces from **8-9 pm**

Siebel 0216 on **Wednesday**
Siebel 1111 on **Thursday**

Attendance

https://goo.gl/iYigpc

Keyword given at end of lecture

StateFarm

CS196

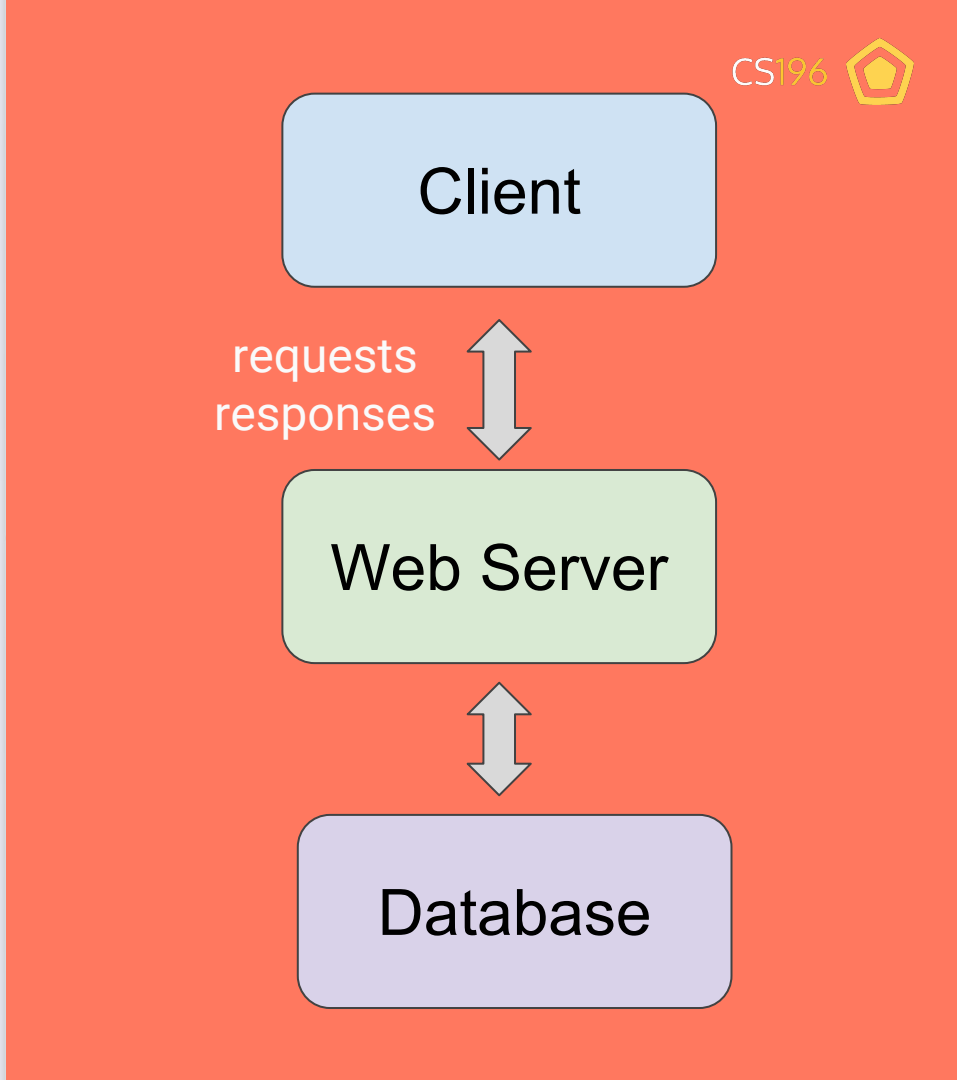
Web Dev

https://codepen.io/aria2828/pen/XeoNJo?editors=1101

Web Programming

HTML	CSS	Javascript
define content	specify layout	define behavior
• Text • Images • Links	• Colors • Alignment • Fonts	• Buttons • Graphics • Interactive features

