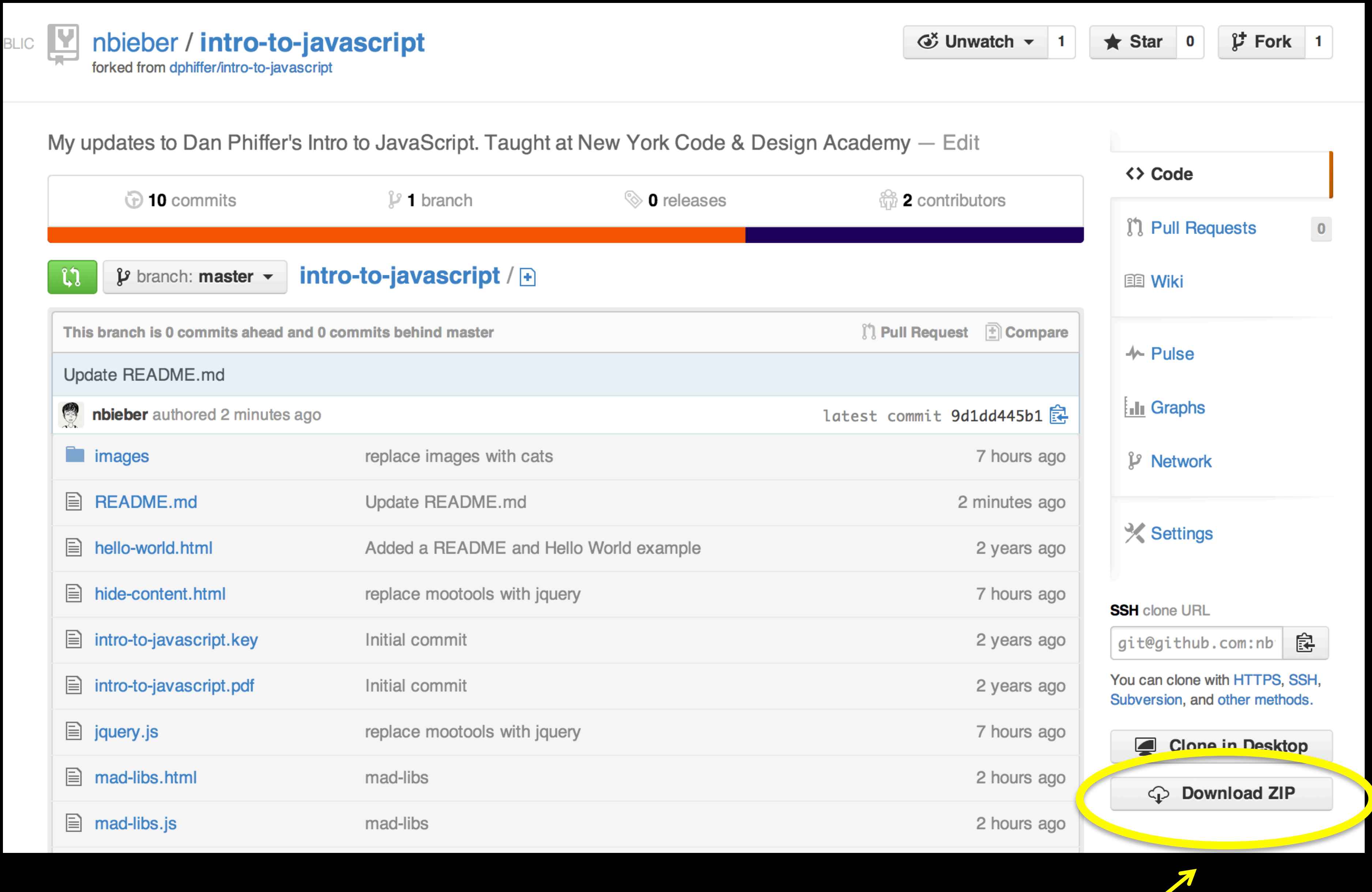
https://github.com/nbieber/intro-to-javascrip



nbieber / **intro-to-javascript**
forked from dphiffer/intro-to-javascript

My updates to Dan Phiffer's Intro to JavaScript. Taught at New York Code & Design Academy — Edit

10 commits · **1** branch · **0** releases · **2** contributors

branch: **master** · **intro-to-javascript** /

This branch is 0 commits ahead and 0 commits behind master · Pull Request · Compare

Update README.md

nbieber authored 2 minutes ago · latest commit 9d1dd445b1

images	replace images with cats	7 hours ago
README.md	Update README.md	2 minutes ago
hello-world.html	Added a README and Hello World example	2 years ago
hide-content.html	replace mootools with jquery	7 hours ago
intro-to-javascript.key	Initial commit	2 years ago
intro-to-javascript.pdf	Initial commit	2 years ago
jquery.js	replace mootools with jquery	7 hours ago
mad-libs.html	mad-libs	2 hours ago
mad-libs.js	mad-libs	2 hours ago

SSH clone URL

`git@github.com:nb`

You can clone with HTTPS, SSH, Subversion, and other methods.

