Intro to JavaScript

•••

September 29th, 2015
DeskHub

Overview

Overview

- What is JavaScript?

Overview

- What is JavaScript?
- Why use JavaScript?

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics
- How are websites built?

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics
- How are websites built?
- DOM basics

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics
- How are websites built?
- DOM basics
- Javascript basics

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics
- How are websites built?
- DOM basics
- Javascript basics
- JavaScript libraries

Overview

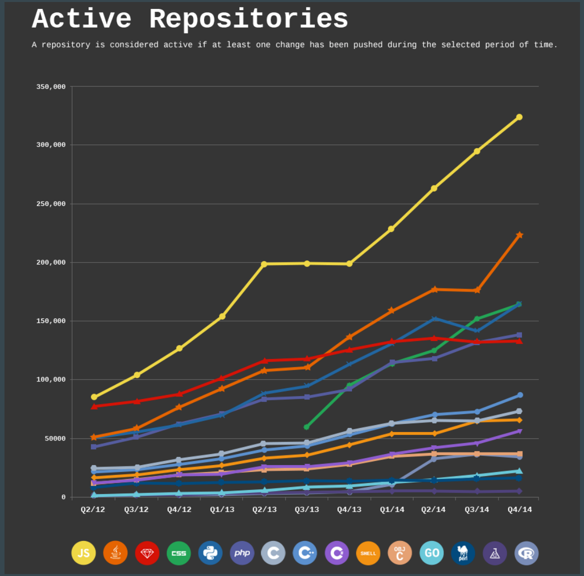
- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics
- How are websites built?
- DOM basics
- Javascript basics
- JavaScript libraries
- JavaScript frameworks

Overview

- What is JavaScript?
- Why use JavaScript?
- Where did JavaScript come from?
- HTML basics
- How are websites built?
- DOM basics
- Javascript basics
- JavaScript libraries
- JavaScript frameworks
- Snake demo

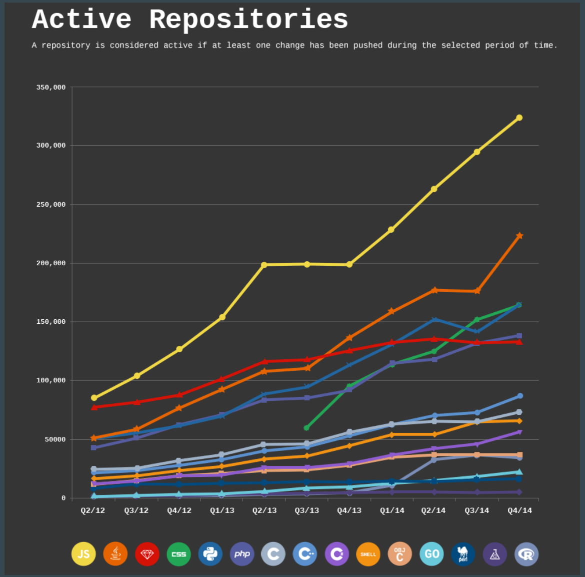
What is JavaScript?

What is JavaScript?



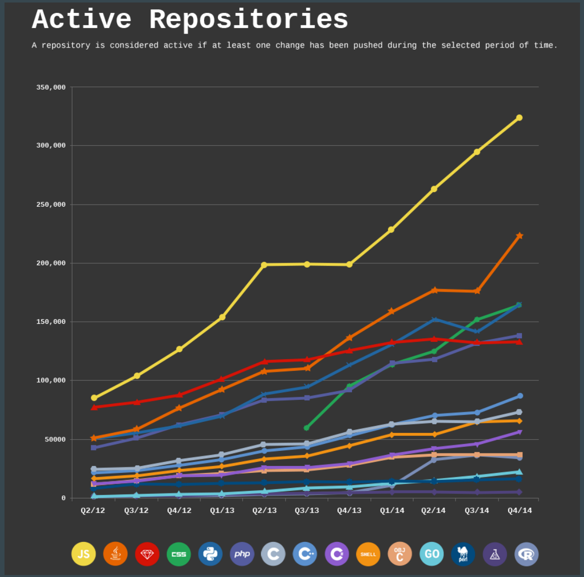
What is JavaScript?

- Most popular programming language in the world



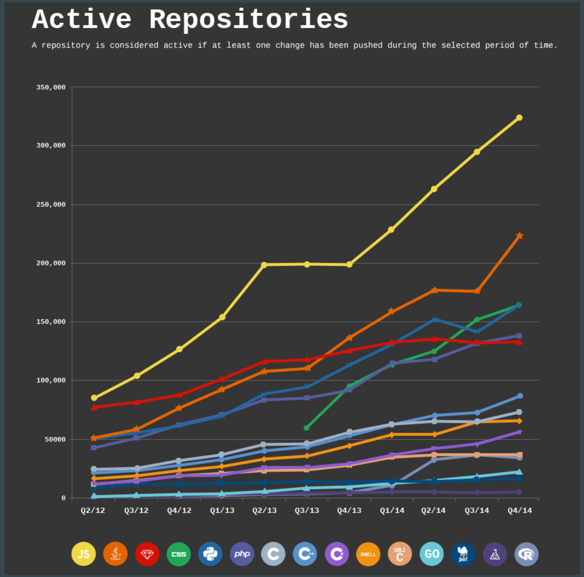
What is JavaScript?

- Most popular programming language in the world
- Runs most commonly in a web browser



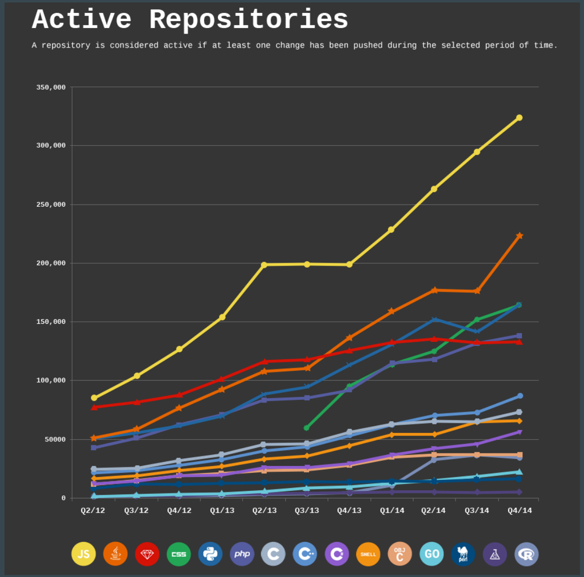
What is JavaScript?

- Most popular programming language in the world
- Runs most commonly in a web browser
- Adds interactivity to web sites



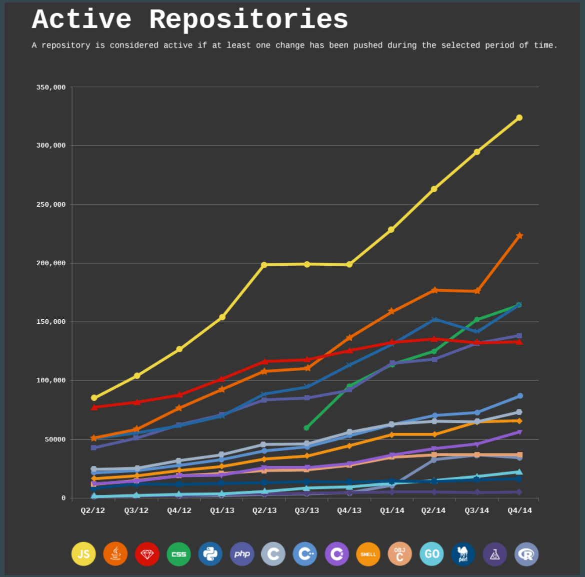
What is JavaScript?

- Most popular programming language in the world
- Runs most commonly in a web browser
- Adds interactivity to web sites
- Can also be used on the server (Node.js)



What is JavaScript?

- Most popular programming language in the world
- Runs most commonly in a web browser
- Adds interactivity to web sites
- Can also be used on the server (Node.js)
- Robots! Refrigerators! Toasters!



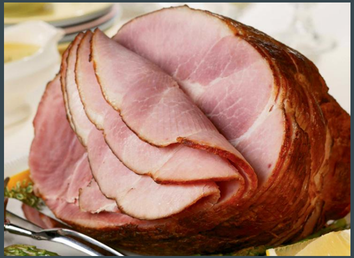
Warning!

Warning!

JavaScript and Java are very different things!

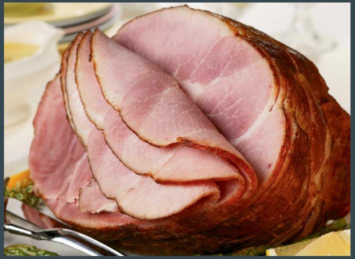
Warning!

JavaScript and Java are very different things!



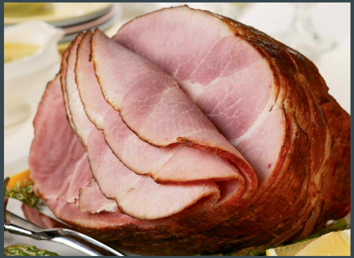
Warning!

JavaScript and Java are very different things!



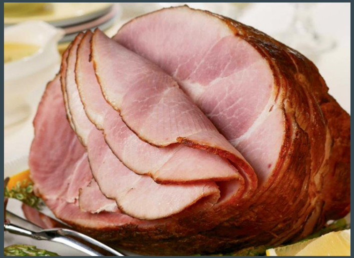
Warning!

JavaScript and Java are very different things!



Warning!

JavaScript and Java are very different things!



Why use JavaScript?

Why use JavaScript?

- You have to!

Why use JavaScript?

- You have to!
- Interactivity

Why use JavaScript?

- You have to!
- Interactivity
- In-browser games

Why use JavaScript?

- You have to!
- Interactivity
- In-browser games
- Faster than going back to the server

Why use JavaScript?

- You have to!
- Interactivity
- In-browser games
- Faster than going back to the server
- Build entire applications in the browser

Where did JavaScript come from?

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript
- FRAGMENTATION (“best viewed” in Netscape/Internet Explorer)

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript
- FRAGMENTATION (“best viewed” in Netscape/Internet Explorer)
- In 1996, standardization through Ecma International

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript
- FRAGMENTATION (“best viewed” in Netscape/Internet Explorer)
- In 1996, standardization through Ecma International
- ECMAScript standard produced by the TC39 committee in June 1997

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript
- FRAGMENTATION (“best viewed” in Netscape/Internet Explorer)
- In 1996, standardization through Ecma International
- ECMAScript standard produced by the TC39 committee in June 1997
- Editions 2, 3, 5, 5.1 from 1998 - 2011 (4 was abandoned)

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript
- FRAGMENTATION (“best viewed” in Netscape/Internet Explorer)
- In 1996, standardization through Ecma International
- ECMAScript standard produced by the TC39 committee in June 1997
- Editions 2, 3, 5, 5.1 from 1998 - 2011 (4 was abandoned)
- 6th edition published in June 2015 (ES6 or ES2015)

Where did JavaScript come from?

- Netscape wanted a “lightweight” language to compete with Java
- Developed in TEN DAYS by Brendan Eich in 1995
- Developed as Mocha, shipped as LiveScript, changed to JavaScript
- Microsoft followed with JScript
- FRAGMENTATION (“best viewed” in Netscape/Internet Explorer)
- In 1996, standardization through Ecma International
- ECMAScript standard produced by the TC39 committee in June 1997
- Editions 2, 3, 5, 5.1 from 1998 - 2011 (4 was abandoned)
- 6th edition published in June 2015 (ES6 or ES2015)
- Yearly releases from now on (ES2016, etc.)

The Web Triumvirate

The Web Triumvirate

HTML

Markup language

Content

The Web Triumvirate

HTML

Markup language

Content

CSS

Stylesheet language

Styling and layout

The Web Triumvirate

HTML

Markup language

Content

JS

General-purpose programming language

Behavior

CSS

Stylesheet language

Styling and layout

HTML Basics

HTML Basics

```
<p>hello world</p>
```

HTML Basics

```
<p>hello world</p>
```

Opening tag

HTML Basics

```
<p>hello world</p>
```

Opening tag

Closing tag

HTML Basics

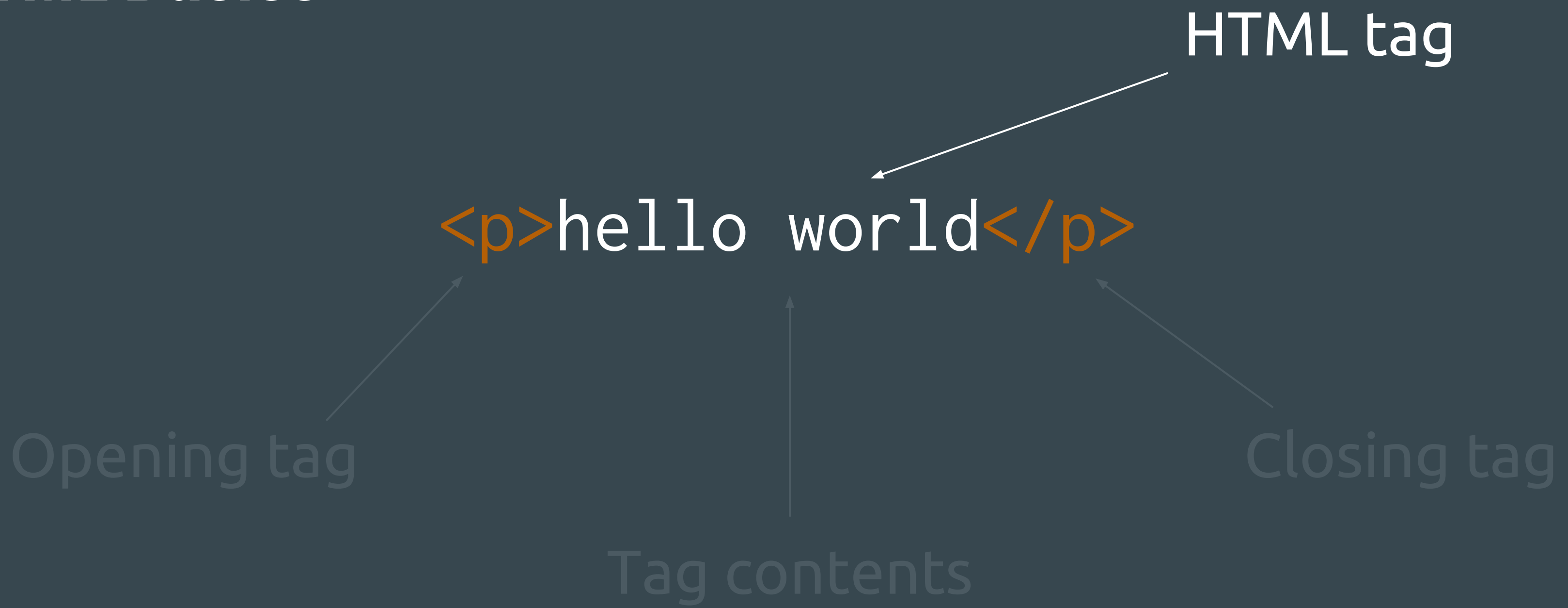
```
<p>hello world</p>
```

Opening tag

Tag contents

Closing tag

HTML Basics


```
<p>hello world</p>
```

HTML Basics

```
<p>hello world</p>
```

HTML Basics

```
<div>
  <p>hello world</p>
</div>
```

HTML Basics

```
<div                            >
  <p>hello world</p>
</div>
```