Client Server Side Example



Hyper Text Markup Language

Content wrapped in tags

<tagname > content </tagname>

File Level Tags

Provide structure for web page

<html> <html>
<head> </head>
<body> </body>

Block Tags

Contains content

It always starts on a new line and takes the full width available

<h1> </h1>

<h2> </h2>

Inline Tags

Contains content

Doesn't take up a new line and doesn't take up the max width

Links

<a href=
“google.com”>Google

a = hyperlink

href = hypertext reference
source of file

Images

```
<img src = “sipsTea.jpg”
     alt= “Sips Tea Image”/>
```

src = source of image

alt = alternative isn’t rendered

Lists

CS196

Bullet List

```
<ul>
    <li>Sips Tea</li>
    <li>I love boba tea</li>
</ul>
```

Numbered Lists

```
<ol>
    <li>Sips Tea</li>
    <li>I love boba tea</li>
</ol>
```

Span and Div

Adding some structure to our web page

span is an **inline element** while **div** is a **block element**

- Used to group things together, really helpful for when you need to style a group of things

Classes and Ids

Classes: can be used so that different elements can have the same style

Id: an id must be unique, therefore it is used to differentiate certain elements/groups of elements. (Really useful when you start to use js)

Cascading Style Sheet

```
Tagname {
    Property: value
}
```

Access elements inside of html and give them properties

- Changing text color
- Text Alignment
- Text font
- Background color
- Text size

Examples

```
body {
    color: blue;
    text-align: center;
}
p {
    font-family: Times, serif;
    background-color: lightblue;
    font-size: 40px;
}
```

JavaScript

Scripting Language

Java != JavaScript

Can change html elements

Javascript Example


```
<button type=”button”onclick=”MyFunction()”> Click Me! <button>
<p id=”demo”> This is a demonstration. </p>


<script>
    function myFunction() {
        document.getElementById("demo").innerHTML = "Hello
JavaScript!";
    }
</script>
```

