Scala.js
Safety & Sanity in the wild west of the web
Li Haoyi, Dropbox, 20 July 2015
1.1 Who am I and what is Scala.js?

- Haoyi works on devtools @ Dropbox
  - Previously web-infra
    - Previously-previous DfB web

- Scala.js isn’t a Dropbox thing
  - Originally PhD project from some guy in Switzerland
  - Reasonably active open-source community
  - Not used @ Dropbox at all
1.2 What is Scala.js?

Write Scala, Run Javascript, Make Website!

Compiler takes care of in between

100s of kb of code, ~0.9-4x slower as “raw” JS

Supports entire Scala language, many libraries
1.3 Javascript

```javascript
var xhr = new XMLHttpRequest()
xhr.open("GET",
   "http://api.openweathermap.org/"+
   "data/2.5/weather?q=Singapore"
)
xhr.onload = function(e){
   if (xhr.status === 200)
      document.body.textContent = xhr.responseText
}
xhr.send()
```
let xhr = new XMLHttpRequest()

xhr.open("GET",
    "http://api.openweathermap.org/"+
    "data/2.5/weather?q=Singapore"
    )

xhr.onload = (e) => {
    if (xhr.status === 200)
        document.body.textContent = xhr.responseText
}

xhr.send()
Let `xhr` = new XMLHttpRequest()

xhr.open("GET",

"http://api.openweathermap.org/" +
"data/2.5/weather?q=Singapore"
)

xhr.onload = (e) => {

if (xhr.status === 200)

    document.body.textContent = xhr.responseText

}

xhr.send()
val xhr = new XMLHttpRequest()

xhr.open("GET",
    "http://api.openweathermap.org/"+
    "data/2.5/weather?q=Singapore"
)

xhr.onload = (e: Event) => {
  if (xhr.status == 200)
    document.body.textContent = xhr.responseText
}

xhr.send()
```scala
val xhr = new XMLHttpRequest()
xhr.open("GET",
   "http://api.openweathermap.org/"+
   "data/2.5/weather?q=Singapore"
)
xhr.onload = (e: Event) => {
   if (xhr.status == 200)
      document.body.textContent = xhr.responseText
}
xhr.send()
```
val (obj, misc) = objects(i)
val t = obj.intersectionTime(ray)
if (t > Epsilon &&
    t < length - Epsilon){
  visible = false
}

var tup = self.Ve.objects[i]
if (null !== tup)
    obj = tup._1, misc = tup._2
else
    throw (new MatchError).init(tup)

var t = obj.intersectionTime(ray)
t > Example$.Epsilon &&
t < length - Example$.Epsilon &&
(visible = !1)
1.7 Scala.js to Javascript

```scala
def IntersectionTimeExample(): Unit = {
  val (obj, misc) = objects(i)
  val t = obj.intersectionTime(ray)
  if (t > Epsilon && t < length - Epsilon) {
    visible = false
  }
}
```

```javascript
function IntersectionTimeExample() {
  var tup = self.Ve.objects[i]
  if (null !== tup)
    obj = tup._1, misc = tup._2
  else
    throw (new MatchError).init(tup)
}
```

```javascript
var t = obj.intersectionTime(ray)
if (t > Example$.Epsilon && t < length - Example$.Epsilon && (visible = !1))
```
1.7 Scala.js to Javascript

val (obj, misc) = objects(i)
val t = obj.intersectionTime(ray)
if (t > Epsilon &&
    t < length - Epsilon){
    visible = false
}

var tup = self.Ve.objects[i]
if (null !== tup)
    obj = tup._1, misc = tup._2
else
    throw (new MatchError).init(tup)

var t = obj.intersectionTime(ray)
t > Example$.Epsilon &&
t < length - Example$.Epsilon &&
(visible = !1)
val (obj, misc) = objects(i)
val t = obj.intersectionTime(ray)
if (t > Epsilon &&
    t < length - Epsilon){
    visible = false
}

var tup = self.Ve.objects[i]
if (null !== tup)
    obj = tup._1, misc = tup._2
else
    throw (new MatchError).init(tup)

var t = obj.intersectionTime(ray)
t > Example$.Epsilon &&
t < length - Example$.Epsilon &&
( visible = !1)
1.8 Notes from the Demo

Fast turn-around time

Compile errors when you make a mistake

Accurate in-editor autocomplete
2.1 How does Scala.js compare to...
2.2 Everyone wants a better web

Safer
More modular, expressive, reusable code
One language across client/server
Async support
More tool-able & better tooling
Fewer warts
2.3 Safety

Uncaught TypeError: undefined is not a function
  o.extend.trim
  b
  d.fx.step.(anonymous function)
  o.fx.update
  o.fx.step
  F
  o.fx.custom
2.4 More Expressive

```javascript
race = (winner, runners...) =>
    print winner, runners

race = function() {
    var winner = arguments[0]
    var runners =
        2 <= arguments.length ?
        slice.call(arguments, 1) : [];
    print(winner, runners);
};
```
2.5 One language for client/server

# This has been ported to our Python Emstring class
# Please keep them both in sync if you need to change something!

```python
class Emstring:
    @em_snippet: (s, maxchars=50, location=0.75) ->
        new Emstring(s.toString()).snippet(maxchars, location).toString()
```
ajaxFoo((a) =>
  bar(a, (b) =>
    baz(a, (c) =>
      b + c
    )
  )
)

async{
  var a = wait(ajaxFoo())
  wait(bar(a)) + wait(baz(a))
}
2.7 More Toolable/Better Tooling

```javascript
a(tabi)(
  v tabindex
  Attr[Int]
  Press ^ to choose the selected (or first) suggestion and insert a dot afterwards
)

p(float.left)(
  "This is my first paragraph"
),
```

Pattern: left: **StylePair**

Is a keyword indicating that the element must float on the left side of its containing block.