Introduction to JavaScript

Nicole Bieber

(Slides adapted from Dan Phiffer's Intro to JavaScript)

roduction to Me

oftware Developer at Amplify

tudied computer science at MIT

ewest member of the NYCDA

eam

0

Why program computers?

Everything is deeply intertwingled. In an
mportant sense there are no “subjects” at all;
here is only all knowledge, since the cross-
connections among the myriad topics of this
world simply cannot be divided up neatly.”

—Ted Nelson, Computer Lib/Dream Machines

Vhen human beings acquired language, we earned not just how to listen but how to speak. Vhen we gained literacy, we learned not just how o read but how to write. And as we move into an ncreasingly digital reality, we must learn not just ow to use programs but how to make them.”

–Douglas Rushkoff, Program or Be Programmed

he single most significant change in the politics
cyberspace is the coming of age of this simple
ea: The code is law. The architectures of
berspace are as important as the law in defining
nd defeating the liberties of the Net.”

–Lawrence Lessig, The Code Is the Law

1
Learning a new language

Code is text

Programming is typing

Programming is very careful typing

Programming is fast typing

Programming is figuring out why it broke

Programming in general

- A series of text files that get compiled and executed
- Code is “digested,” going from human-readable to a hardware-ready form
- Ultimately programs run as assembly, low-level instructions for your CPU

JavaScript

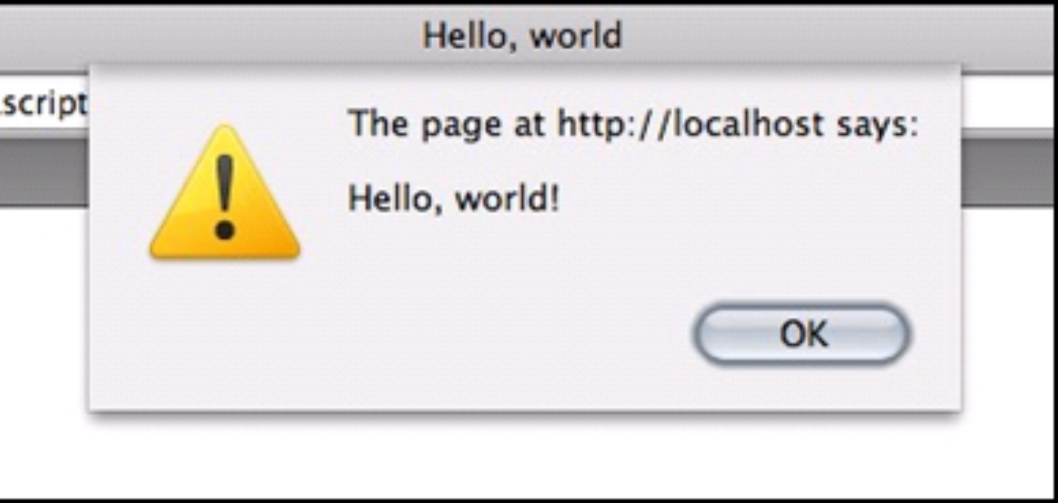
- Increasingly the web page scripting language
- Most likely the widest deployed runtime
- JavaScript has nothing to do with Java, except some syntax similarities

How is this different from Rails?

- Rails is server-side – it runs on the machine that is hosting your webpage
- JavaScript is client-side – it runs on the browser that is viewing your webpage
- JavaScript lets you mess with how thing look/behave on the page

A simple line of code

```
alert("Hello, world!");
```



The page at http://localhost says:

