

Building Startup Systems





Agenda

No Class Wed Sep 13th. Class extended to Sep 27th.

- Review HW 02
- The Browser
- CSS
- JavaScript
- HW 03



Assignment 02 Recap

- Circle CI setup
- Create a RESTful API
- Write a Test
- Deploy in the Cloud

00. The Goal

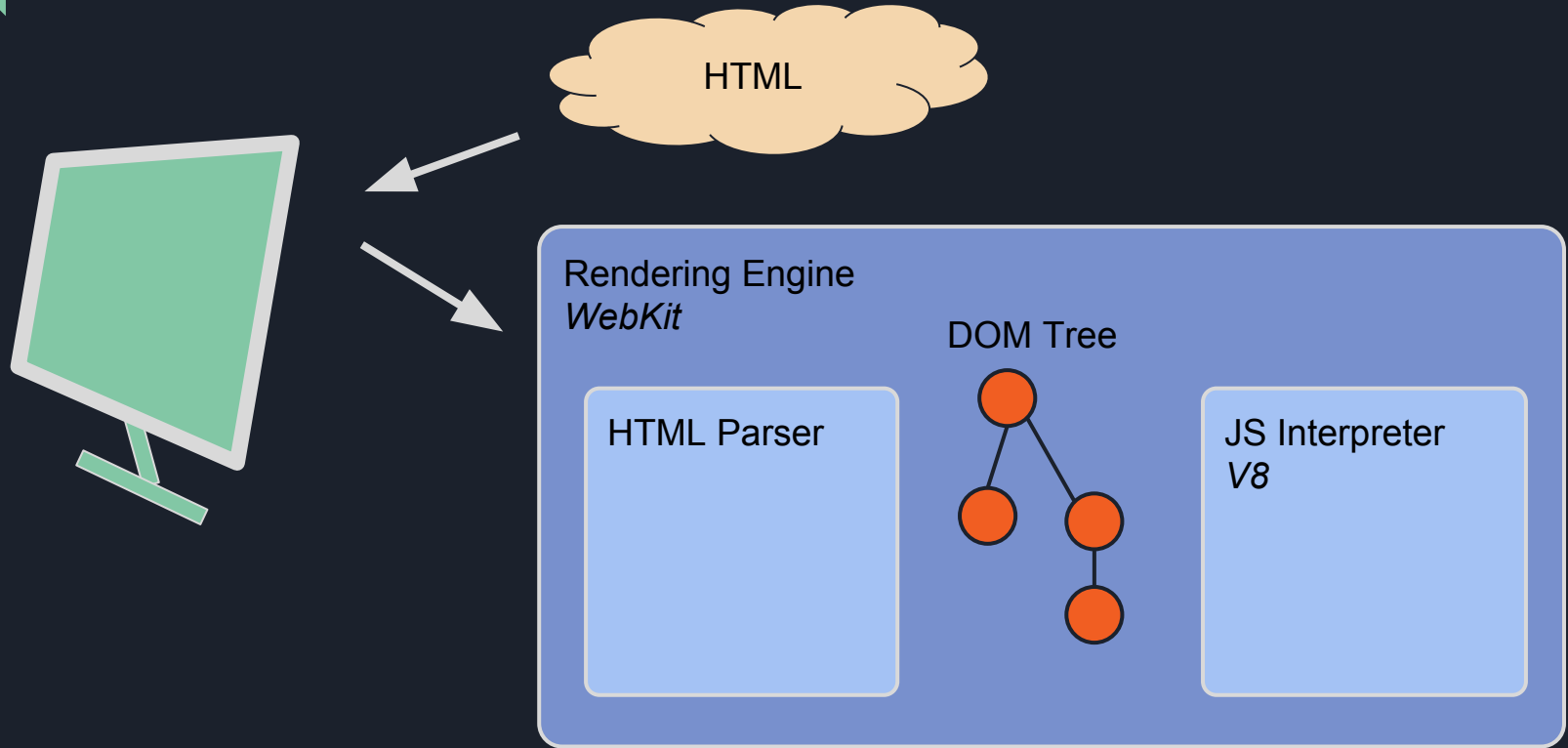




01. The Browser



Your Browser is an Interpreter

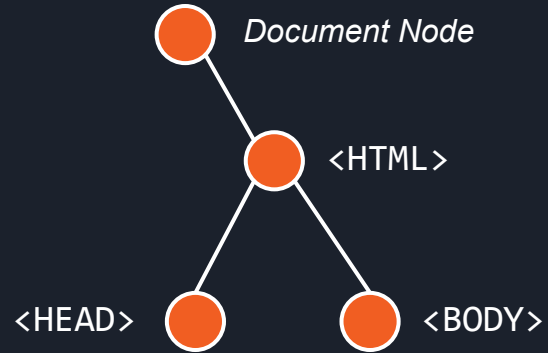


HTML as a Tree

... with XML-ish Syntax

```
<HTML>  
  <HEAD></HEAD>  
  <BODY>  
  
    </BODY>  
</HTML>
```