HTML Basics

```
<div align="center">
  <p>hello world</p>
</div>
```

HTML Basics

```
<div align="center">
  <p>hello world</p>
</div>
```

HTML Basics

```
<html>




    <div align="center">
      <p>hello world</p>
    </div>

</html>
```

HTML Basics

```
<html>




  <body>
    <div align=”center”>
      <p>hello world</p>
    </div>
  </body>
</html>
```

HTML Basics

```
<html>
  <head>

  </head>
  <body>
    <div align="center">
      <p>hello world</p>
    </div>
  </body>
</html>
```

HTML Basics

```html
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <div align=”center”>
      <p>hello world</p>
    </div>
  </body>
</html>
```

HTML Basics

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <div align="center">
      <p>hello world</p>
    </div>
  </body>
</html>
```

HTML Basics

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>




  </body>
</html>
```

HTML Basics

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p id="one">first</p>
    <p id="two" class="fave">second</p>
    <p class="fave">third</p>
  </body>
</html>
```

HTML Basics

```html
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p id="one">first</p>
    <p id="two" class="fave">second</p>
    <p class="fave">third</p>
  </body>
</html>
```

HTML Basics

IDs should be unique to a page!

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p id="one">first</p>
    <p id="two" class="fave">second</p>
    <p class="fave">third</p>
  </body>
</html>
```

HTML Basics

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p id=”one”>first</p>
    <p id=”two” class=”fave”>second</p>
    <p class=”fave”>third</p>
  </body>
</html>
```

HTML Basics

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p id="one">first</p>
    <p id="two" class="fave">second</p>
    <p class="fave">third</p>
  </body>
</html>
```

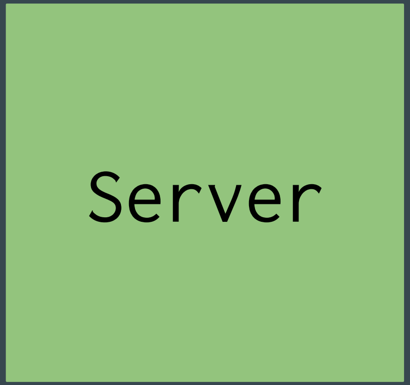
HTML Basics

Classes don’t have to be unique.