Web Servers

Delivers Web Content
Used to

- Host Websites
- Gaming
- Data Storage
- Running enterprise applications

Model View Controller

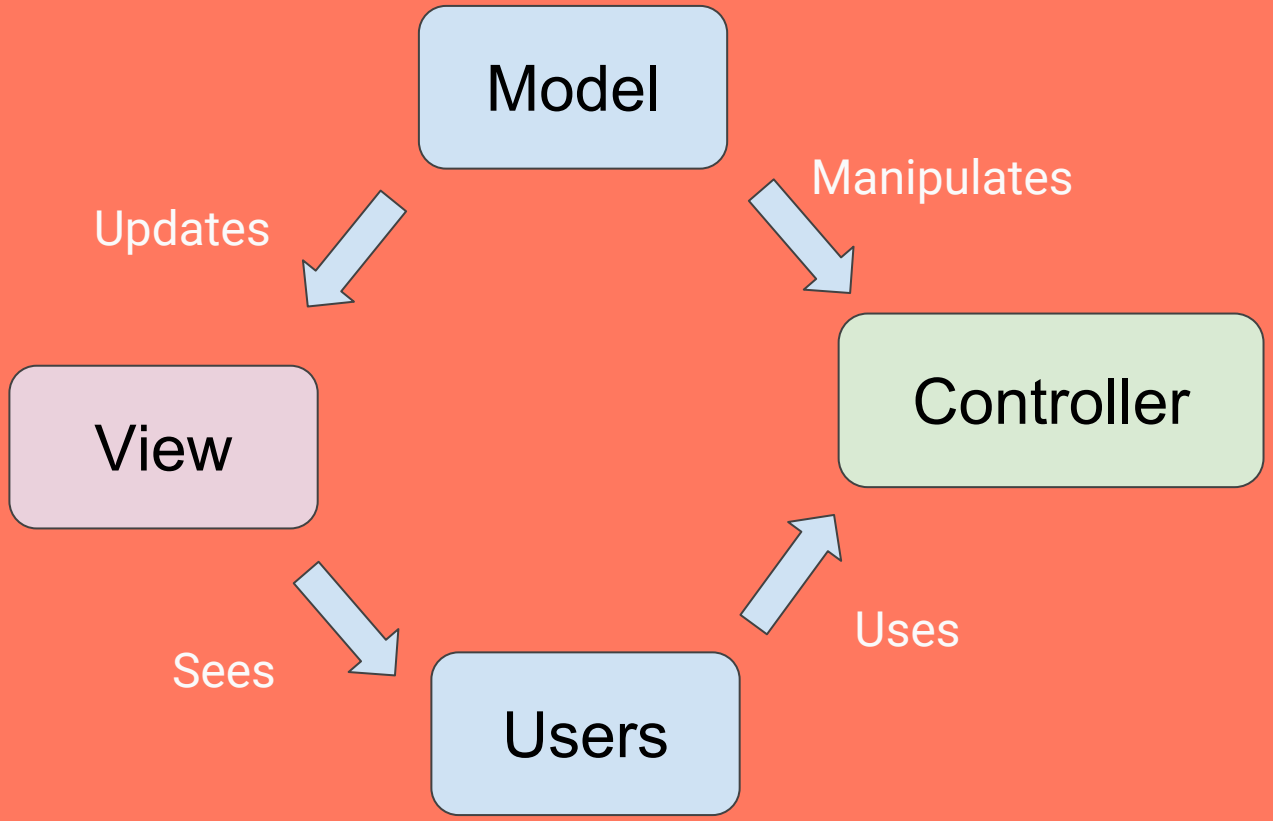
Architectural Pattern

Model: Expresses Behavior, Manages Data

View: Representation of Information

Controller: Handles inputted data

MVC


Model
Manipulates
Updates
Controller
View
Uses
Sees
Users

CS196

JavaScript

What is JavaScript?

Java Script

Only runs in the browser...

Terrible syntax...

Just a toy...

scripting or **ipt anguage** is a programming **lan age** that supp programs written for a special run-time environment that automate the execution of tasks that could alternatively be executed one-by-one by a human operator. **Scripting languages** are often interpreted (rather than compiled).

JavaScript Is...

- high-level
- dynamic
- weakly typed
- object based
- multi-paradigm (imperative, event-driven, functional, object-oriented)
- interpreted

Why Learn JavaScript?