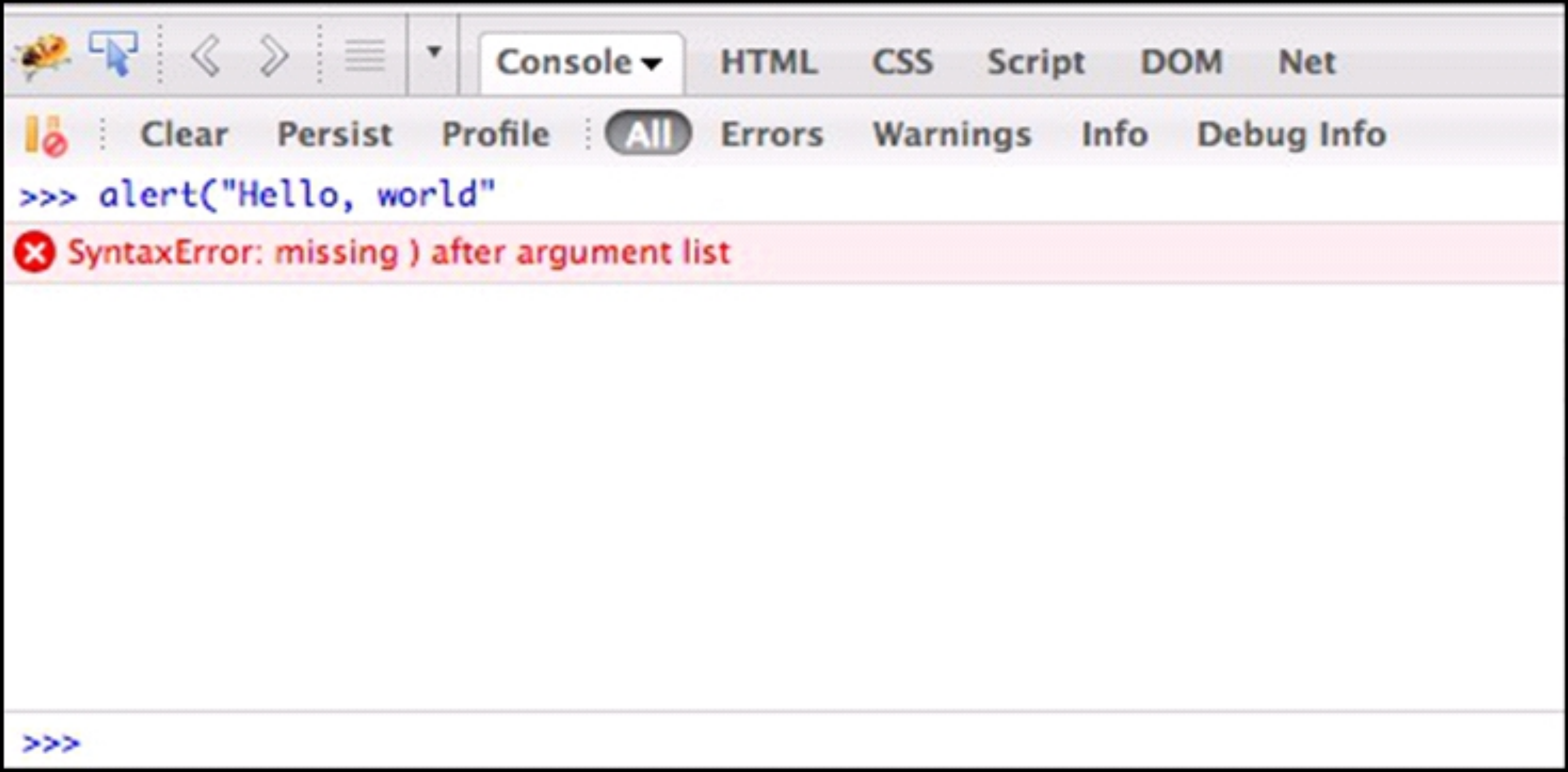
Hello, world!