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p id=”one”>first</p>
    <p id=”two” class=”fave”>second</p>
    <p class=”fave”>third</p>
  </body>
</html>
```

Every Website Ever (pretty much)

Every Website Ever (pretty much)



Every Website Ever (pretty much)

Server

Every Website Ever (pretty much)

Server

Ruby Go Haskell

Java Python

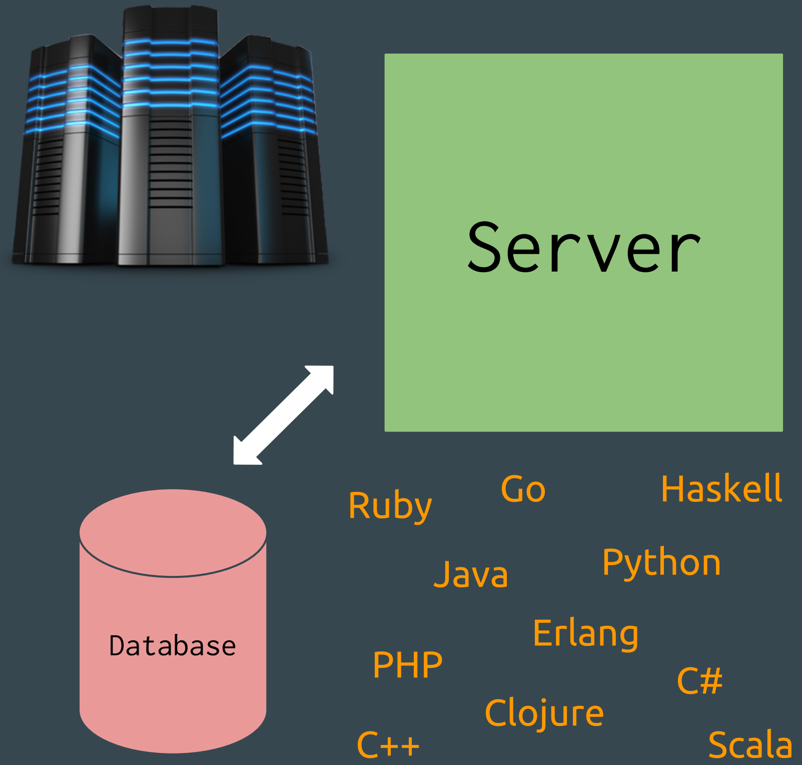
Erlang

PHP C#

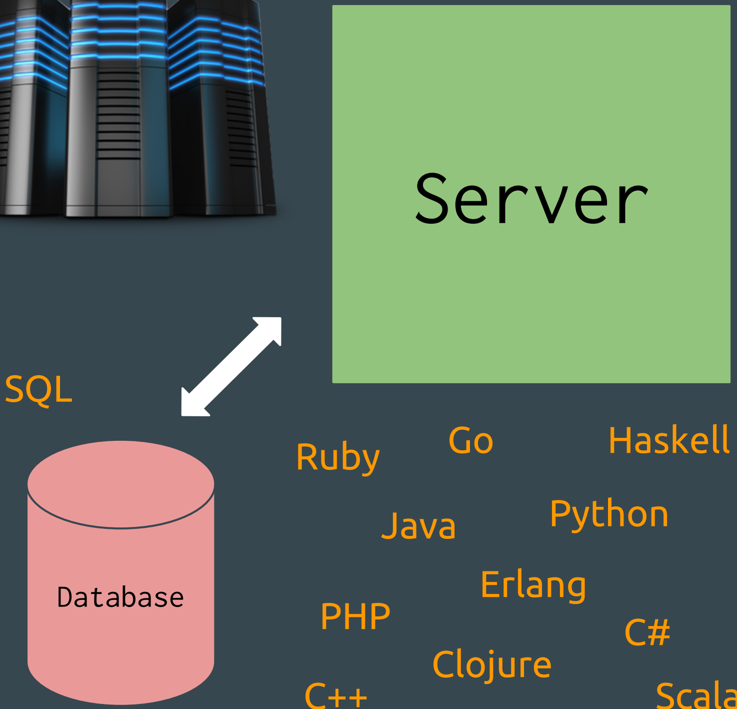
Clojure

C++ Scala

Every Website Ever (pretty much)



Every Website Ever (pretty much)



Ruby Go Haskell

Java Python

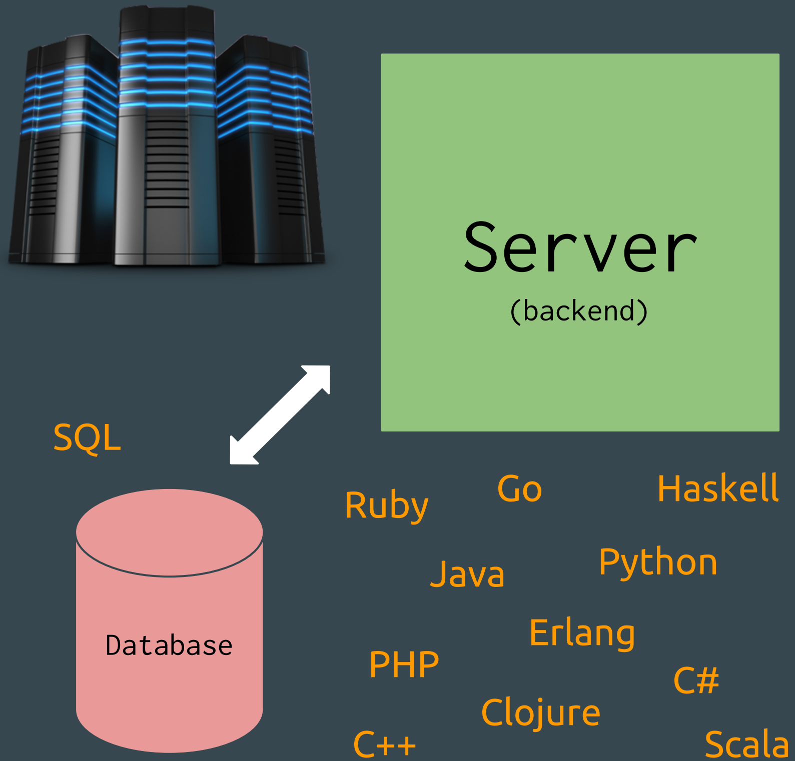
Erlang

PHP C#

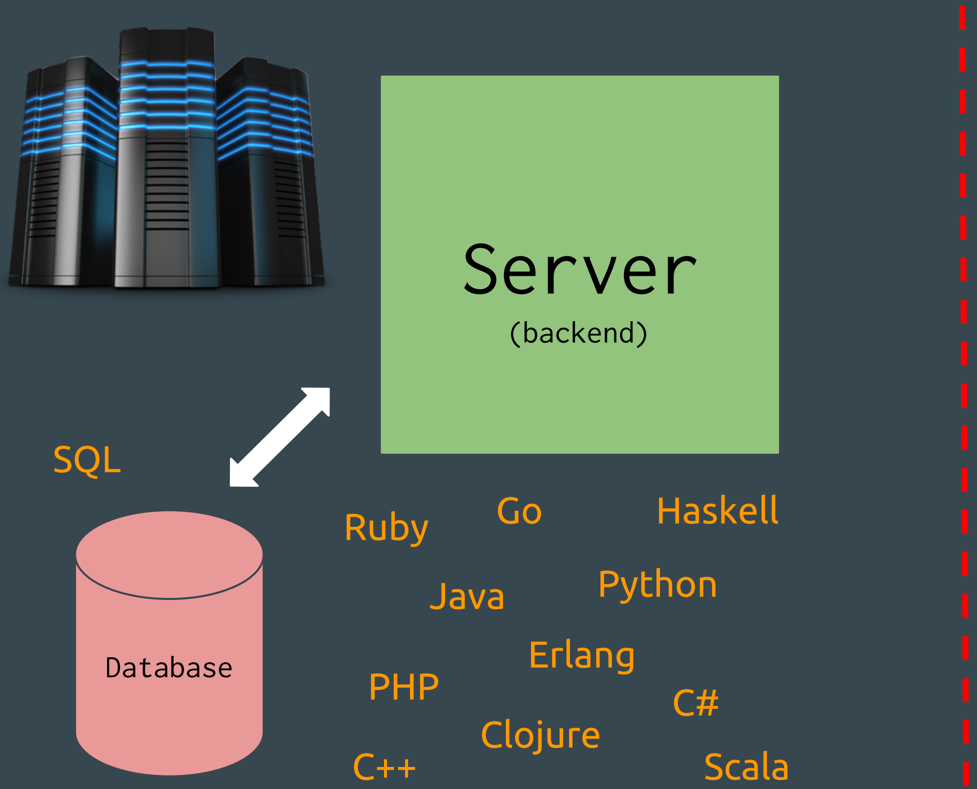
Clojure

C++ Scala

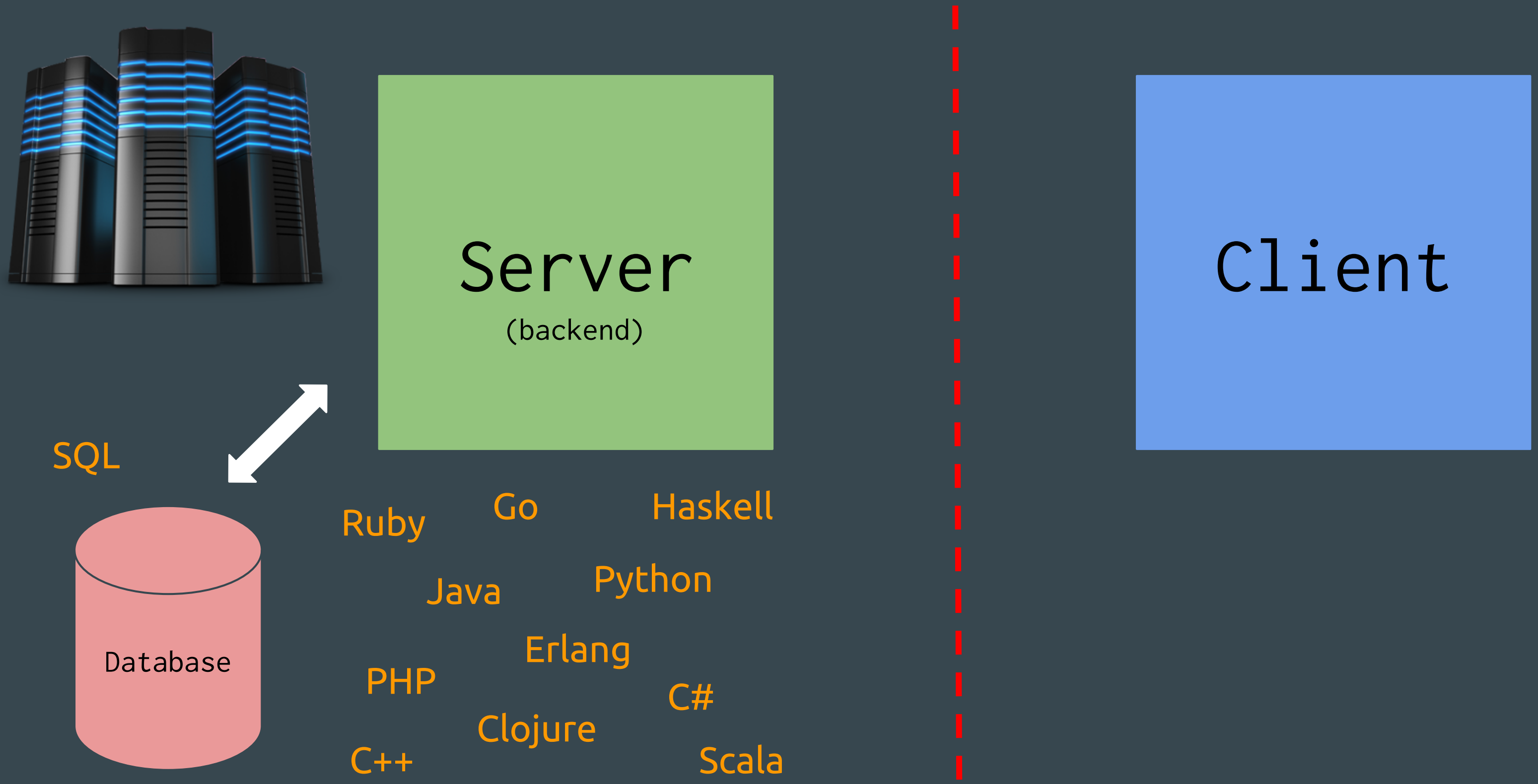
Every Website Ever (pretty much)



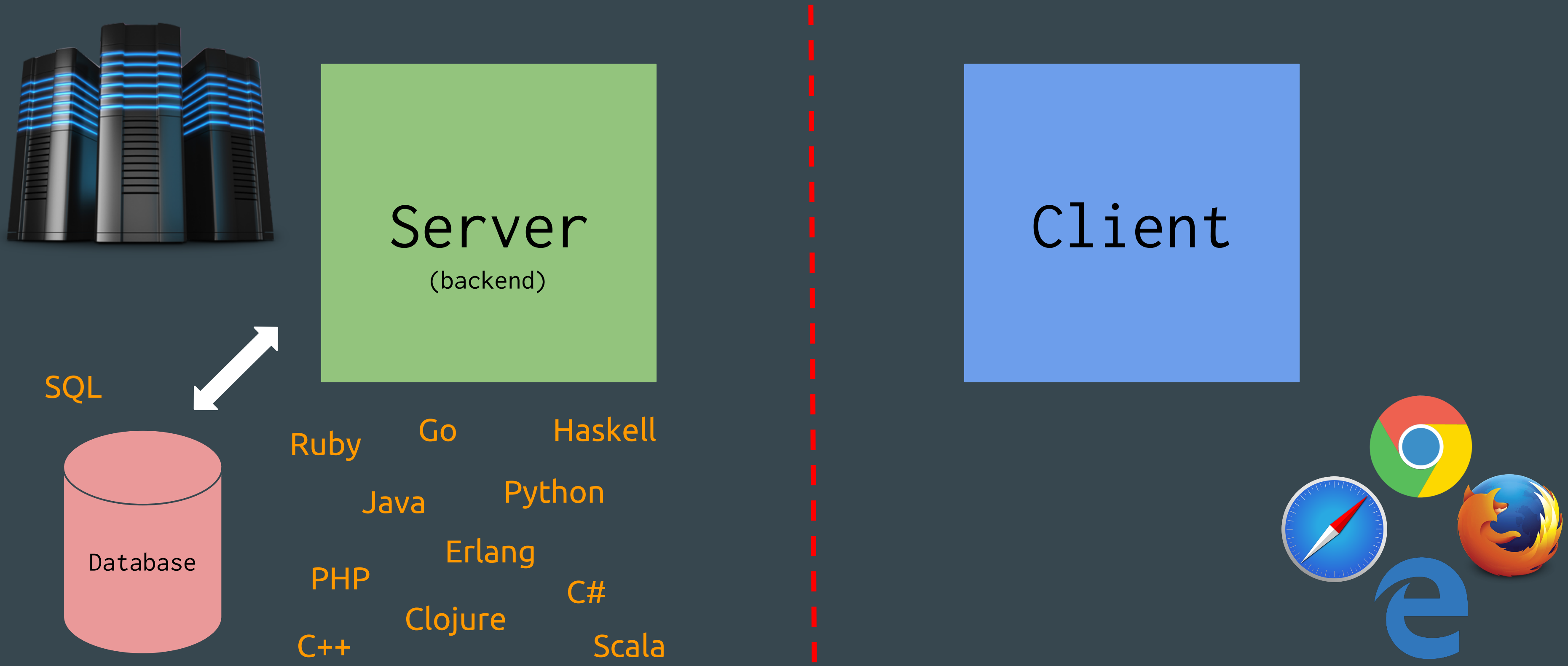
Every Website Ever (pretty much)



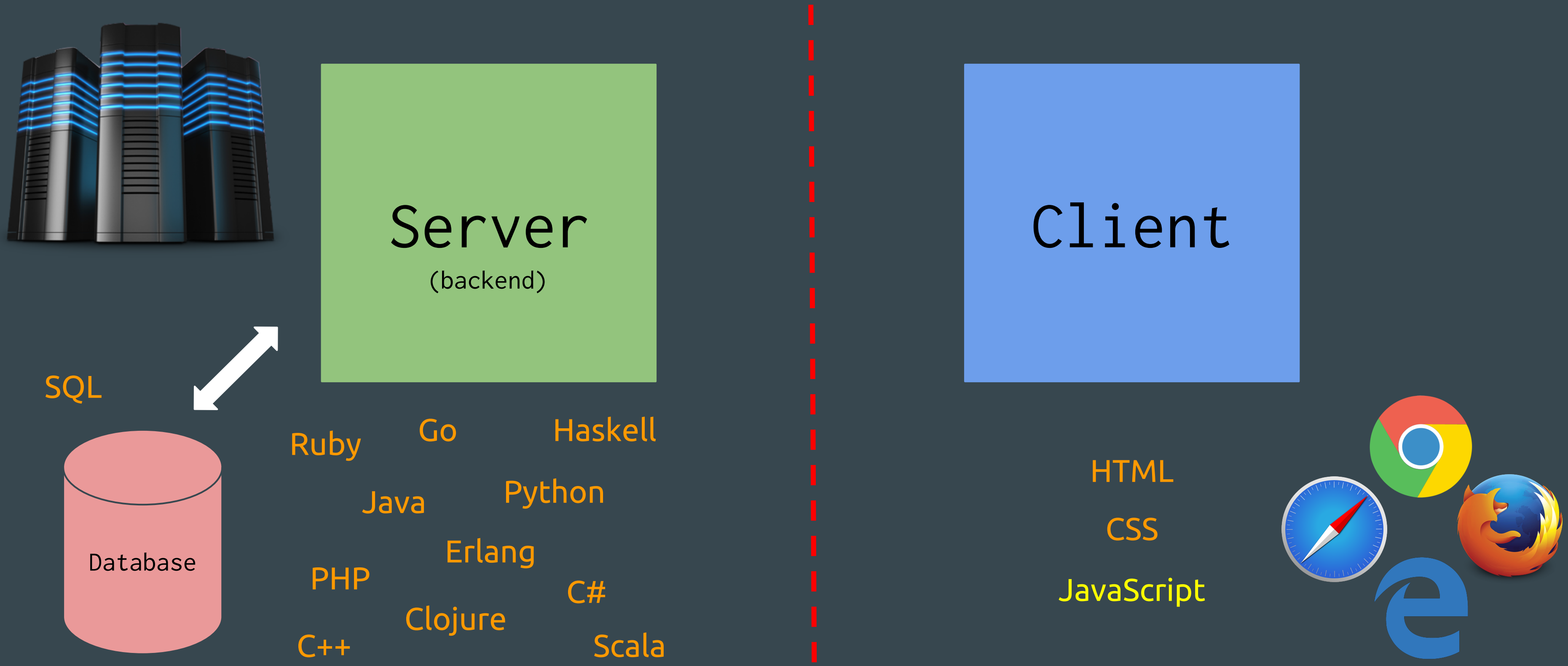
Every Website Ever (pretty much)



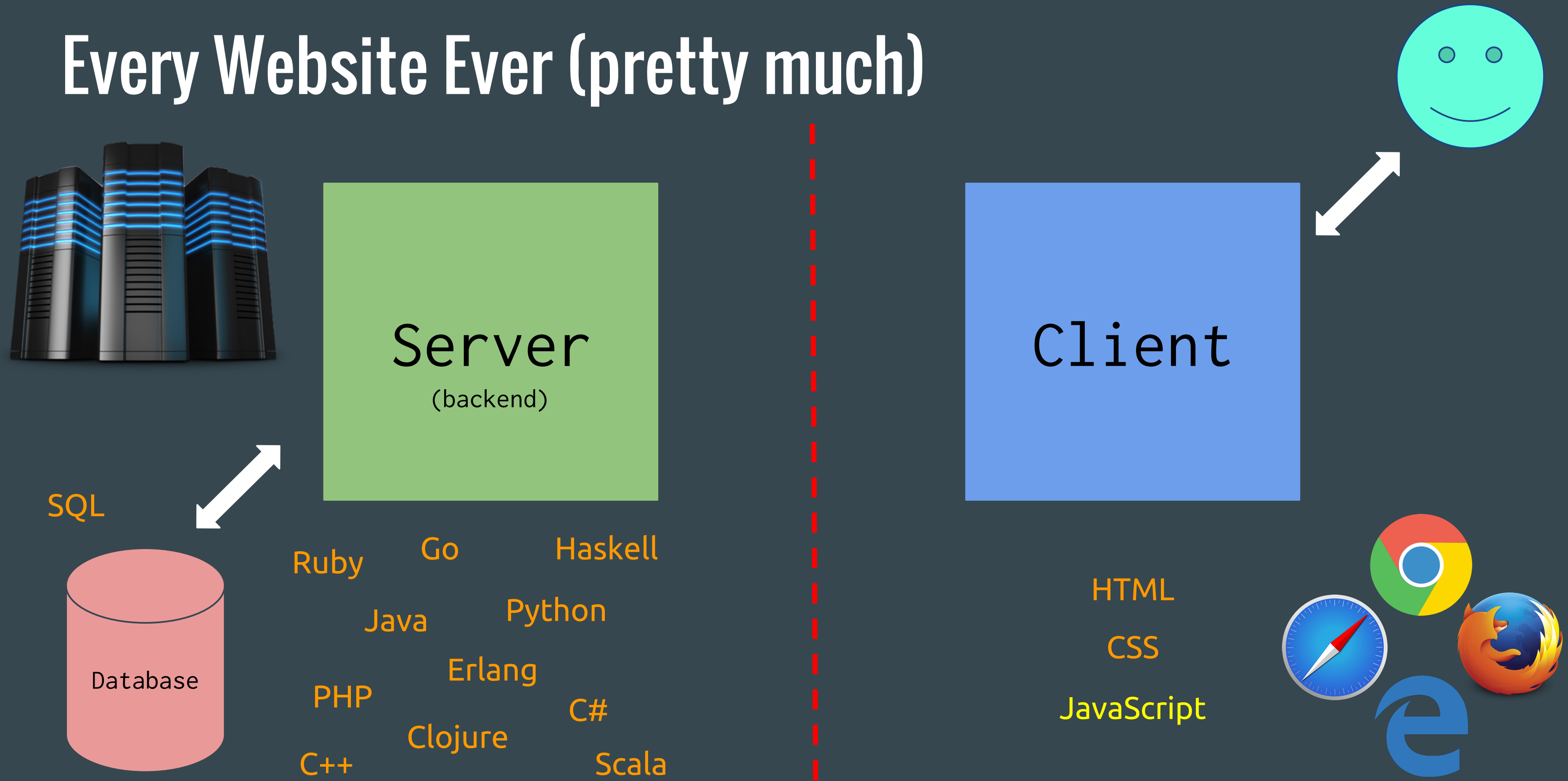
Every Website Ever (pretty much)



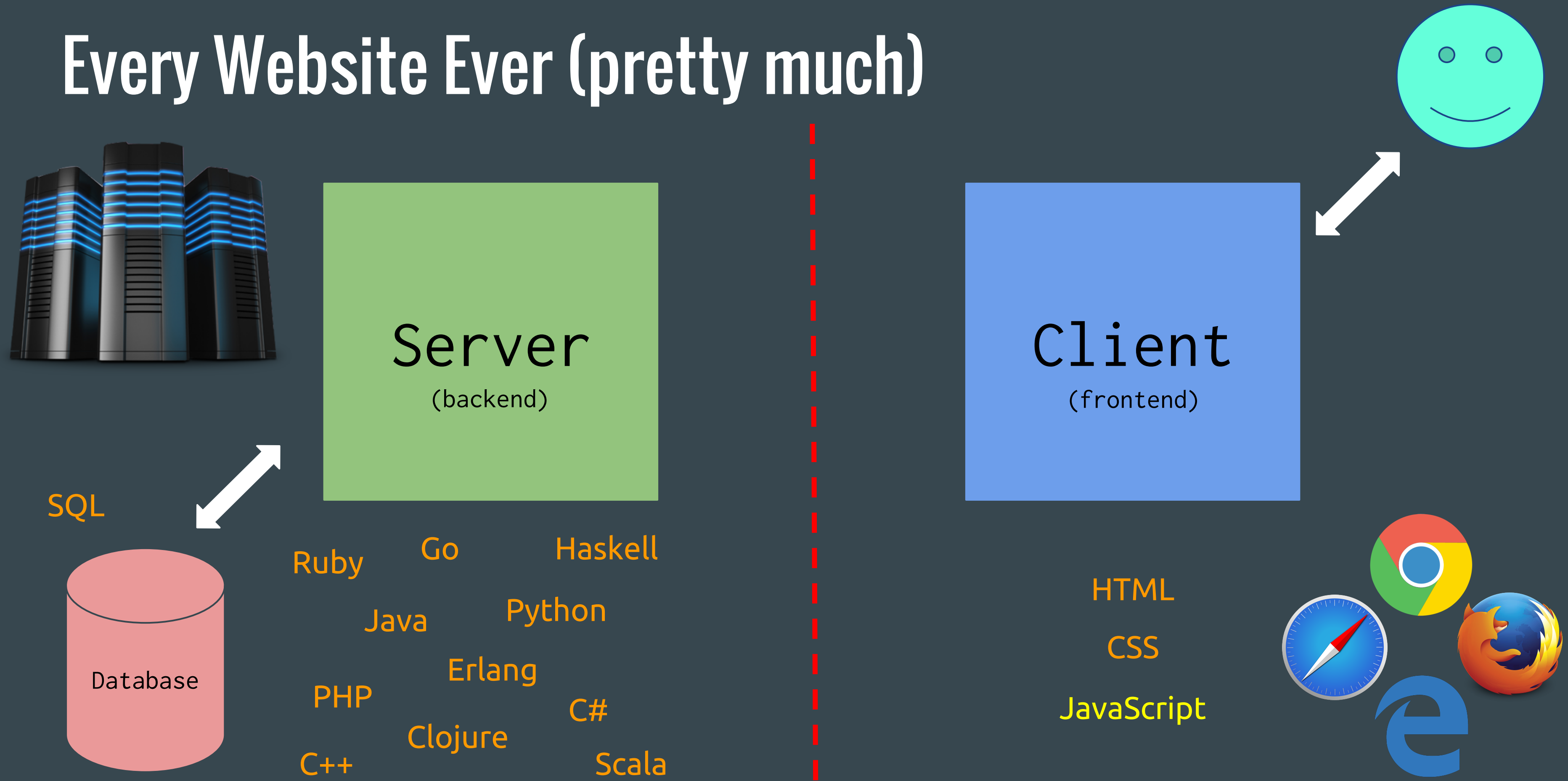
Every Website Ever (pretty much)



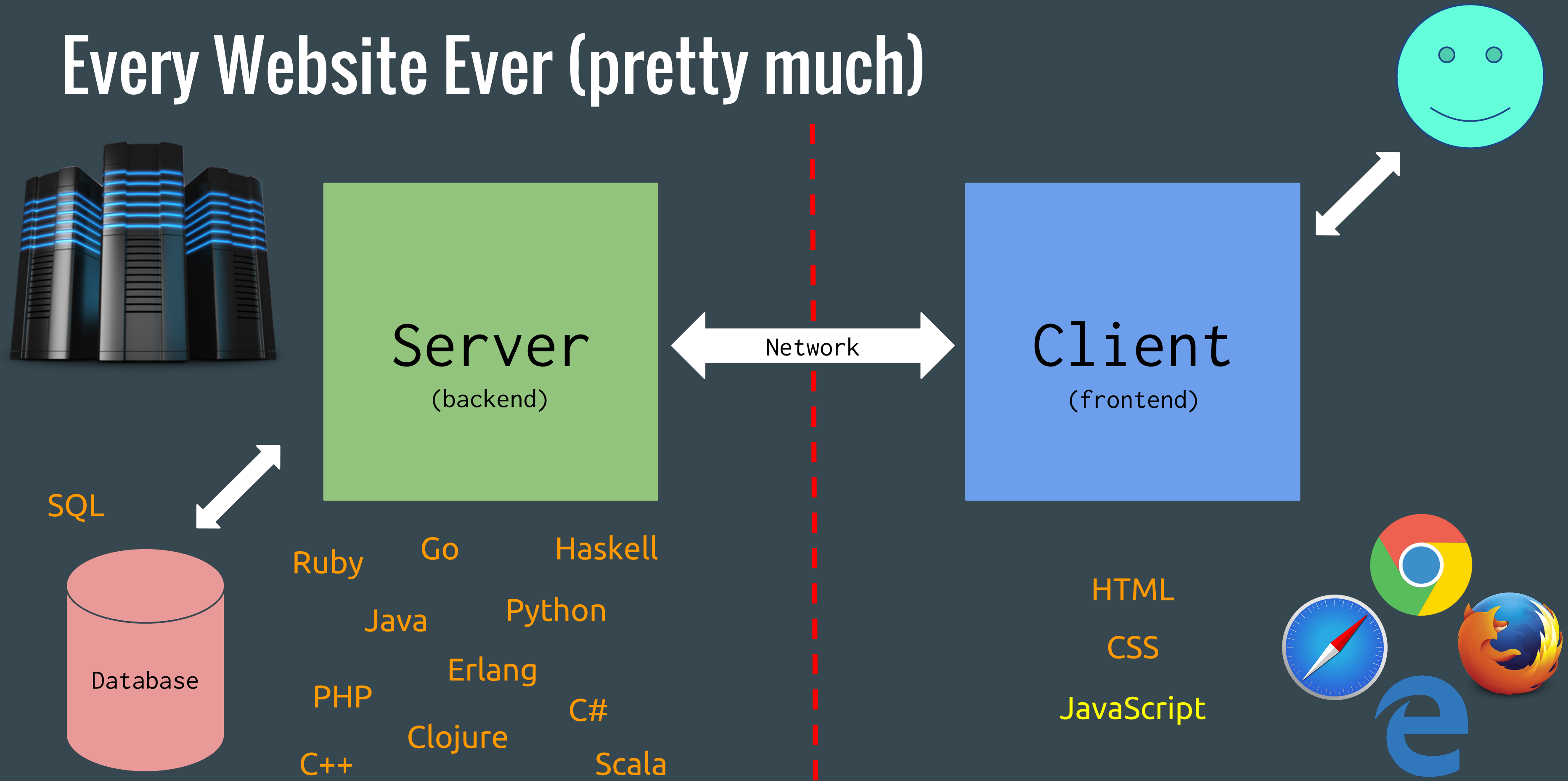
Every Website Ever (pretty much)



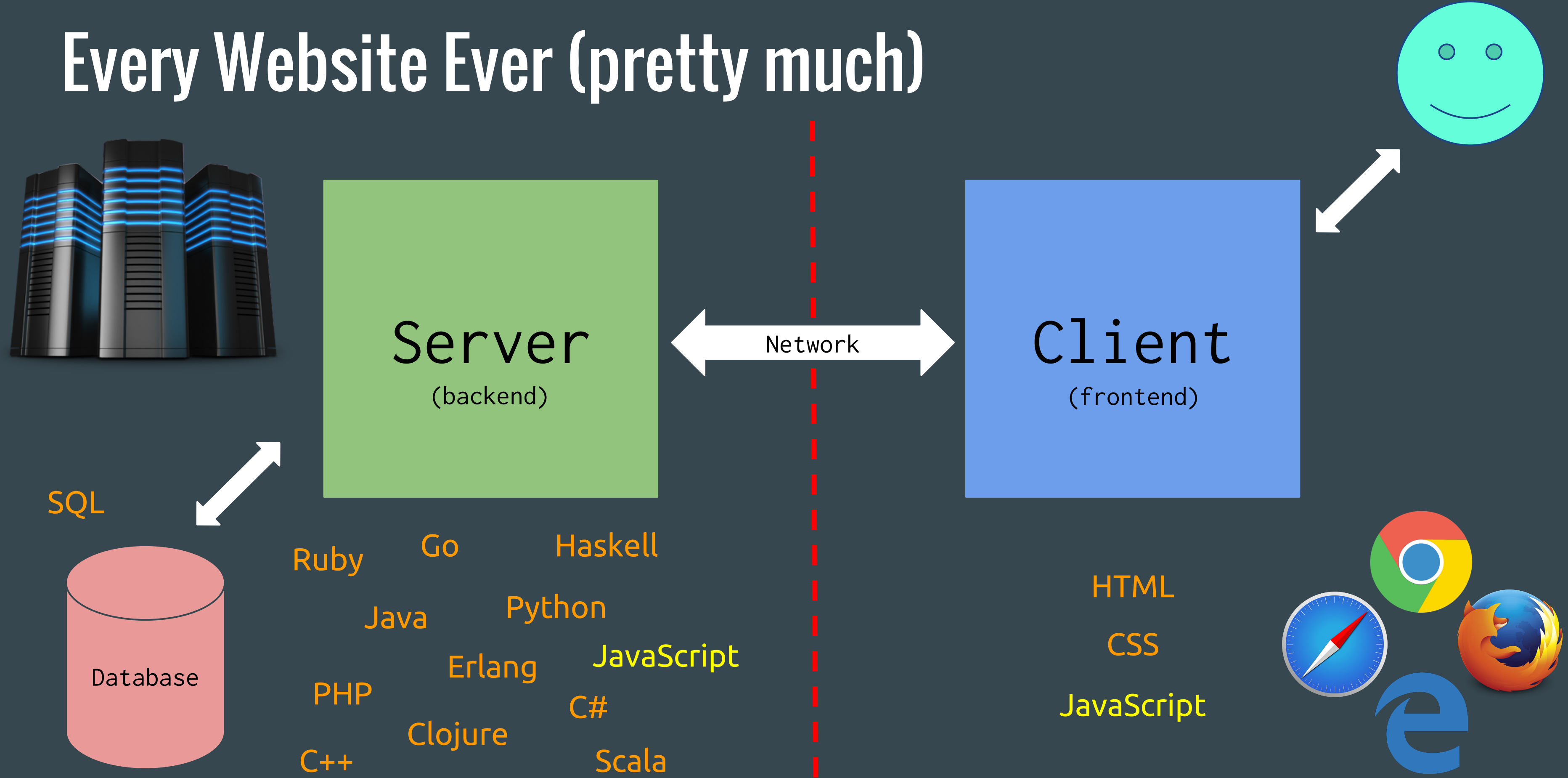
Every Website Ever (pretty much)



Every Website Ever (pretty much)



Every Website Ever (pretty much)



How a Website Gets Loaded

How a Website Gets Loaded

1. User instructs the browser to load a URL