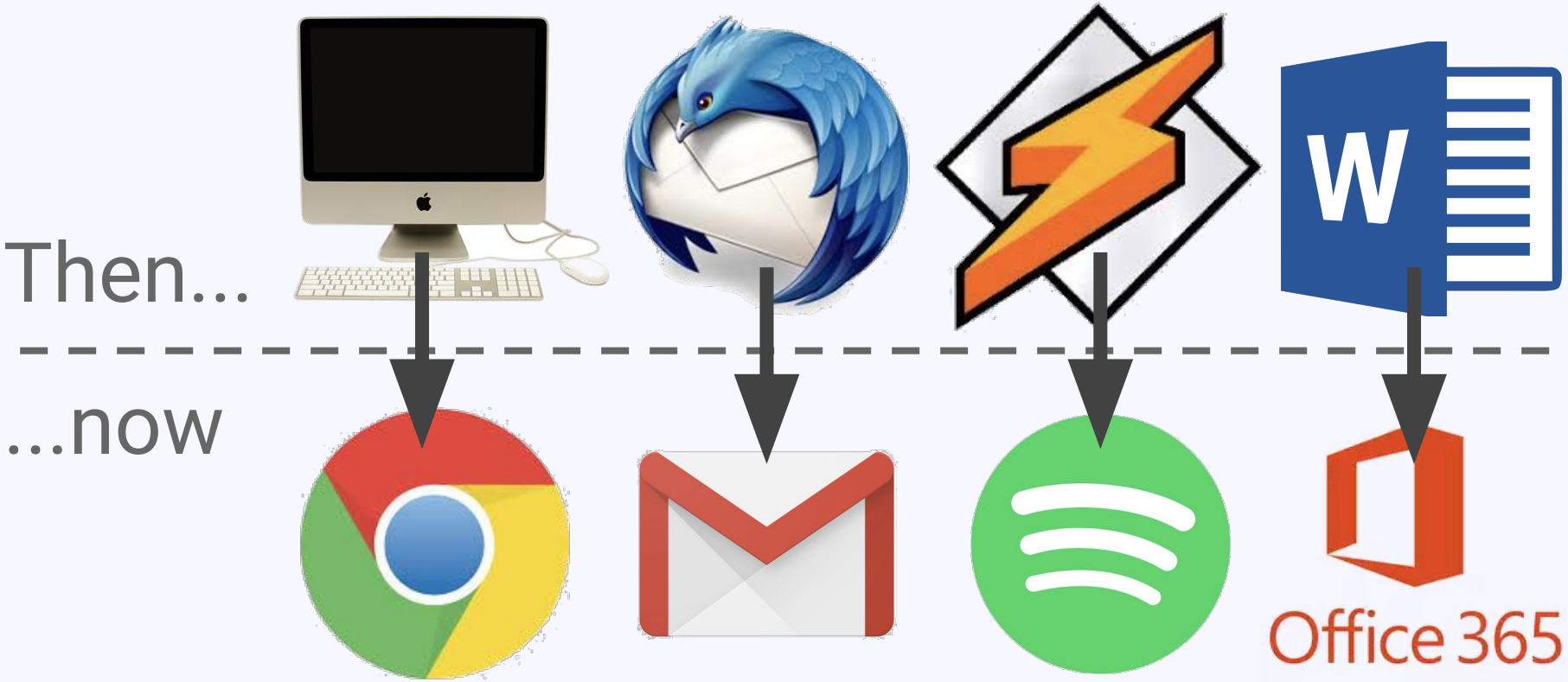
You Have No Choice

JavaScript, Java, Python, PHP, Ruby, C++, C, Objective C, R, Go, Perl, Swift, Scala, Groovy, Rust, D

JavaScript

You Have No Choice

You Have No Choice



JavaScript Runs Everywhere

JavaScript

Java

Script

JavaScript, Java, Python, PHP, Ruby, C++, C, Objective C, R, Go, Perl, Swift, Scala, Groovy, Rust, D