Let’s try this using Chrome Developer Tools

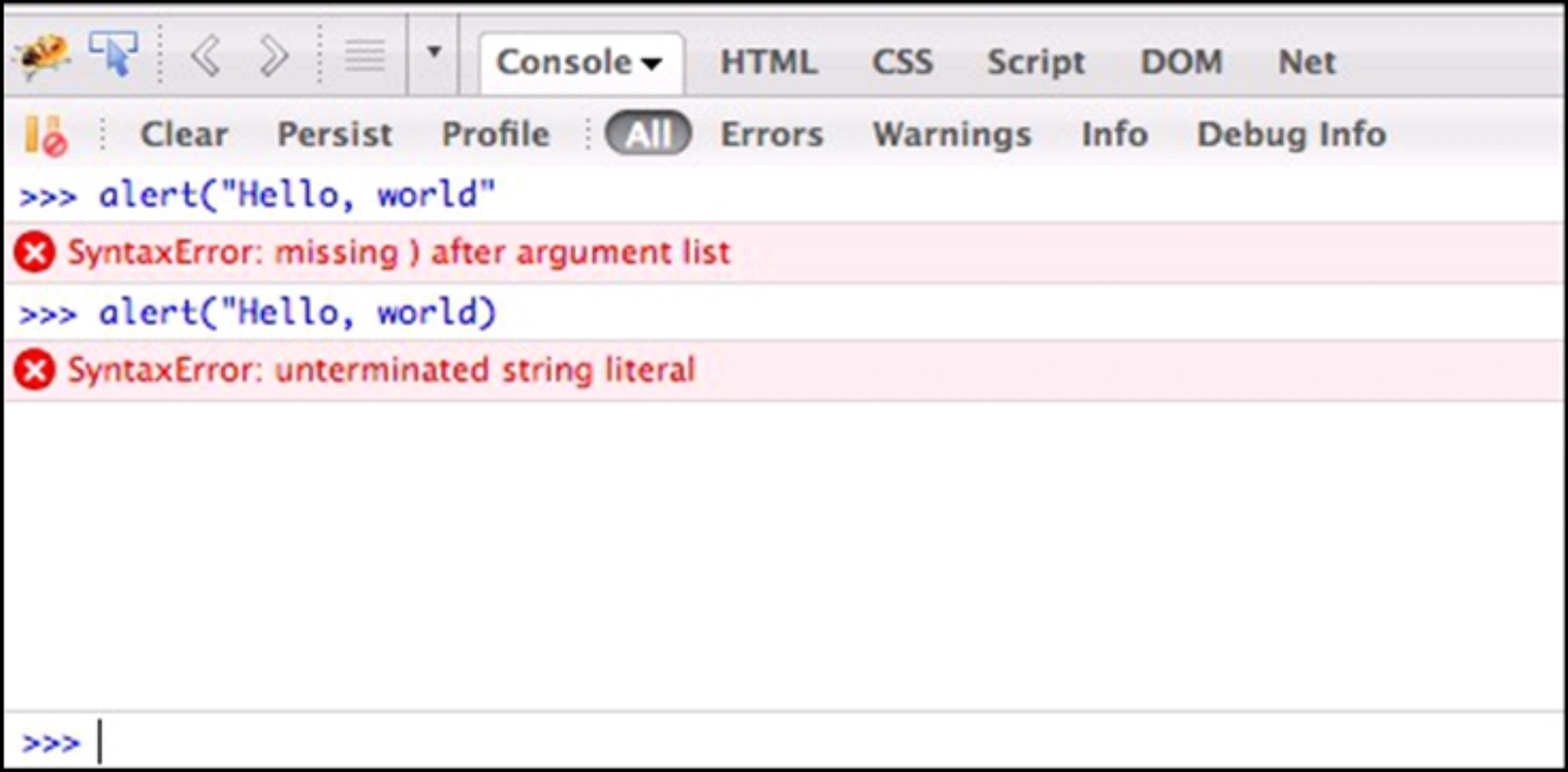
2
Writing code

Compilers are unforgiving

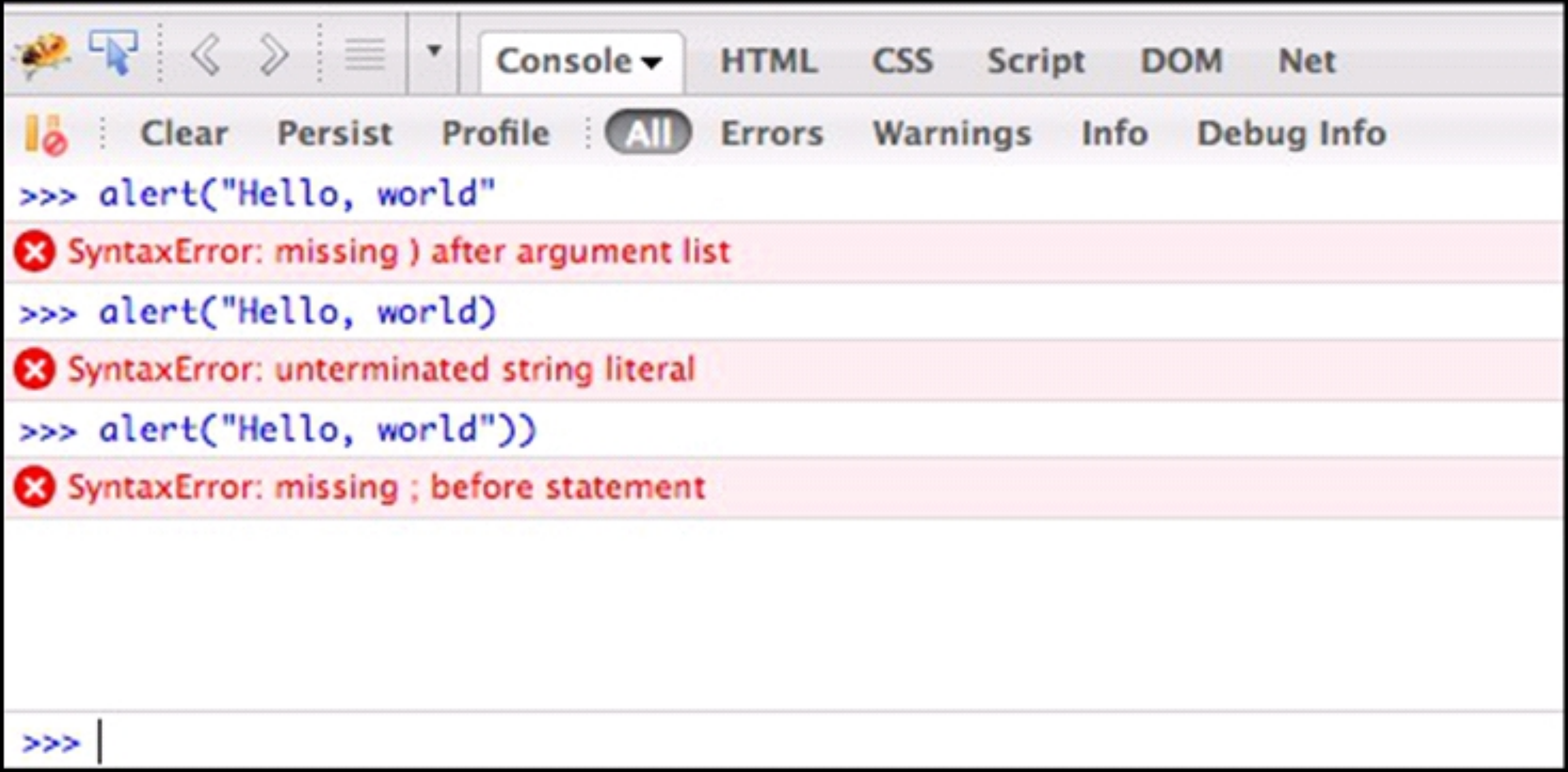
- The computer cuts you no slack
- All code is subject to bugs
- The error console is your friend
- Debugging is about identifying, characterizing, and resolving problems



```
>>> alert("Hello, world"
SyntaxError: missing ) after argument list
>>>
```



```
>>> alert("Hello, world"
SyntaxError: missing ) after argument list
>>> alert("Hello, world)
SyntaxError: unterminated string literal
>>>
```



```
>>> alert("Hello, world"
SyntaxError: missing ) after argument list
>>> alert("Hello, world)
SyntaxError: unterminated string literal
>>> alert("Hello, world"))
SyntaxError: missing ; before statement
>>>
```