How a Website Gets Loaded

1. User instructs the browser to load a URL
2. The browser requests the page from the server

How a Website Gets Loaded

1. User instructs the browser to load a URL
2. The browser requests the page from the server
3. The server returns some HTML to the browser

How a Website Gets Loaded

1. User instructs the browser to load a URL
2. The browser requests the page from the server
3. The server returns some HTML to the browser
4. The browser parses the HTML

How a Website Gets Loaded

1. User instructs the browser to load a URL
2. The browser requests the page from the server
3. The server returns some HTML to the browser
4. The browser parses the HTML
5. The browser constructs its own representation of the document (DOM)

How a Website Gets Loaded

1. User instructs the browser to load a URL
2. The browser requests the page from the server
3. The server returns some HTML to the browser
4. The browser parses the HTML
5. The browser constructs its own representation of the document (DOM)
6. If the HTML contains references to CSS or JavaScript, the browser fetches them

DOM Basics

DOM Basics

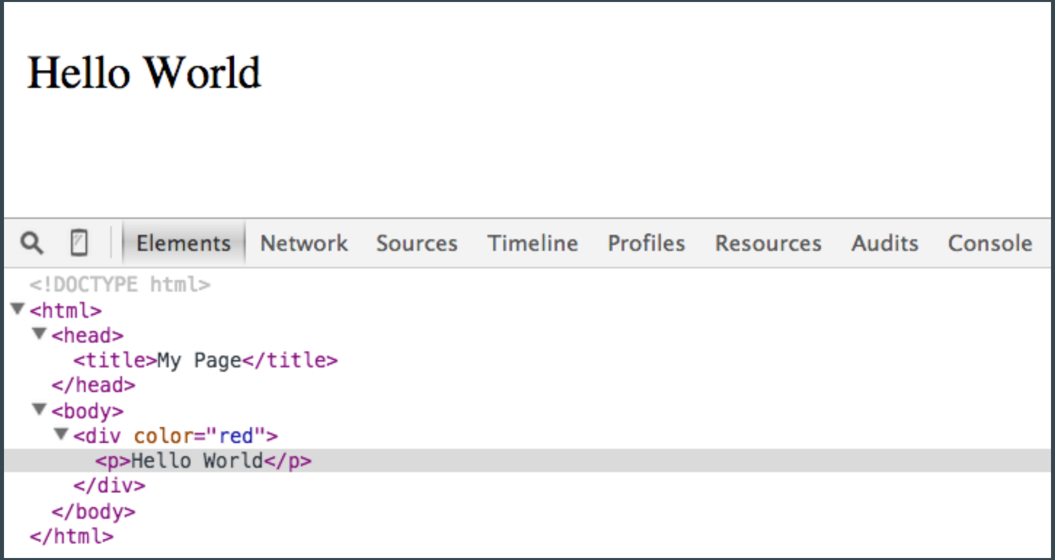
- DOM = Document Object Model

DOM Basics

- DOM = Document Object Model
- The browser’s representation of the HTML it was given

DOM Basics

- DOM = Document Object Model
- The browser’s representation of the HTML it was given