Full Stack


```
$ node yourcode.js
```

- Server side
- Since 2009

```
<script>yourcode</script>
```

- Browser side
- Since forever

JavaScript Runs Everywhere

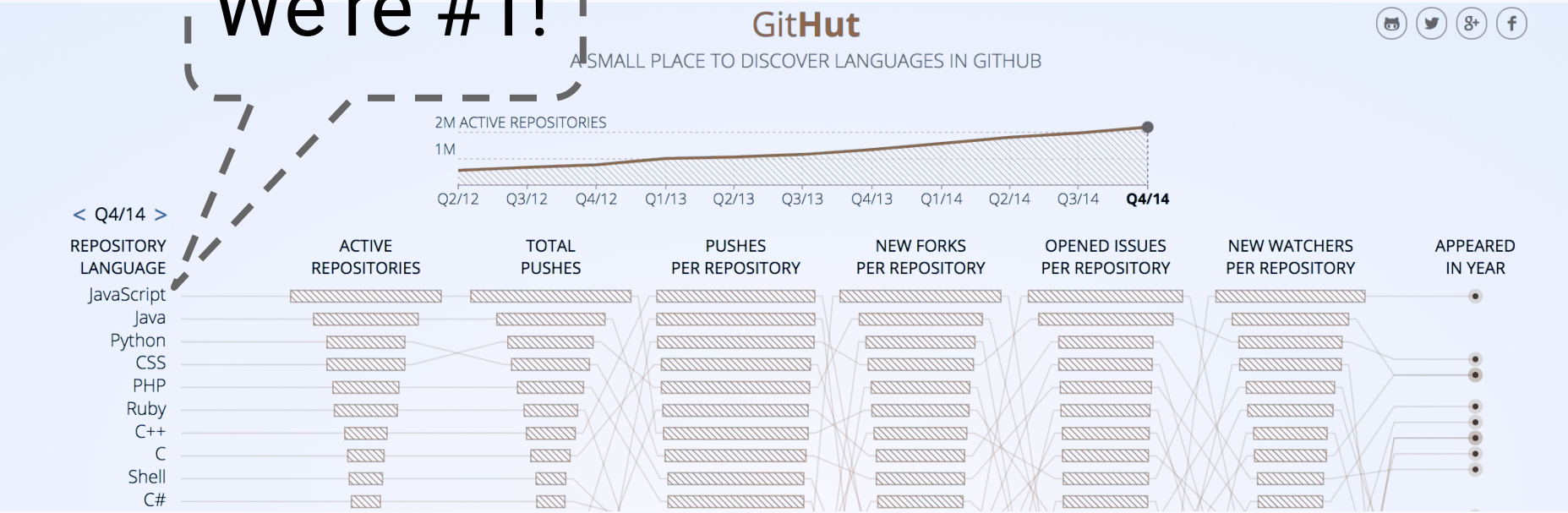
And JavaScript can be used to run *almost anything*:

- Web applications
- Web APIs
- Actual scripts
- Command-line utilities
- GUI applications (meaning web applications)

JavaScript Has a Great Community

CS196

We're #1!



http://githut.info/

JavaScript Has a Great Community

JavaScript Has a Fantastic Package Manager

```
$ cd myproject
$ npm install --save multimatch


$ git clone <whatever>
$ cd <whatever>
$ npm i
```

JavaScript Has a Fantastic Package Manager

CS196

Stats

4 downloads in the last day

12 downloads in the last week

31 downloads in the last month

No open issues on GitHub

No open pull requests on GitHub

Stats

541,491 downloads in the last day

3,266,244 downloads in the last week

11,827,825 downloads in the last month

222 open issues on GitHub

74 open pull requests on GitHub

These are **not** the droids I'm looking for...