A simple line of code

```
alert("Hello, world!");
```

on name

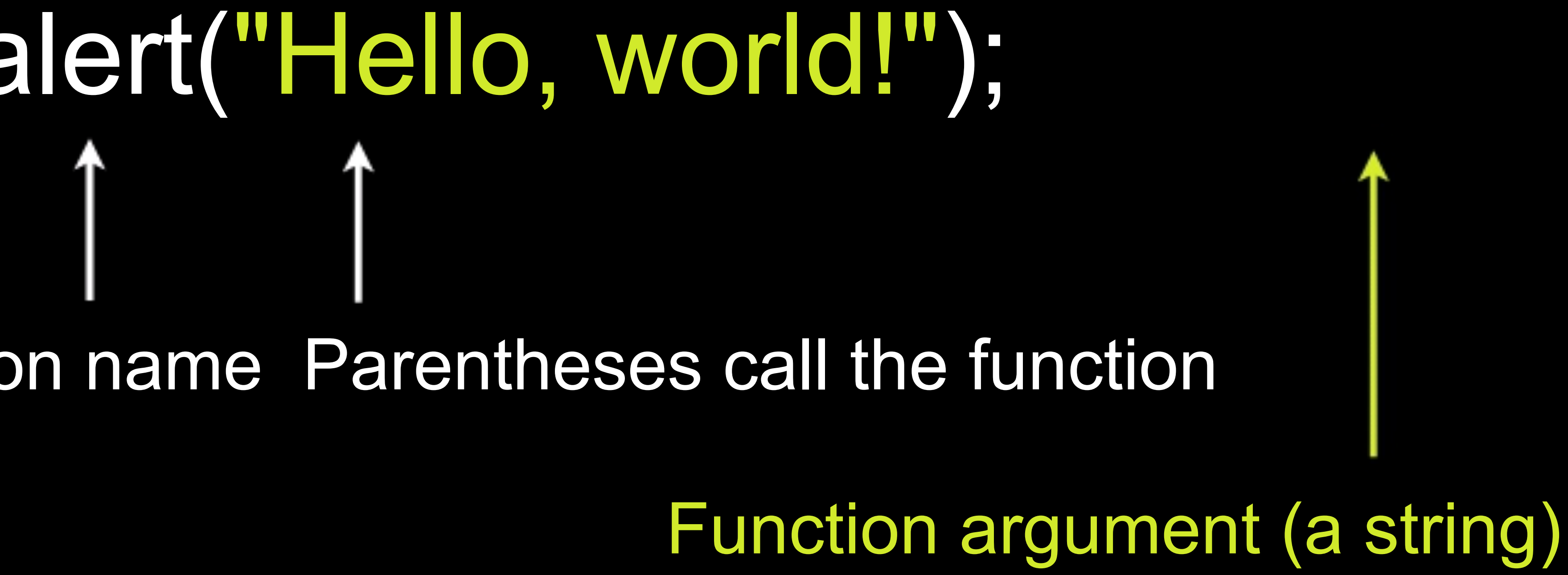
A simple line of code

```
alert("Hello, world!");
```