```
<!DOCTYPE html>
▼<html>
  ▼<head>
      <title>My Page</title>
    </head>
  ▼<body>
    ▼<div color="red">
        <p>Hello World</p>
      </div>
    </body>
</html>
```

Time for some JavaScript!

DOM Manipulation with JavaScript

DOM Manipulation with JavaScript

```
document.getElementById('one');
```

DOM Manipulation with JavaScript

```
document.getElementById('one');
```

object

DOM Manipulation with JavaScript

```
document.getElementById('one');
```

object

property access

DOM Manipulation with JavaScript

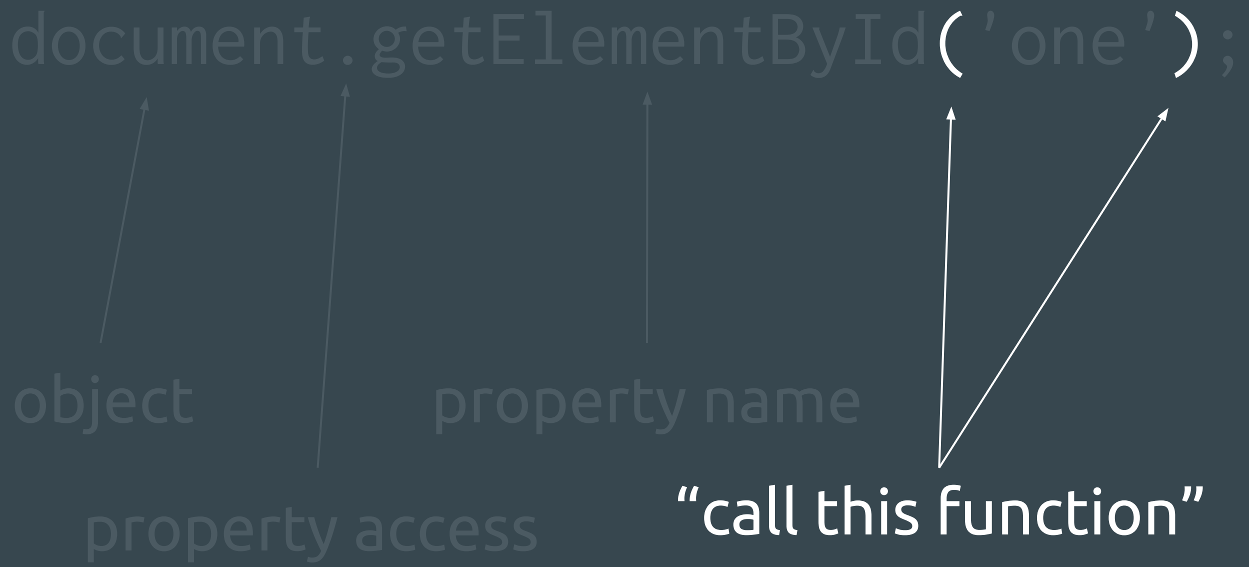
```
document.getElementById('one');
```

object

property name

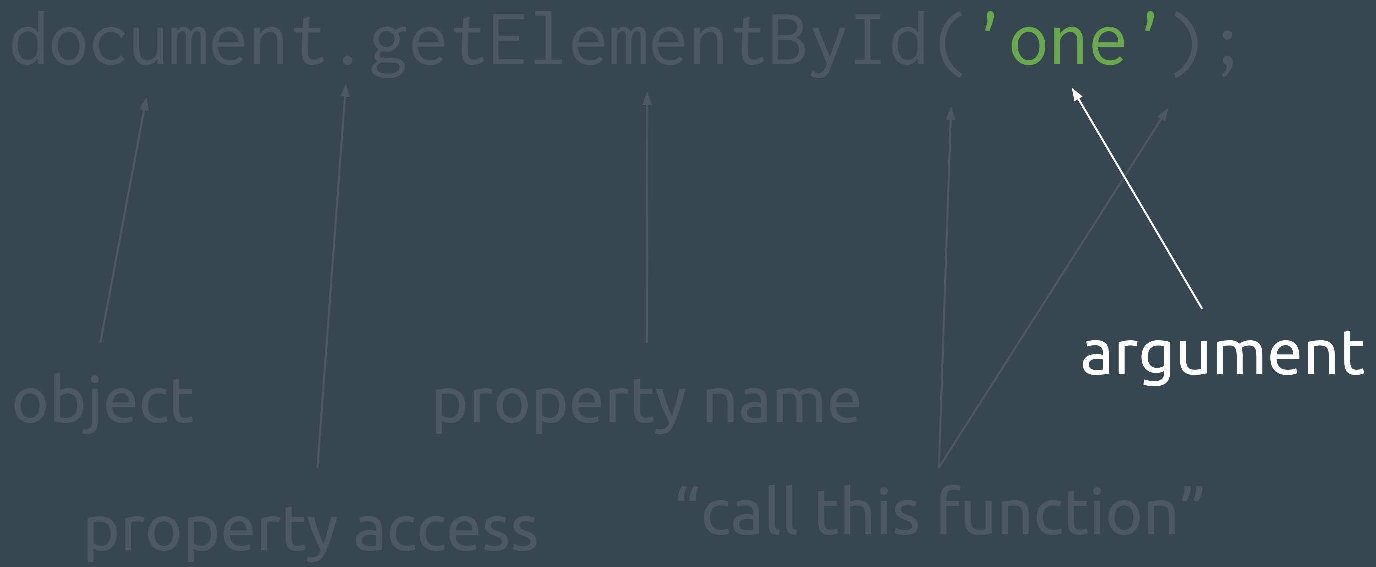
property access

DOM Manipulation with JavaScript



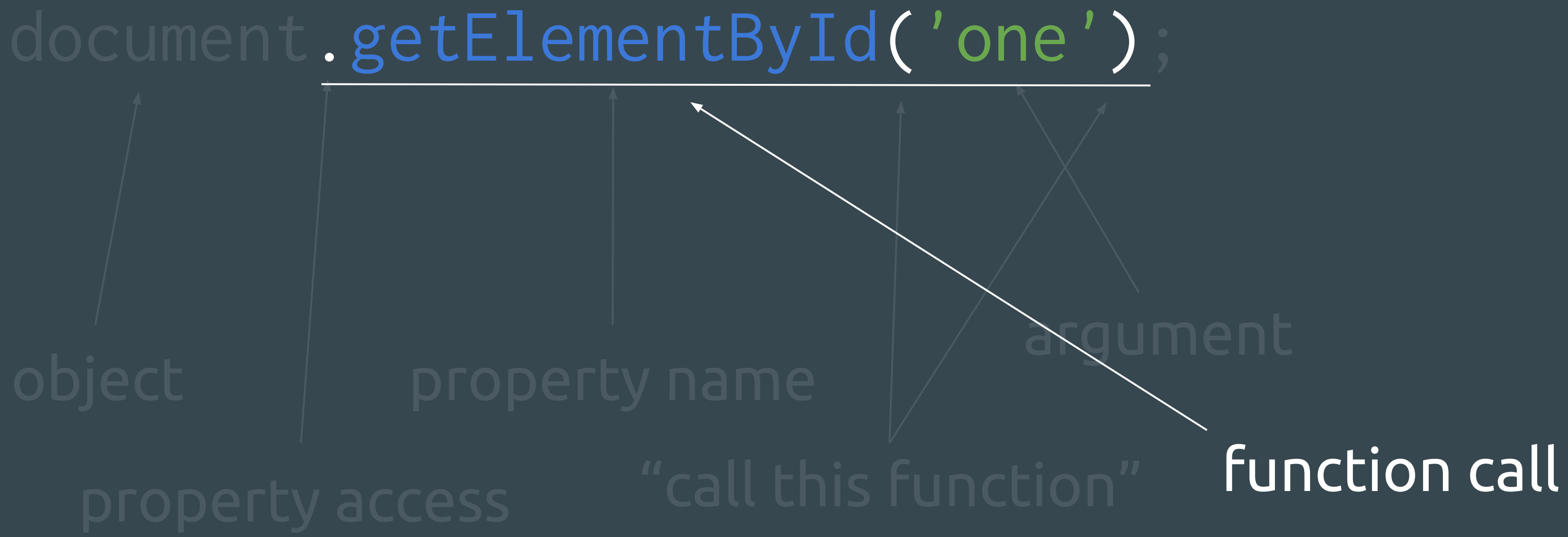
```
document.getElementById('one');
```

DOM Manipulation with JavaScript

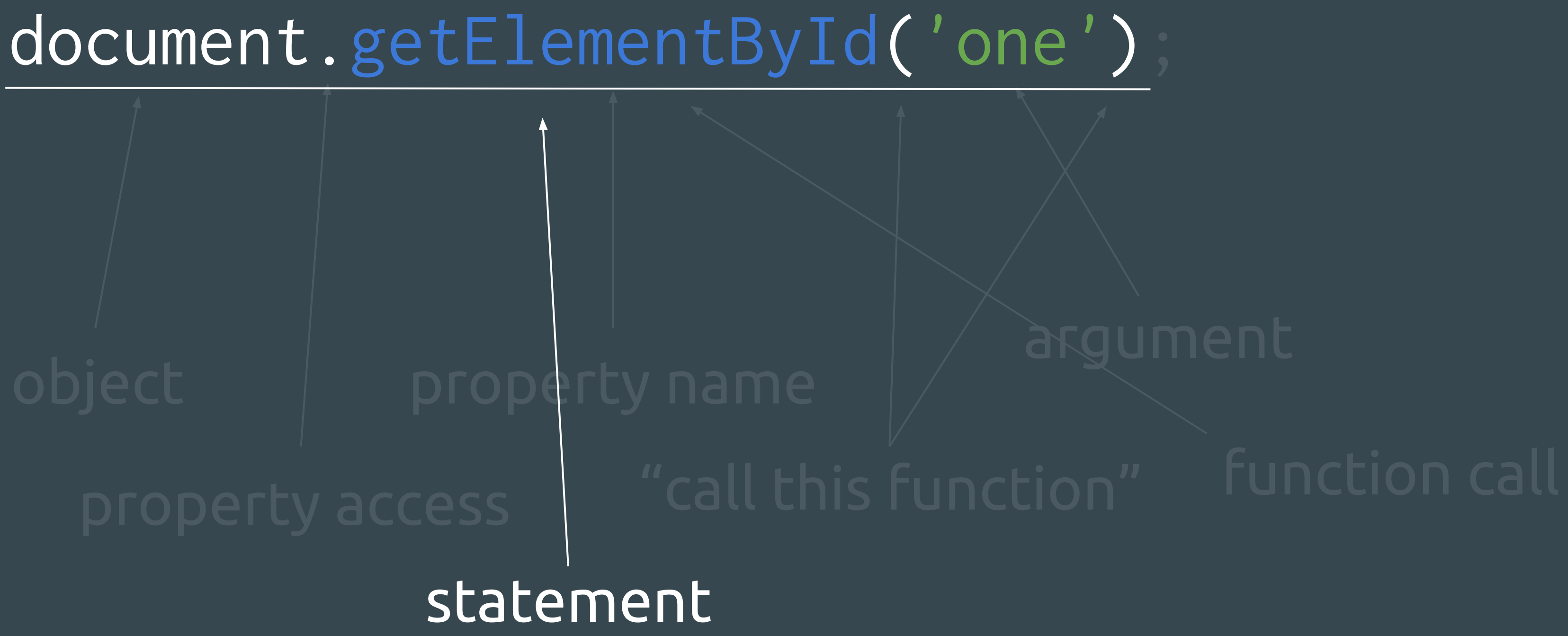


```
document.getElementById('one');
```

DOM Manipulation with JavaScript

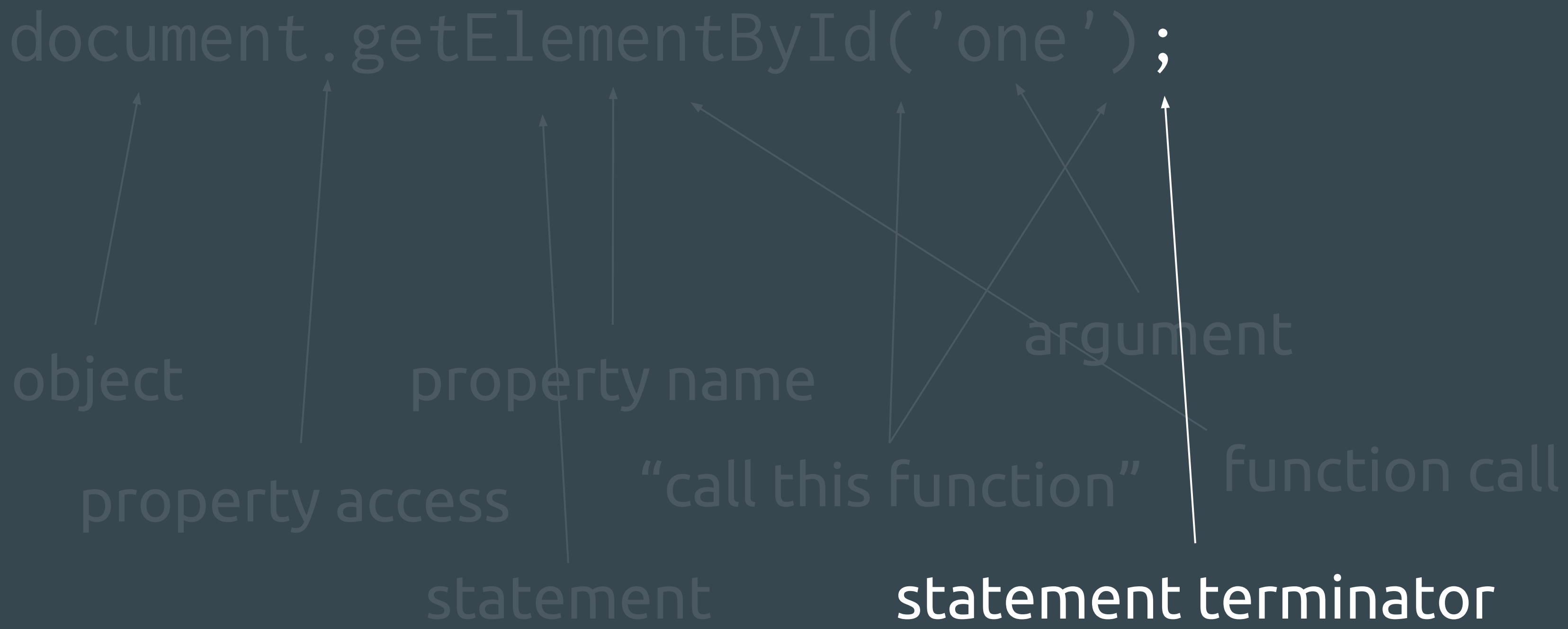