DOM Tree





Important DOM Tree Properties

- *DOM* is to the *Browser* as an *AST* is to the *Parser*
- *Elements* in *HTML* are *Nodes* in the *DOM Tree*
- *DOM Nodes* have parents and children
- When rendered, think of *Nodes* visually as a rectangle
- Child elements when rendered appear nested visually
- The *DOM* is useful for rendering, applying *CSS* to, and scripting against

<https://goo.gl/UjGFkx>



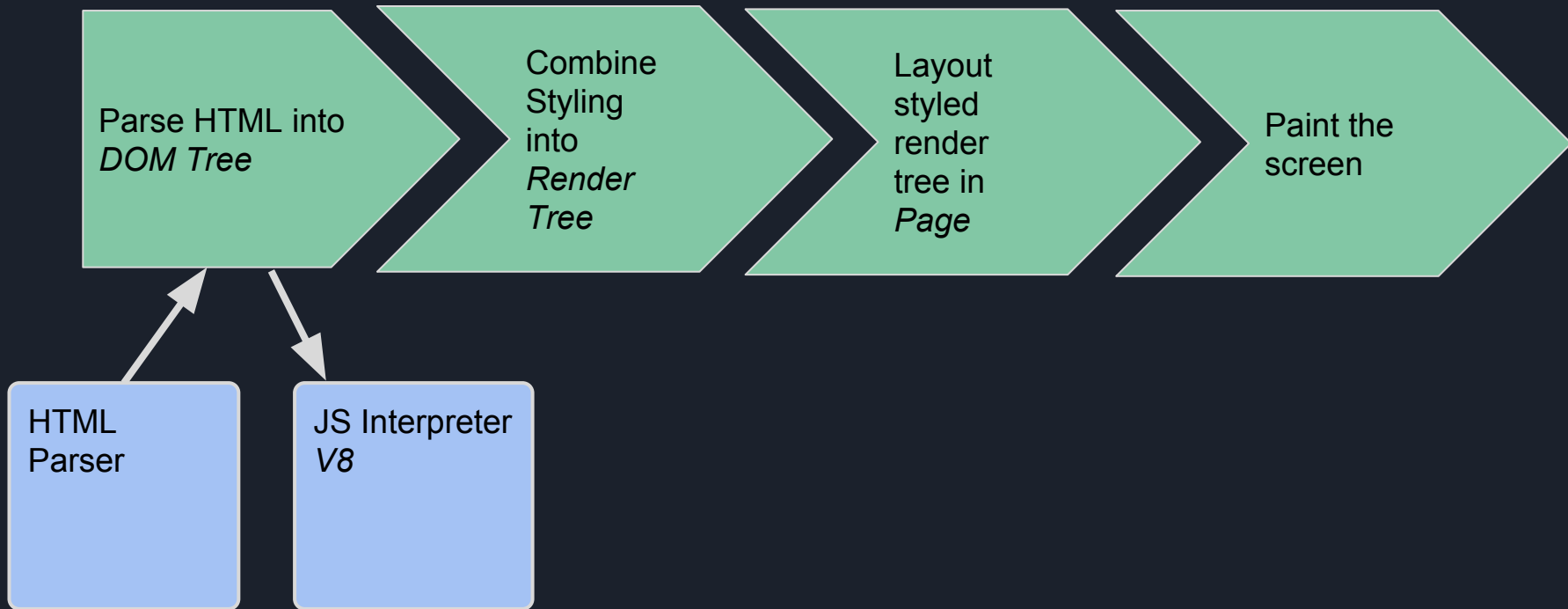
Some Elements are Special

But probably not as many as you think

- IMG
- INPUT
- VIDEO
- BUTTON

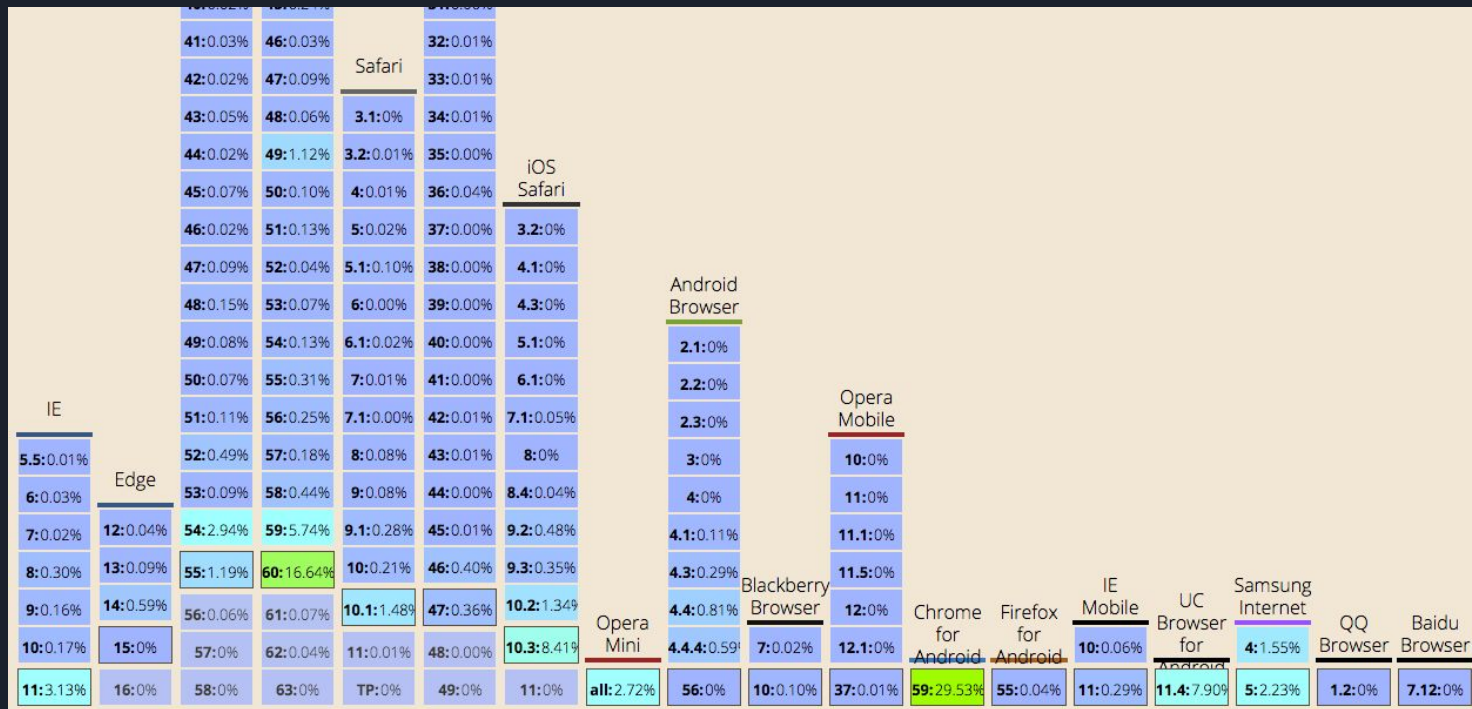


Rendering Pipeline



Writing Browser Programs is Hard

<https://caniuse.com/>