on name

Parentheses call the function

A simple line of code



```
alert("Hello, world!");
```

on name Parentheses call the function

Function argument (a string)

A simple line of code

```
alert("Hello, world!");
```

on name Parentheses call the function

Function argument (a string)

Designates the end of the

More simple functions

rompt(string);

confirm(string

3

Variables

The variable metaphor

“Variables are like a box
you can put data into.”

The variable metaphor

The variable metaphor

Variables

- Variables store data for future use
- var x = y; is how you assign a new variable in JavaScript
- We can now refer to x in future lines of code, and know it means y

Variables (boolean type)

- Variables store data for future use
- var x = true; is how you assign a new variable in JavaScript
- We can now refer to x in future lines of code, and know it means true

Variables (boolean type)

- Variables store data for future use
- var x = false; is how you assign a new variable in JavaScript
- We can now refer to x in future lines of code, and know it means false

Variables (numeric type)

- Variables store data for future use
- var x = 47; is how you assign a new variable in JavaScript
- We can now refer to x in future lines of code, and know it means 47

Variables (string type)

- Variables store data for future use
- var x = "pony"; is how you assign a new variable in JavaScript
- We can now refer to x in future lines of code, and know it means pony.

Variable logic

```
/ What is the value of z?
var x = 3;
var y = x + 1;
var z = y;
```

Try it out…

Create some variables in the console.

A variable a with a String in it
A variable b with a Number in it
A variable c with a Boolean in it

Now let’s try something crazy…

Dynamic Typing

Variables in JavaScript can be any type (just like n Ruby on Rails!)

Types of variables:

- undefined
- Null
- String
- Number
- Boolean
- Array
- Object
- Function

Whoa! They’re different!

4
Functions

Multiple lines of code

```
var msg = "Hello, world!";
var func = alert;
unc(msg);
```

Designate the ends of the l

Multiple lines of code

```
var msg = "Hello, world!";
var func = alert;
unc(msg);
```

The first line stores a string

Multiple lines of code

```
var msg = "Hello, world!";
var func = alert;
unc(msg);
```

The second line stores a function

Multiple lines of code

```
var msg = "Hello, world!";
var func = alert;
unc(msg);
```

The third line executes the stored function with the string

Commenting code

```
/ First we store the message
var msg = "Hello, world!";

/ Next, we choose a function to call
var func = alert;

/ Finally, we combine the two
unc(msg);
```

Commenting code

```
*

This code demonstrates the standard
Hello World program, over three lines
nstead of just one.

*/
var msg = "Hello, world!";
var func = alert;
unc(msg);
```

Creating a new function

```
/ Outputs a simple message
unction output_message() {
    var msg = "Hello, world!";
    var func = alert;
    func(msg);
```

Calling our function

```
/ Outputs a simple message
unction output_message() {
    var msg = "Hello, world!";
    var func = alert;
    func(msg);


output_message();
```

Arguments

```
/ Outputs a simple message
unction output_message(msg) {
    var func = alert;
    func(msg);


output_message("Hello, world!");
output_message("¡Hola, mundo!");
```

6
Objects

Creating Objects

```
var person=new Object();
person.firstname="John";
person.lastname="Doe";
person.age=50;
person.eyecolor="blue";
```

or

```
var person={firstname:"John",
            lastname:"Doe",
            age:50,
            eyecolor:"blue"};
```

JavaScript on the web

```
<script>

// JavaScript code is typically embedded in HTML
// <script> tags

</script>
```

HTML + JavaScript

```
<html>
  <head>
    <title>HTML + JavaScript</title>
  </head>
  <body>
    <p>Stuff *on* the page goes up here.</p>
    <script>

    // JavaScript code that modifies the page should
    // go below everything else in the <body>.

    </script>
  </body>
</html>
```

HTML + CSS + JavaScript

```
<html>
  <head>
    <title>HTML + CSS + JavaScript</title>
    <style>
    #content {
      background: #000;
    }
    </style>
  </head>
  <body>
    <p id="content">Hello, world!</p>
    <script>
      var content = document.getElementById('content');
      content.style.color = '#fff';
    </script>
  </body>
</html>
```