```
document.getElementById('one');
```

DOM Manipulation with JavaScript



```
document.getElementById('one');
```

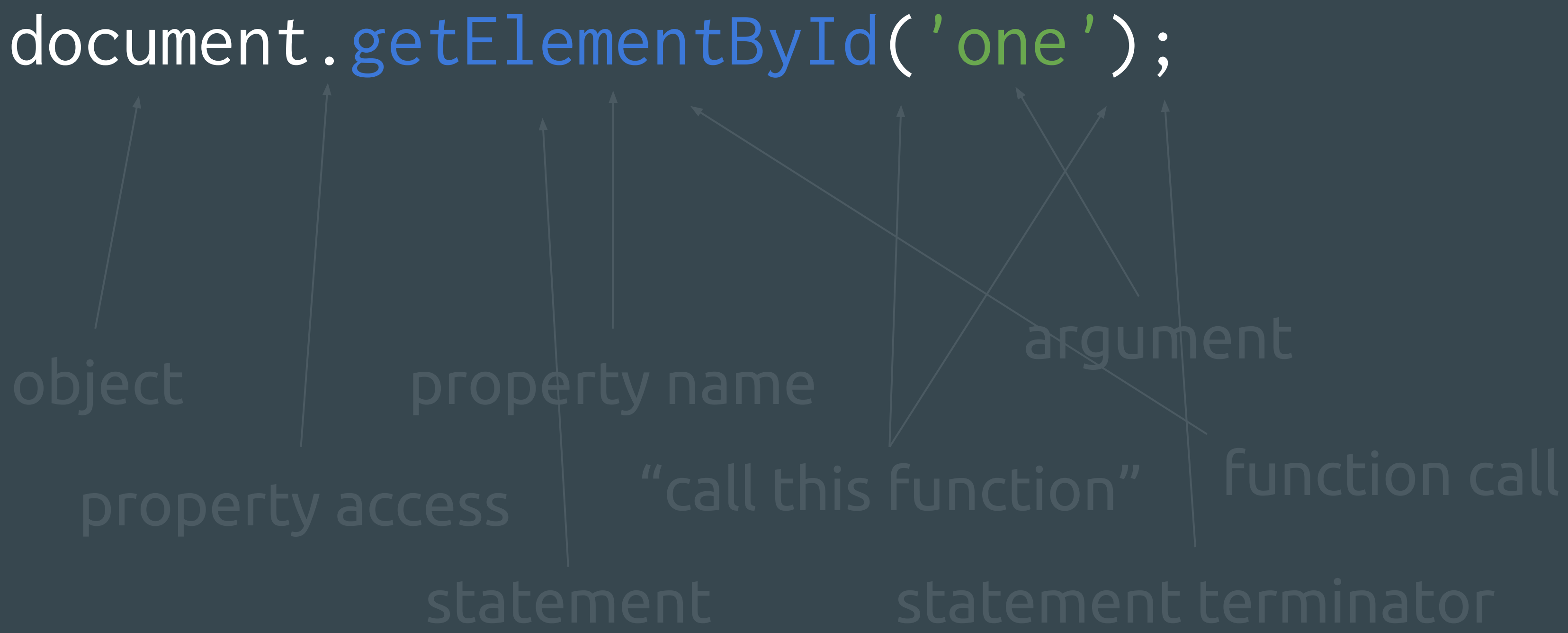
DOM Manipulation with JavaScript



```
document.getElementById('one');
```

DOM Manipulation with JavaScript

```
document.getElementById('one');
```



DOM Manipulation with JavaScript

```
document.getElementById('one');
```

DOM Manipulation with JavaScript

```
document.getElementById('one');
document.getElementsByClassName('fave');
document.getElementsByTagName('p');
```

DOM Manipulation with JavaScript

```
document.getElementById('one');
document.getElementsByClassName('fave');
document.getElementsByTagName('p');

document.querySelectorAll('#one');
document.querySelectorAll('.fave');
document.querySelectorAll('p');
```

Basic JavaScript - Variables and Data Types

Basic JavaScript - Variables and Data Types

```
var greeting = 'hello world';         // string
```

Basic JavaScript - Variables and Data Types

```
var greeting = 'hello world';         // string

var big = 99999;                      // number
var small = 0.0001;                   // number
```

Basic JavaScript - Variables and Data Types

```
var greeting = 'hello world';           // string

var big = 99999;                        // number
var small = 0.0001;                     // number

var yes = true;                         // boolean
var no = false;                         // boolean
```

Basic JavaScript - Variables and Data Types

```
var greeting = 'hello world';           // string

var big = 99999;                        // number
var small = 0.0001;                     // number

var yes = true;                         // boolean
var no = false;                         // boolean

var numbers = [1, 2, 3, 4, 5];          // array
var things = [big, small, yes, no]; // array
```

Basic JavaScript - Control Flow

Basic JavaScript - Control Flow

```
console.log('ten is greater than five!');
```

Basic JavaScript - Control Flow

```
if (10 > 5) {
  console.log('ten is greater than five!');
}
```

Basic JavaScript - Control Flow

```
if (10 > 5) {
  console.log('ten is greater than five!');
} else {

}
```

Basic JavaScript - Control Flow

```
if (10 > 5) {
  console.log('ten is greater than five!');
} else {
  console.log('uh...what?');
}
```

.

Basic JavaScript - Control Flow

Basic JavaScript - Control Flow

```
var i = 10;
```

Basic JavaScript - Control Flow

```
var i = 10;

while (i > 0) {


}
```

Basic JavaScript - Control Flow

```
var i = 10;

while (i > 0) {
  console.log(i);

}
```

Basic JavaScript - Control Flow

```
var i = 10;

while (i > 0) {
  console.log(i);
  i = i - 1;
}
```

Basic JavaScript - Control Flow

```
var i = 10;

while (i > 0) {
  console.log(i);
  i = i - 1;
}

console.log('BLAST OFF!');
```

Basic JavaScript - Control Flow

```
function blastOff() {
  var i = 10;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

Basic JavaScript - Control Flow

```
function blastOff() {
  var i = 10;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

```
blastOff();
```

Basic JavaScript - Control Flow

```
function blastOff(     ) {                  blastOff();
  var i = 10;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

Basic JavaScript - Control Flow

```
function blastOff(start) {
  var i = 10;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

```
blastOff();
```

Basic JavaScript - Control Flow

```
function blastOff(start) {                blastOff();
  var i = start;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

Basic JavaScript - Control Flow

```
function blastOff(start) {
  var i = start;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

```
blastOff(  );
```

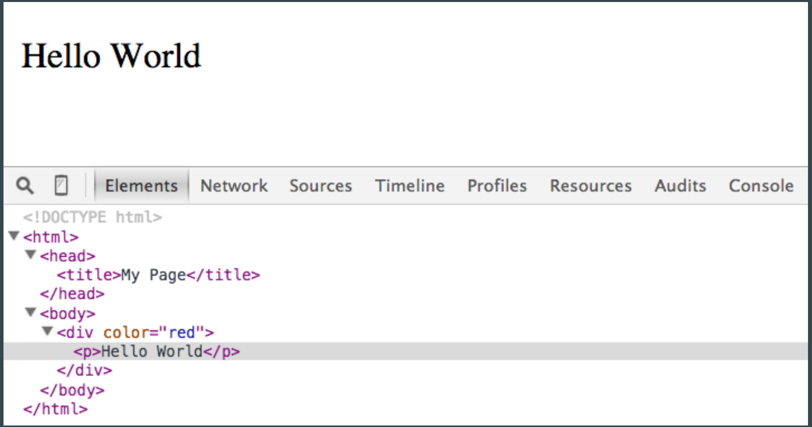
Basic JavaScript - Control Flow

```
function blastOff(start) {
  var i = start;

  while (i > 0) {
    console.log(i);
    i = i - 1;
  }

  console.log('BLAST OFF!');
}
```

```
blastOff(10);
```

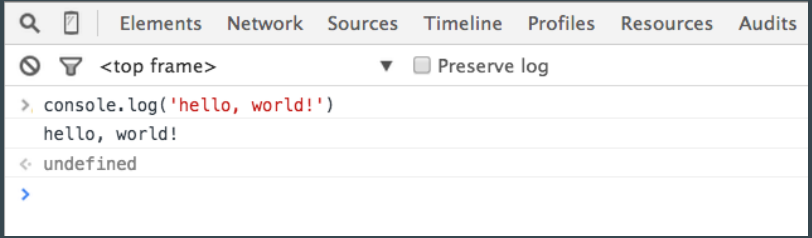
.

Chrome DevTools

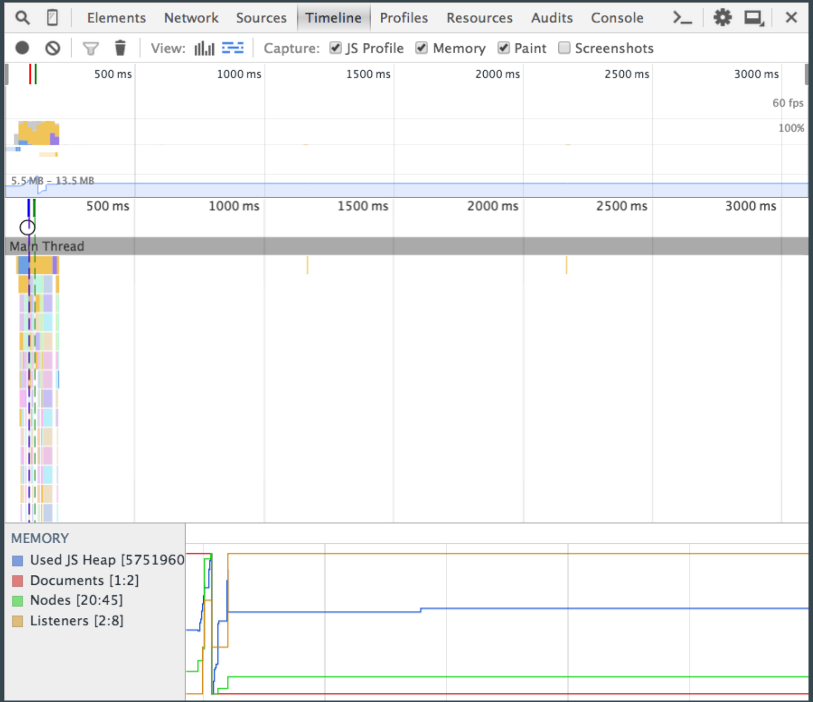


Hello World

```
<!DOCTYPE html>
▼<html>
  ▼<head>
      <title>My Page</title>
    </head>
  ▼<body>
    ▼<div color="red">
        <p>Hello World</p>
      </div>
    </body>
</html>
```



```
> console.log('hello, world!')
  hello, world!
<· undefined
>
```



DEMO

Basic JavaScript

http://js-intro.kevinjs.com

JavaScript Libraries

JavaScript Libraries

Library (noun)

1. a bunch of code someone else has written that others can use so that we aren't solving the same problems over and over again

jQuery

jQuery

- Popular JavaScript library

jQuery

- Popular JavaScript library
- Used on 71.6% of the top million websites (according to builtwith.com)

jQuery

- Popular JavaScript library
- Used on 71.6% of the top million websites (according to builtwith.com)
- Makes common DOM tasks easier

jQuery

- Popular JavaScript library
- Used on 71.6% of the top million websites (according to builtwith.com)
- Makes common DOM tasks easier
- Smooths over browser quirks

jQuery

- Popular JavaScript library
- Used on 71.6% of the top million websites (according to builtwith.com)
- Makes common DOM tasks easier
- Smooths over browser quirks
- A ton of other things

jQuery DOM Manipulation

jQuery DOM Manipulation

```
document.getElementById('one');
document.getElementsByClassName('fave');
document.getElementsByTagName('p');
```

jQuery DOM Manipulation

```
document.                   ('one');
document.                              ('fave');
document.                            ('p');
```

jQuery DOM Manipulation

```
document.                           ('one');
document.                           ('fave');
document.                           ('p');
```

jQuery DOM Manipulation

```
document.                        ('#one');
document.                        ('fave');
document.                        ('p');
```

jQuery DOM Manipulation

```
document.                         ('#one');
document.                         ('.fave');
document.                         ('p');
```

jQuery DOM Manipulation

```
document.querySelectorAll('#one');
document.querySelectorAll('.fave');
document.querySelectorAll('p');
```

jQuery DOM Manipulation

```
$('#one');
$('.fave');
$('p');
```

jQuery DOM Manipulation

```
$('#one');
$('.fave');
$('p');
```

jQuery DOM Manipulation

```
$('#one')
```

jQuery DOM Manipulation

```
$('#one').on(


 );
```

jQuery DOM Manipulation

```
$('#one').on('click'