03. CSS





CSS is the Language for Styling HTML

<https://goo.gl/jvvkVN>

<https://goo.gl/phKJGQ>

<https://goo.gl/fdJooR>



CSS: Selectors and Properties

```
* { ... }
```

match all elements

```
DIV, H1, H2 { ... }
```

match elements by-name

```
DIV P { ... }
```

match nested elements

```
#foo {}
```

matches elements by-id

```
.foo {}
```

matches elements by-class

CSS: Properties

Properties control visibility, positioning, borders, shape, styling, and even layout-mode

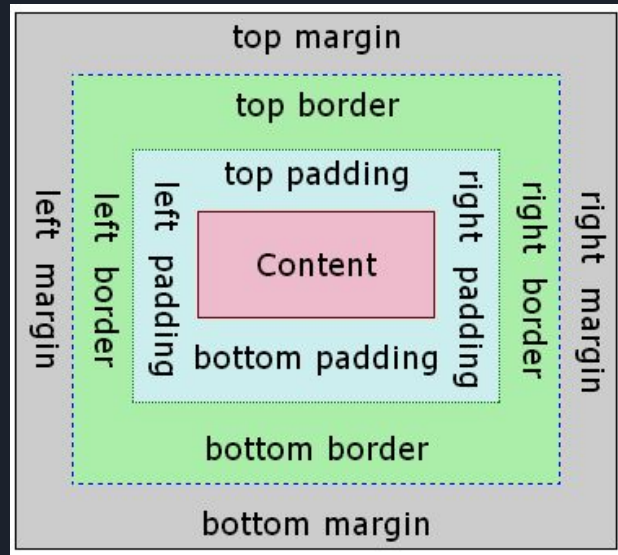
`display`
the most important CSS property

Use the W3C Reference

https://www.w3schools.com/cssref/pr_class_display.asp

Practice!