HTML + CSS + JavaScript

```
<html>
  <head>
    <title>HTML + CSS + JavaScript</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <p>
      Separating code into .js and .css files is a
      good way to keep things tidy.
    </p>
    <script src="scripts.js"></script>
  </body>
</html>
```

6
The DOM

Introducing the DOM…

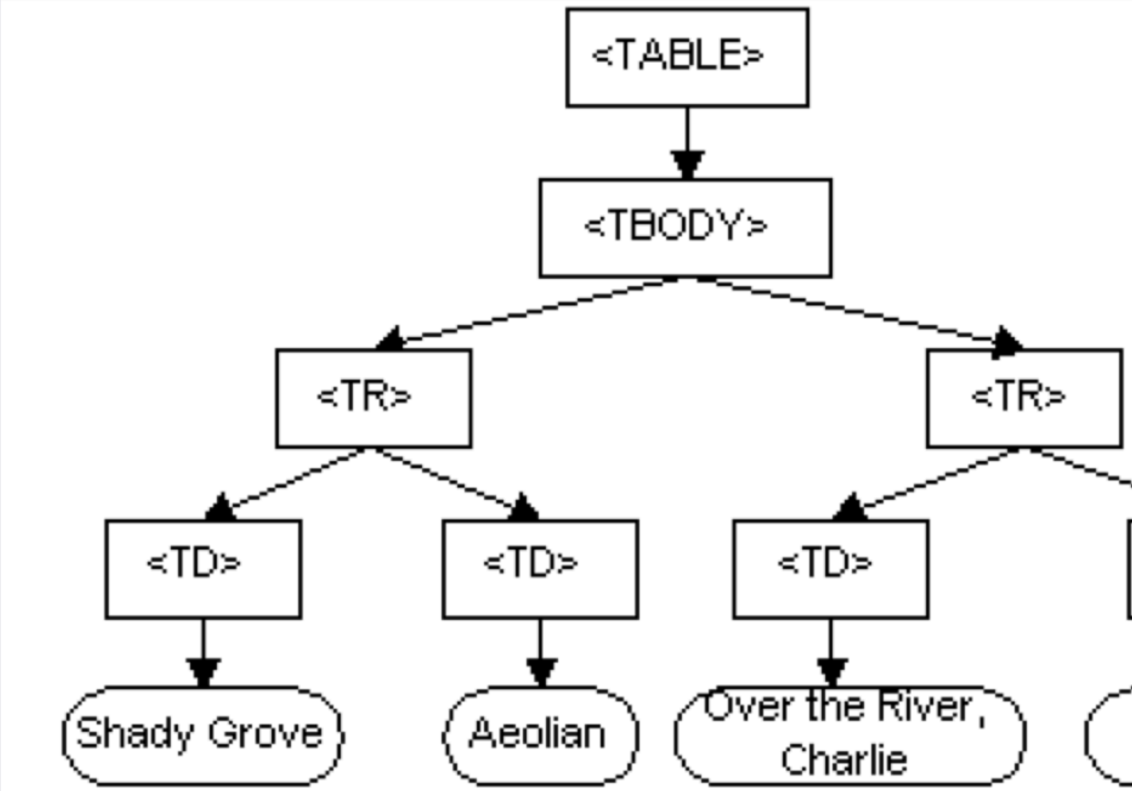
DOM = Document Object Model

```
Shady Grove</td>
Aeolian</td>


Over the River, Charlie</td>

Dorian</td>

>
```