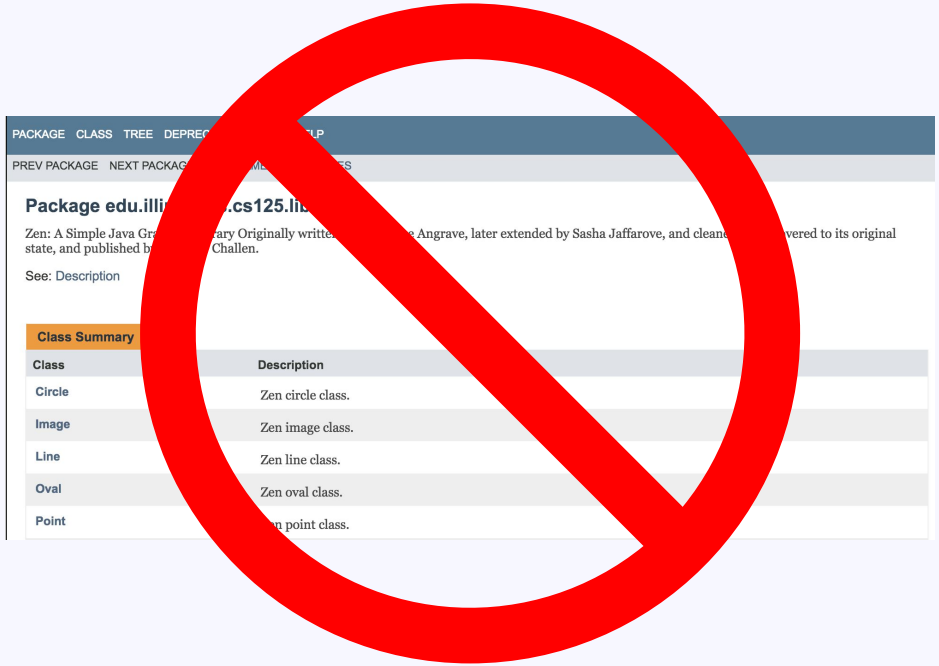
https://npmjs.org

JavaScript Has a Fantastic Package Manager

CS196

NPM makes it *extremely easy* to publish libraries. As a result, JavaScript libraries tend to be:

- Small
- Well-documented
- Well-maintained
- Easy to evaluate



JavaScript is Flexible and Evolving


```
1  var array = [1, 2, 3, 4, 5, 6];
2  var sum = 0;
3  for (var i = 0; i < array.length; i++) {
4    if (array[i] % 2 == 0) {
5      sum += array[i];
6    }
7  }
8  console.log(sum);
```

https://runkit.com/gchallen/javascript-idiom-comparison/0.0.4

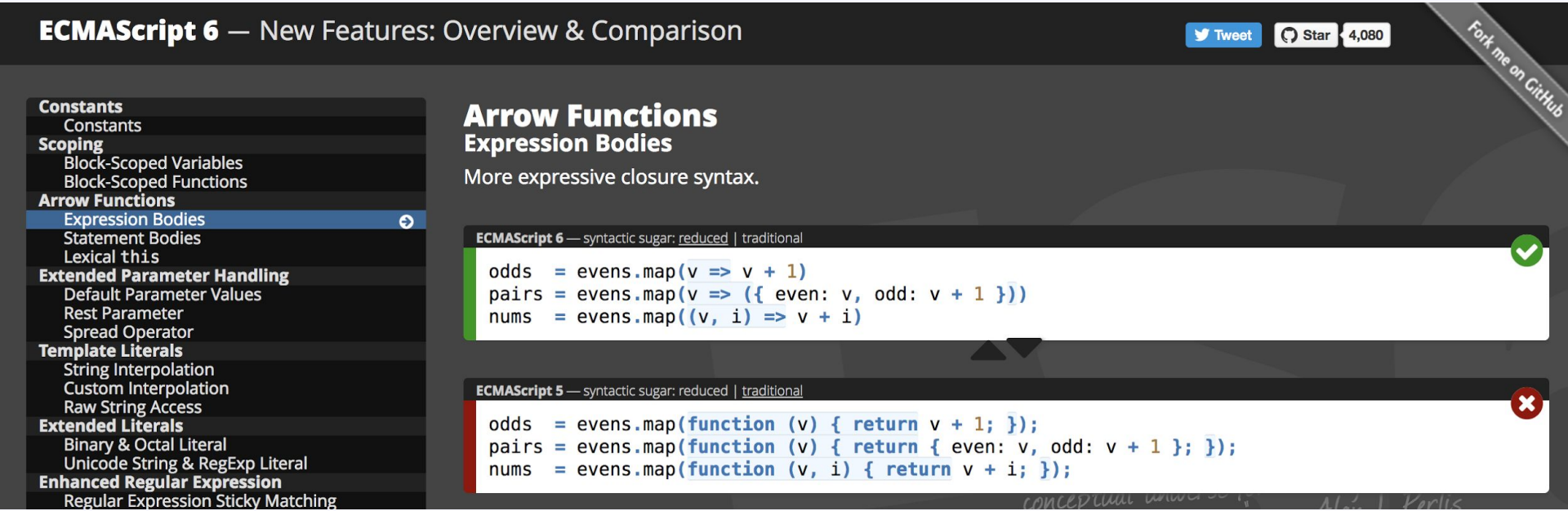
JavaScript is Flexible and Evolving


```
1  const _ = require('underscore' 1.8.3 );
2  let array = [1, 2, 3, 4, 5, 6];
3  let sum = _.chain(array)
4    .filter((value) => {
5      return value % 2 == 0;
6    })
7    .reduce((sum, value) => {
8      return sum += value;
9    }, 0)
10   .value();
11 console.log(sum);
```

https://runkit.com/gchallen/javascript-idiom-comparison/0.0.4

JavaScript is Flexible and Evolving (ES6)

