The Future of the Front End

or: How I Learned To Stop Worrying And Love JavaScript

About me

- @iangreenleaf
- Full-stack Rails developer
- Hire me! dev@iangreenleaf.com
- I wrote a book on CoffeeScript: http://goo.gl/5WtQdg

Predicting the future...

Reality

Internet Explorer is not going away

Poll

Internet Explorer is not going away

Mobile is the new IE

We're never getting rid of JavaScript

Or CSS

Or HTML

Standards will never be standard

A way forward

The dark ages

jQuery

jQuery was a watershed moment

```
$(".my-class");
```

```
   // Check if getElementsByClassName can be trusted
   assertUsableClassName = assert(function( div ) {
    // Opera can't find a second classname (in 9.6)
div.innerHTML = "&lang;div class='hidden e'&rang;&lang;/
if ( !div.getElementsByClassName || !div.getElementsByCl
                     return false;
                           }
// Safari 3.2 caches class attributes and doesn't catch
            div.lastChild.className = "e";
 return div.getElementsByClassName("e").length === 2;
                          }),
$(".my-class");
```

jQuery is great, but a little long in the tooth

Solution: compiled languages!

New Language --> Compiler --> HTML/CSS/JavaScript

JavaScript/CSS/HTML is the new bytecode

- Backwards compatible
- Reasonably powerful
- Awful to write by hand

SCSS (née Sass)

```css
a {
  color: #484C55;
}
```

```
$link-color: #484C55;
a {
  color: $link-color;
}
```

```
$link-color: #484C55;
a {
  color: $link-color;
}
a:hover {
  color: $link-color + #444;
}
```

```
.nav-bar {
  a {
    color: black;
  }
}
```

```
@mixin highlighted {
  background-color: yellow;
  font-style: italic;
}
p.callout {
  @include highlighted();
}
```

SCSS + Compass

```
.avatar {
  // No!!!
  -webkit-border-radius: 5;
     -moz-border-radius: 5;
          border-radius: 5;
}
```

```
.avatar {
  // Yess!!!
  @include border-radius(5);
}
```

Haml

```
#nav-bar.dropdown
  %ul.nav-links
    %li Home
    %li About
```

CoffeeScript

```
num = Math.pow 2, 3
if true == true
  console.log "Tautology!"
```

```
animals = ["dog", "cat", "bird"]
for animal in animals
  console.log animal
```

```
double = (num) ->
  num * 2

double(5)
```

```
class Automobile
  honk: ->
    console.log "Beep!"

car = new Automobile()
car.honk()
```

```
class PoliceCar extends Automobile
  honk: ->
    super
    console.log "Wee-oo wee-oo!"
```

```
document.getElementById("nav-bar").cla
```

JavaScript

JavaScript

...well, ES6 to be precise

```
for (let element of [1, 2, 3]) {
  console.log(element);
}
```

```
class PoliceCar extends Automobile {
  honk() {
    super();
    console.log("Wee-oo wee-oo!");
  }
}
```

```
import {firstName, lastName}
  from './another_file';
```

…and more

ClojureScript

By way of Clojure

```
(def a 5)
(+ a 2 3)
```

```clojure
(def shout (fn [x]
  (.toUpperCase x)))
(shout "hello")
(map shout ["hello" "world"])
```

```
(def shout (fn [x]
  (.toUpperCase x)))
(shout "hello")
(map shout ["hello" "world"])
```

```
(def foo (js-obj "bar" "baz"))
; var foo = {bar: "baz"};
(set! (.-bar foo) "quux")
(def clojure-foo (js->clj foo))
(clj->js clojure-foo)
```

asm.js

Can I use asm.js now?
...sorta.

- LLVM is a many-to-many compiler toolset
- Emscripten can go from LLVM to JavaScript
- C --(clang)--> LLVM --(Emscripten)--> asm.js

The best is yet to come

We're never getting rid of
HTML/CSS/JavaScript

...but I don't care.

Fin