Objects in the DOM

Objects exist for the document, the window, and everything in them

Some useful functions…

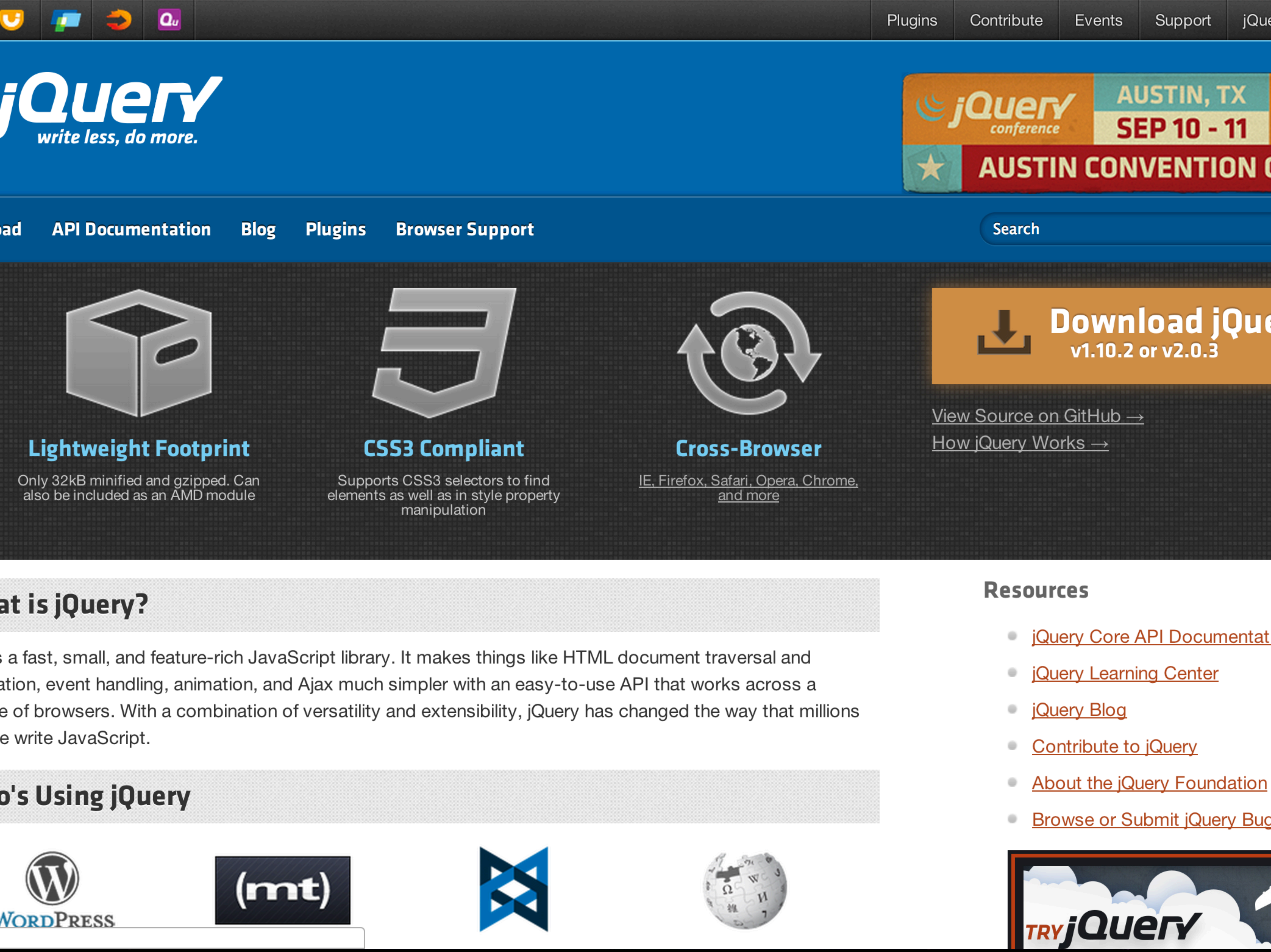
document.getElementById()

element.innerHTML

Find the full list at http://www.w3schools.com/
sref/dom_obj_document.asp)

7

Libraries



Lightweight Footprint

Only 32kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser

IE, Firefox, Safari, Opera, Chrome, and more

View Source on GitHub →

How jQuery Works →

at is jQuery?

s a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and ation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a e of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions e write JavaScript.

o's Using jQuery

Resources

- jQuery Core API Documentat
- jQuery Learning Center
- jQuery Blog
- Contribute to jQuery
- About the jQuery Foundation
- Browse or Submit jQuery Bug

Hide content

```
<html>
  <head>
      <title>Hide content</title>
  </head>
  <body>
      <p id="hide">Click to hide me!</p>
      <script src=”jquery.js"></script>
      <script>
      $('#hide').click(function () {
          $('#hide').fadeOut("slow");
      });
      </script>
  </body>
</html>
```

7

Slide show

Slide show HTML

```
<html>
  <head>
    <title>Slide show</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <div id="slides">
      <div id="inner">
        <img src="images/1.jpg" />
        <img src="images/2.jpg" />
        <img src="images/3.jpg" />
        <img src="images/4.jpg" />
      </div>
    </div>
    <script src=”jquery.js"></script>
    <script src="script.js"></script>
  </body>
</html>
```

Slide show CSS

```
#slides {
  width: 991px;
  height: 671px;
  margin: 0 auto;
  overflow: hidden;
  position: relative;


#inner {
  position: absolute;
  left: 0;
  top: 0;


#slides img {
  float: left;
```

Slide show JavaScript

8
What next?

Codecademy

Eloquent JavaScrip

A modern introduction
to programming

Marijn Haverbeke

Resources

- Eloquent JavaScript
- w3schools.com
- Mozilla devmo
- WebMonkey
- The Rhino Book

- Visual Quickstart Guide
- Lynda tutorials
- JavaScript: The Good P
- Codecademy