CS196



http://es6-features.org/

JavaScript is Flexible and Evolving (ES7)

CS196

Async Functions

Of all the features in ES2016+, async functions are by far the biggest game changer. If you only learn one thing from this article, it should be that you need to start using async functions as soon as possible.

Why, you ask?

If you've ever written code like this, you know the pain that is asynchronous workflow in JavaScript:

```
1  function createEmployeeWorkflow(cb){
2
3    createEmployee(function(err, employee){
4      If (err) { return cb(err); }
5
6      if (employee.needsManager()) {
7
8        selectManager(employee, function(err, manager){
9          If (err) { return cb(err); }
```

JavaScript will continue challenging you as you grow as a **hacker**

Programming Is Fun?

Programming Is Fun!

(If it's not, you're using the wrong tools.)

JavaScript Is Most Fun

- Easy things are *really easy*
- Hard things are **possible**
- You can write *beautiful* code
- The libraries and community support are **fantastic**

JS

```
var
arr
3];
```

ELPFUL

Roger Hargreaves

Java

```
int[]
array
3};
```

A Brief (And Interesting) History of JavaScript (And Java)

Released

Development Time

Development Team

Result

Java's "Write Once, Run Anywhere"

1. Install an otherwise useless piece of software called the Java interpreter
2. Download a compiled class file and hope that it runs on your computer, and that you can even launch it

JavaScript's "Write Once, Run Anywhere"

1. Install a web browser
2. Visit a website

Moral of the Story:

The Internet Wins

Concurrency in JavaScript

What is Concurrency?

In computer science, concurrency is the decomposability property of a program, algorithm, or problem into order-independent or partially-ordered components or units. This means that even if the concurrent units of the program, algorithm, or problem are executed out-of-order or in partial order, the final outcome will remain the same. This allows for parallel execution of the concurrent units, which can significantly improve overall speed of the execution in multi-processor and multi-core systems.

OK, But What Is Concurrency *Really*?

If some part of your code has to stop, another part can keep going

Why Would Your Code Stop?

There are a variety of reasons that your code might have to stop:

1.
2.
3.
4.

Concurrency Is Not Parallelism

Concurrent code *can* harness, but doesn't *require* parallelism

Concurrency In JavaScript

JavaScript emerged from the browser...

1. One thread per tab that does *everything*: rendering HTML, applying CSS rules, listening for user input, painting... and **running JavaScript**.
2. As a result all JavaScript blocking operations are done asynchronously. **There is no way (until recently) for a function to wait for a blocking request to complete!**

Concurrency Example


```
1  const request = require('request' 2.83.0 );
2  let response = request('https://www.google.com');
3  console.log(response);
```

Concurrency Example

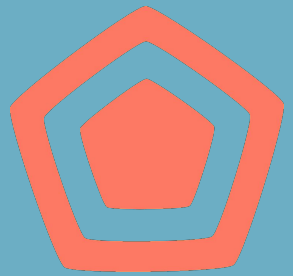

```
1  const request = require('request' 2.83.0 );
2  let response = request('https://www.google.com', function (error, response, body) {
3    console.log(body.length);
4  });
```

https://runkit.com/gchallen/asynchronous-example/0.0.1

Let's Write Some JavaScript!

Fork this RunKit example:

https://goo.gl/VjJ78L



Keyword

Statefarm2