MDN
2.8 Fewer Warts

```javascript
javascript> ["10", "10", "10", "10"].map(parseInt)
[10, NaN, 2, 3] // WTF
```
4.1 What is a web application?
4.1 What is a web application?
4.1 What is a web application?
4.1 What is a web application?
4.2 Typed HTML!

div(
    float.left,
    p("I am cow"),
    p("Hear me moo")
)

<div
    style="float: left">
    <p>I am cow</p>
    <p>Hear me moo</p>
</div>
4.2 Typed HTML!

div(
    float.elft,
    p("I am cow"),
    p("Hear me moo")
)
4.2 Typed HTML!

dvi(
    float.left,
    p("I am cow"),
    p("Hear me moo")
)

Not found: value dvi
dvi(^

Compilation failed
4.3 What is a web application?
4.4 Hygienic, Typed CSS!

```scala
trait Simple{
  def btn = cls(
    color := "red",
    height := 125
  )

  def fade = cls.hover(
    opacity := 0.5
  )
}

.$pkg-Simple-btn{
  color: red;
  height: 125px;
}

.$pkg-Simple-fade:hover{
  opacity: 0.5;
}
```
4.4 Hygienic, Typed CSS!

trait Simple{
  def btn = cls(
    color := "red",
    height := 125
  )
  def fade = cls.hover(
    opacity := 0.5
  )
}

.$pkg-Simple-btn{
  color: red;
  height: 125px;
}

.$pkg-Simple-fade:hover{
  opacity: 0.5;
}
trait Simple{
    def btn = cls(
        colro := "red",
        height := 125
    )
    def fade = cls.hover(
        opacity := 0.5
    )
}
4.4 Hygienic, Typed CSS!

trait Simple{
  def btn = cls(
    color := "red",
    height := 125
  )
  def fade = cls.hovre(
    opacity := 0.5
  )
}

value hovre is not a member of object cls
  def fade = cls.hovre(
    ^

Compilation failed
4.5 Hygienic, Typed CSS!

```haskell
val x = div(
    cls := """"$pkg-Simple-btn $pkg-Simple-fade"",
    h1(...),
    p(...)
)
```

```html
<div class=""""$pkg-Simple-btn $pkg-Simple-fade"">
    <h1>...</h1>
    <p>...</p>
</div>
```
4.5 Hygienic, Typed CSS!

```scala
import Simple._
val x = div(
  btn,
  fade,
  h1(...),
  p(...) 
)
```

```html
<div class="$pkg-Simple-btn $pkg-Simple-fade">
  <h1>...</h1>
  <p>...</p>
</div>
```
import Simple._

val x = div(
  btn,
  fadee,
  h1(...),
  p(...) 
)
4.6 What is a web application?
4.7 Ajax!

```javascript
import dom._

var xhr = new XMLHttpRequest()
xhr.open("http://www.bit.ly")
xhr.onload = (x) => {
    ...
}
xhr.send()
```

```javascript
var xhr = new XMLHttpRequest()
xhr.open("http://www.bit.ly")
xhr.onload = (x: Event) => {
    ...
}
xhr.send()
```
4.7 Ajax!

// Javascript
$j.ajax("/api/list", { 
  data: inputBox.value,
  onComplete: function(res){ ... }
})
4.7 Ajax!

// Javascript
$j.ajax("/api/list", {
    data: inputBox.value,
    onComplete: function(res){ ... }
})

How do we know this correct?
4.7 Ajax!

// Javascript
$j.ajax("/api/list", { data: inputBox.value, onComplete: function(res){ ... } })

How do we know this correct? And this value?
4.7 Ajax!

// Javascript
$.ajax("/api/list", {
  data: inputBox.value,
  onComplete: function(res){ ... }
})

How do we know this correct?
And this value?
And that we’re using this res the right way?
4.7 Typed Ajax!

// Javascript
$j.ajax("/api/list", {
  data: inputBox.value,
  onComplete: function(res){ ... }
})

// Scala.js
val res = Ajax[Api].list(inputBox.value).call()
Live Demo

Typed Ajax
5.1 What is a web application?

[Diagram showing interactions between HTML, Browser, Server, Database, and highlighting the term 'Safety']

Maybe Safety?
5.2 Scala.js gives you...

- Shared client-server code

- Shared client-server *libraries*

- A reasonable language to write the client in
5.3 Scala.js gives you...

Everything that

- ES6 gives (string-interp, const, class, =>, …)
- Typescript gives (types, generics, …)
- Persistent collections,
- Monads, macros, mixins, …
- Higher kinded types, virtual classes, …
5.3 Scala.js does *not* give you

undefined is not a function
Mal-formed HTML
Un-used CSS classes
Using un-defined CSS classes
CSS class-name collisions
Mal-formed Ajax requests
Fun uses of Scala.js

Ray Tracing

Platform Games

TodoMVC

Djinni.js
5.4 Safety & Sanity on the Web

Shared code between Client & Server

Enforce safety throughout the entire application, not just the Javascript

Not 12 months from now, but today!
Questions?