<http://flexboxfroggy.com/>



04. JavaScript





History

- Created in 1995
- AJAX / XHR sparked massive innovation
- Severe fundamental weaknesses have creative solutions
- Radically extended to unexpected environments
- A language and community in adolescence



5 Things You Must Know about JS

- No language support for importing / modules
- Single-threaded w/ heavy callback usage
- No real class system or support for inheritance
- Functions are 1st-class objects
- On the web principally used for DOM manipulation



OOP Refresher

- An Object combine data + functions that operate on that data
- Class defines the *type* of the object - specifies the fields and functions



Objects in JavaScript

```
{ }           // This creates an object literal
```

```
function f() { // This creates a function, an object  
} ;
```

```
new f();       // This creates an object (?!!)
```



What does `new` do and why is it there?

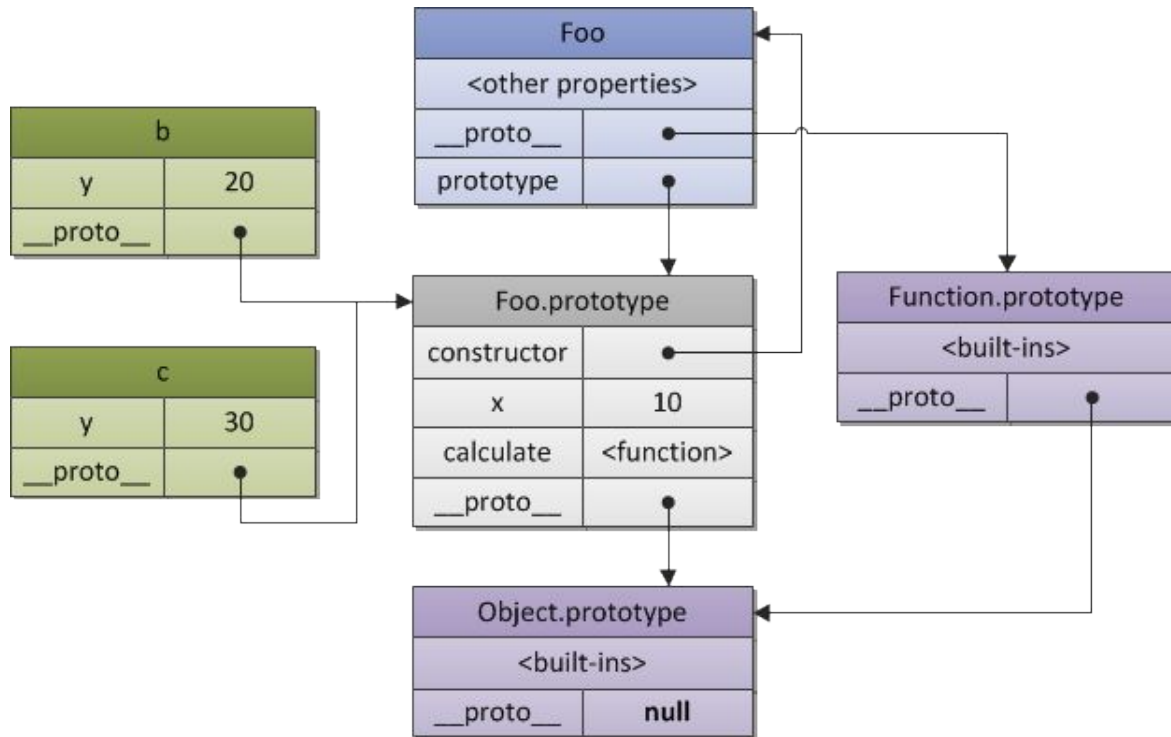
How do these differ? `x = new Foo()` vs `x = {}`

`new` does these things:

1. An Object is created `{}`
2. Special member `{}.__proto__` set to `Foo.prototype`
3. `Foo()` is invoked with the new object as `this`
4. Calling `x.<anything>` searches `prototype.<anything>`

Understanding (2) and (4) are the key to mastering JavaScript

Remember this one weird trick



<http://dmitrysoshnikov.com/ecmascript/javascript-the-core/>