 );
```

jQuery DOM Manipulation

```
$('#one').on('click', function () {


});
```

jQuery DOM Manipulation

```
$('#one').on('click', function () {
  console.log('one was clicked!');

});
```

jQuery DOM Manipulation

```
$('#one').on('click', function () {
  console.log('one was clicked!');
  $(this).css('color', 'blue');
});
```

Direct DOM Manipulation

```
$('#one').on('click', function () {
  console.log('one was clicked!');
  $(this).css('color', 'blue');
});
```

Direct DOM Manipulation

```
$('#one')                                .on('click', function () {
  console.log('one was clicked!');
  $(this).css('color', 'blue');
});
```

Direct DOM Manipulation

```
document.getElementById('one').on('click', function () {
  console.log('one was clicked!');
  $(this).css('color', 'blue');
});
```

Direct DOM Manipulation

```
document.getElementById('one').                ('click', function () {
  console.log('one was clicked!');
  $(this).css('color', 'blue');
});
```

Direct DOM Manipulation

```
document.getElementById('one').addEventListener('click', function () {
  console.log('one was clicked!');
  $(this).css('color', 'blue');
});
```

Direct DOM Manipulation

```
document.getElementById('one').addEventListener('click', function () {
  console.log('one was clicked!');
  this.style.color = 'blue';
});
```

JavaScript Frameworks

JavaScript Frameworks

- Bigger than a library

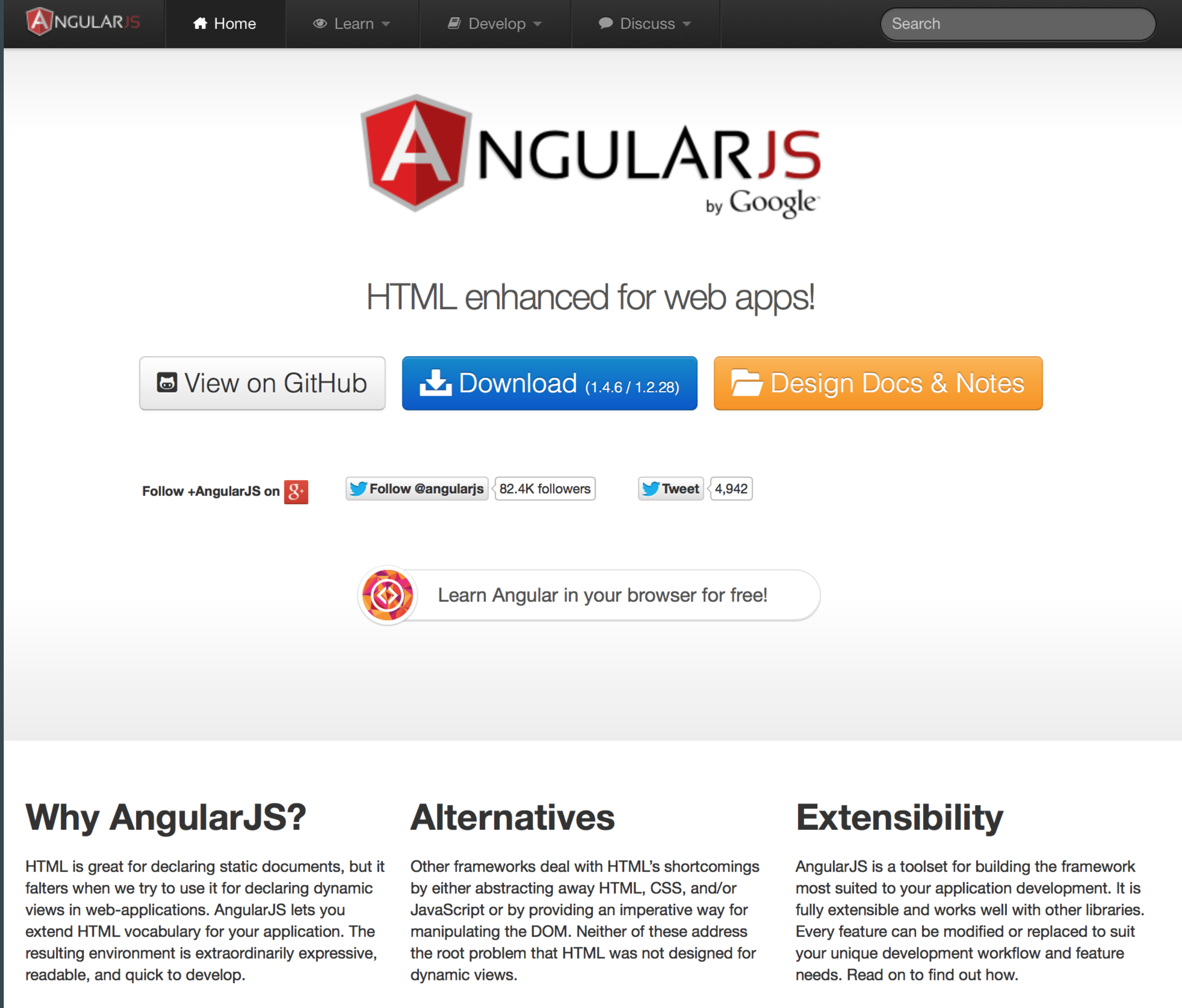
JavaScript Frameworks

- Bigger than a library
- A library gives you some tools to use in your code

JavaScript Frameworks

- Bigger than a library
- A library gives you some tools to use in your code
- A framework imposes **structure** on your code

AngularJS



HTML enhanced for web apps!

Why AngularJS?

HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS lets you extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop.

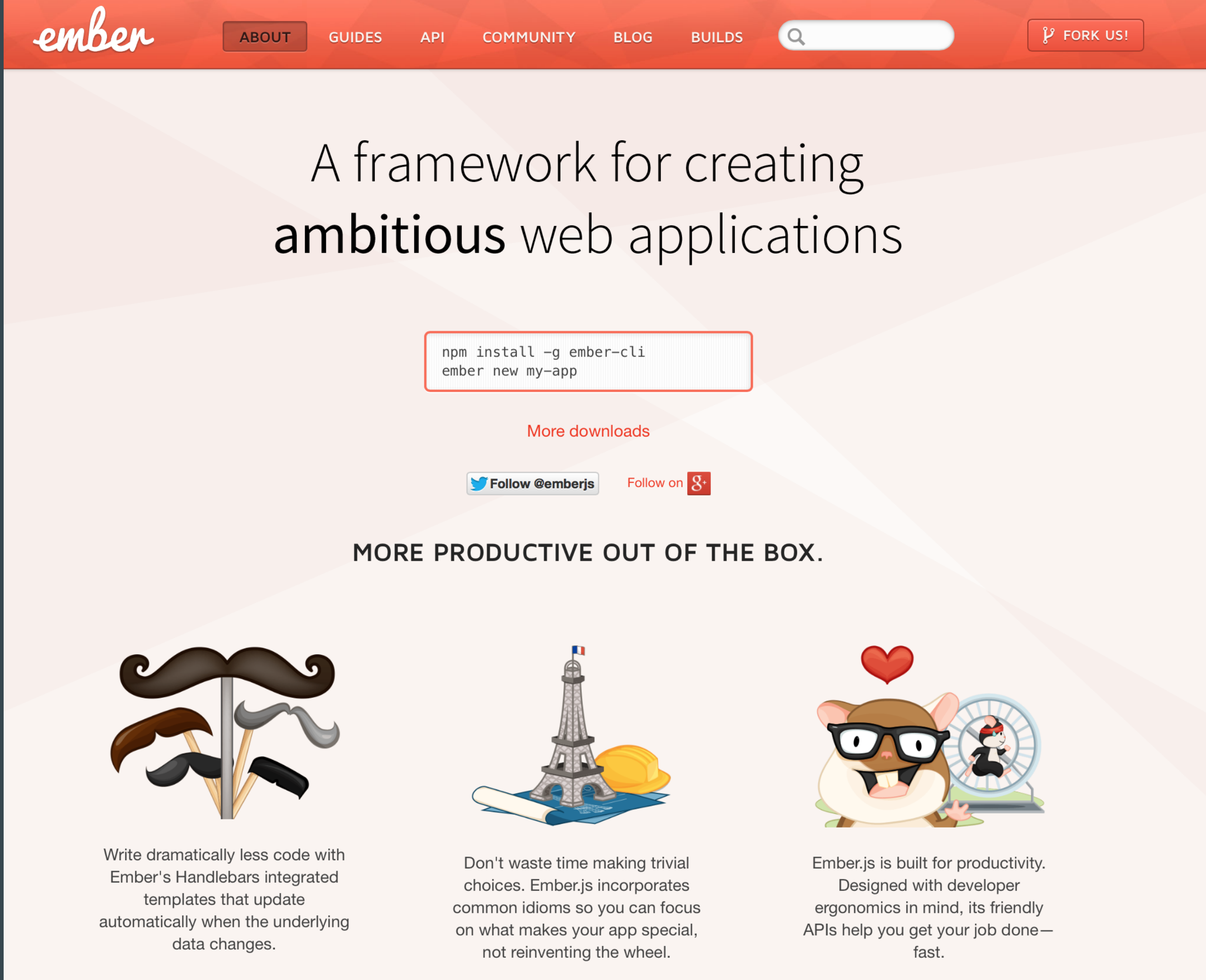
Alternatives

Other frameworks deal with HTML's shortcomings by either abstracting away HTML, CSS, and/or JavaScript or by providing an imperative way for manipulating the DOM. Neither of these address the root problem that HTML was not designed for dynamic views.

Extensibility

AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Read on to find out how.

Ember.js



A framework for creating **ambitious** web applications

```
npm install -g ember-cli
ember new my-app
```

More downloads

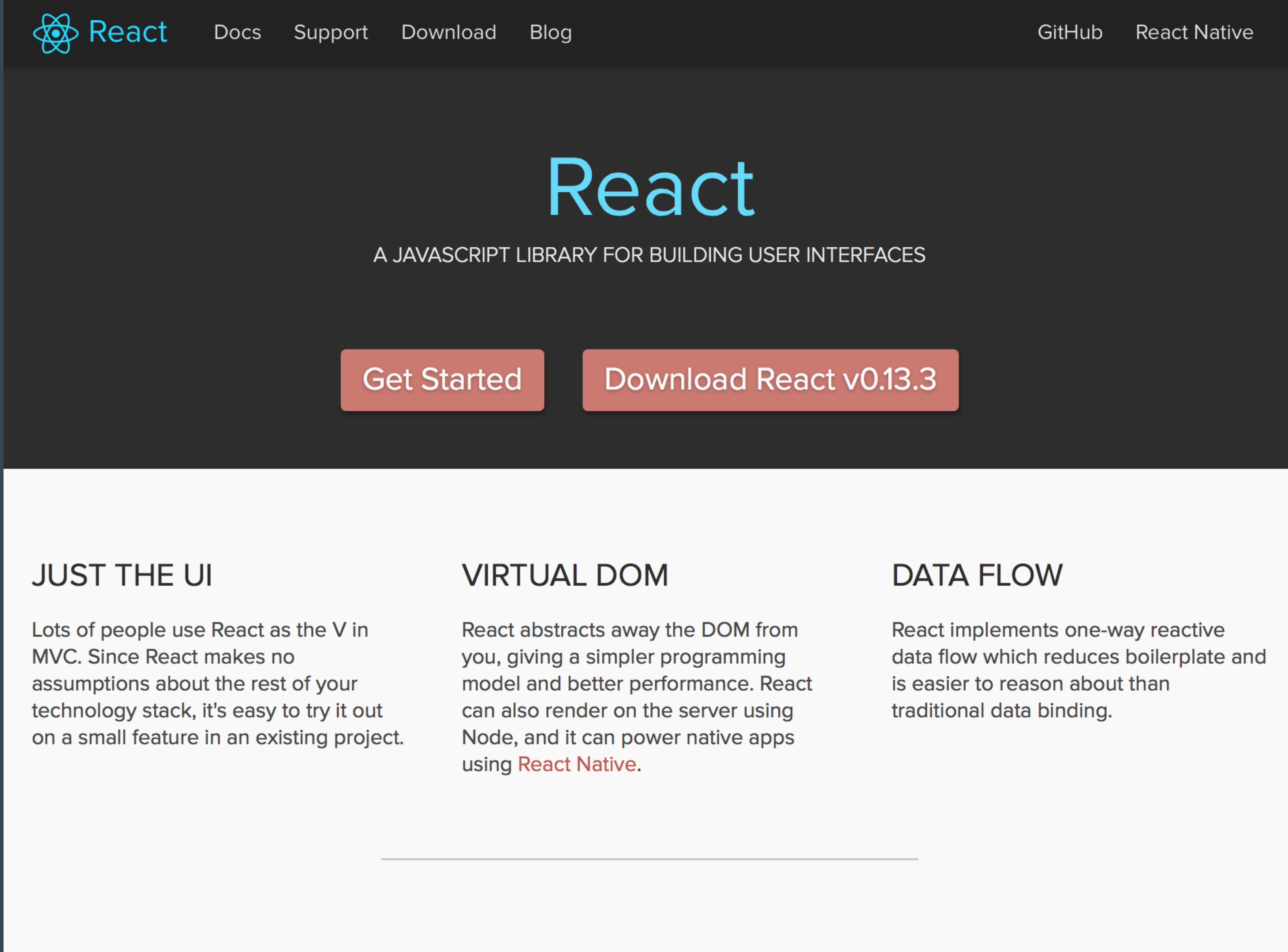
MORE PRODUCTIVE OUT OF THE BOX.

Write dramatically less code with Ember's Handlebars integrated templates that update automatically when the underlying data changes.

Don't waste time making trivial choices. Ember.js incorporates common idioms so you can focus on what makes your app special, not reinventing the wheel.

Ember.js is built for productivity. Designed with developer ergonomics in mind, its friendly APIs help you get your job done—fast.

React



React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

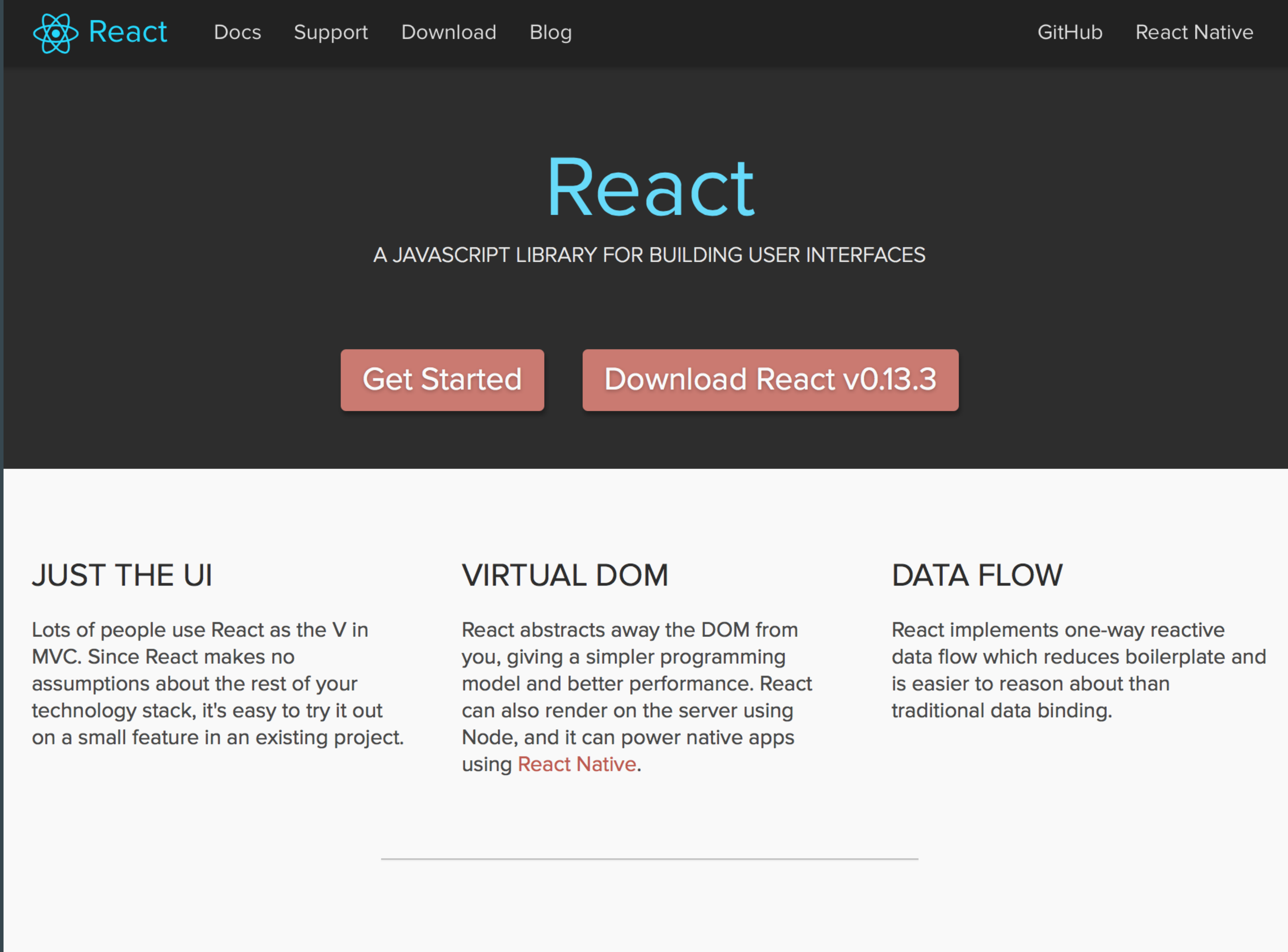
React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native.

DATA FLOW

React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

React

(technically not a "framework")



React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native.

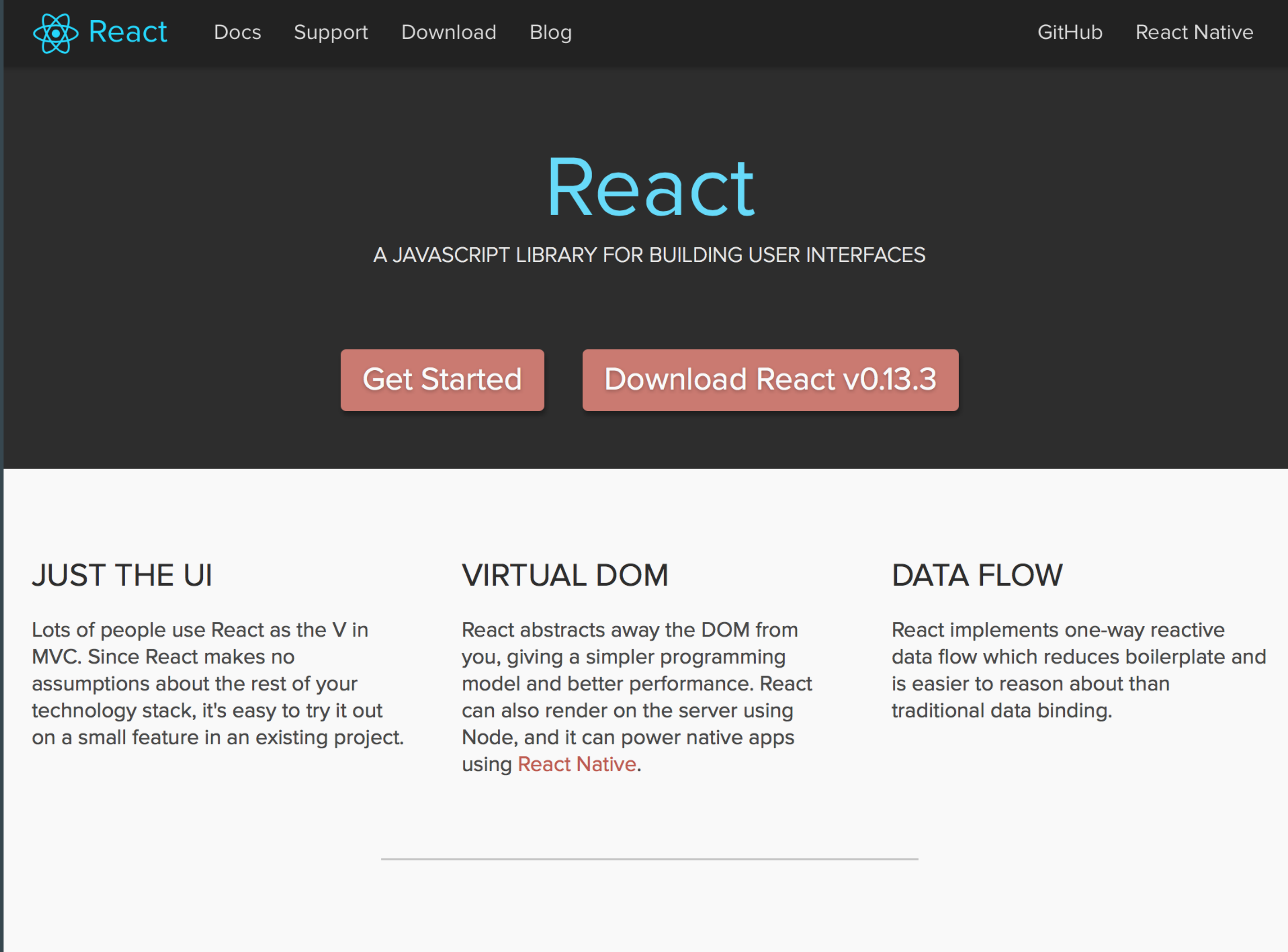
DATA FLOW

React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

React

(technically not a "framework")

(so it's really React PLUS a bunch of stuff)



React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native.

DATA FLOW

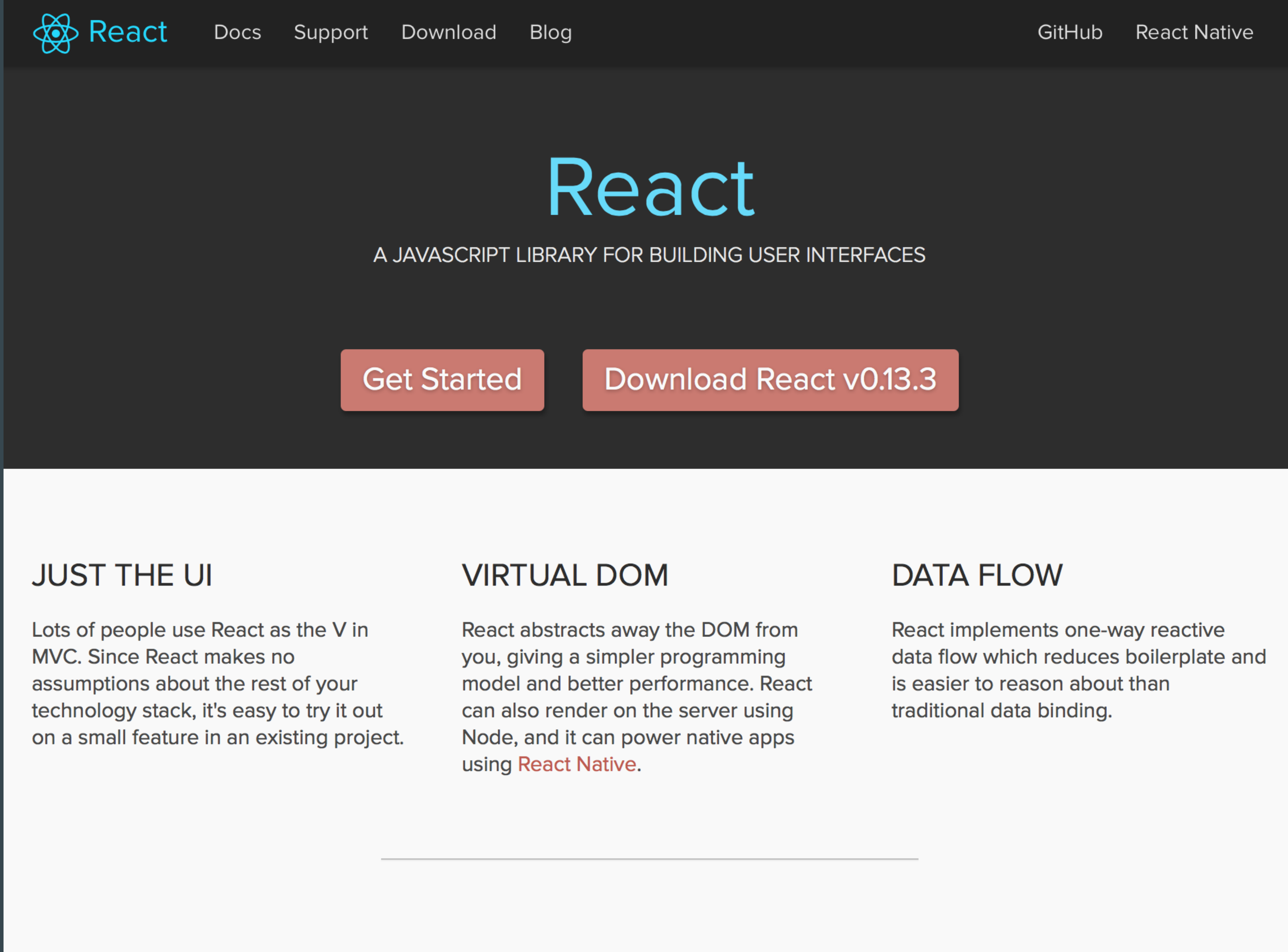
React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

React

(technically not a "framework")

(so it's really React PLUS a bunch of stuff)

(but it's still awesome)



React

A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

JUST THE UI

Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.

VIRTUAL DOM

React abstracts away the DOM from you, giving a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native.

DATA FLOW

React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.

Snake Demo

Snake Demo

- Create a new folder on your desktop called “Snake”

Snake Demo

- Create a new folder on your desktop called “Snake”
- Open the folder in Sublime Text

Snake Demo

- Create a new folder on your desktop called “Snake”
- Open the folder in Sublime Text
- In the Snake folder, create two files:
  - `index.html`
  - `snake.js`

Snake Demo

- Create a new folder on your desktop called “Snake”
- Open the folder in Sublime Text
- In the Snake folder, create two files:
  - `index.html`
  - `snake.js`
- Copy the HTML and JavaScript from my site into the appropriate files

Snake Demo

- Create a new folder on your desktop called “Snake”
- Open the folder in Sublime Text
- In the Snake folder, create two files:
    - `index.html`
    - `snake.js`
- Copy the HTML and JavaScript from my site into the appropriate files
- Open `index.html` in Chrome and play Snake!

Upcoming Tech Talent South Courses

- **Code Immersion**
    - 8 weeks, starts October 13th
    - Strongbox West
    - Full-time meets Monday - Thursday, 8am - 12:30pm
    - Part-time meets Mondays and Wednesdays, 6pm - 9pm

- **JavaScript 101**
    - 6 weeks, starts November 3rd
    - DeskHub (here!)
    - Meets Tuesdays, 6pm - 9pm
    - Taught by me :)

Upcoming Tech Talent South Courses

- Intro to Web Design and Creation
  - 8 weeks, starts January 5th
  - Strongbox West
  - Meets Mondays or Tuesdays, 6pm - 9pm
- More!
  - Check them out at techtalentsouth.com

THANK YOU!

Questions?

Kevin Smith

http://github.com/ksmithbaylor

ksmithbaylor@gmail.com