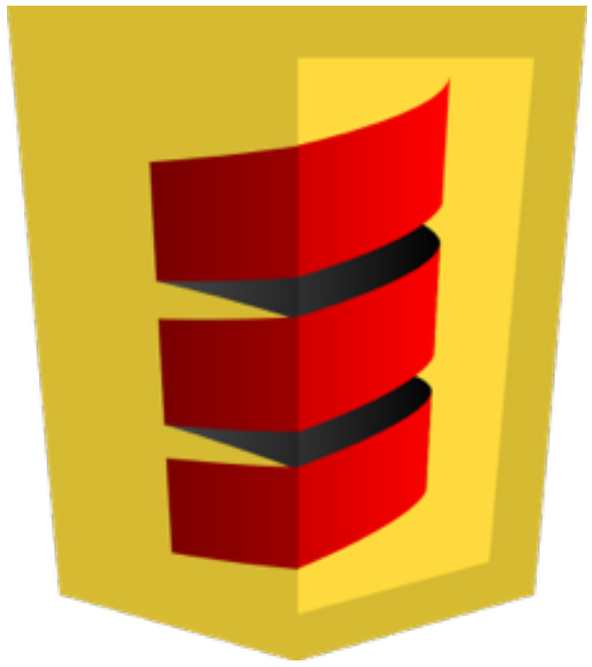
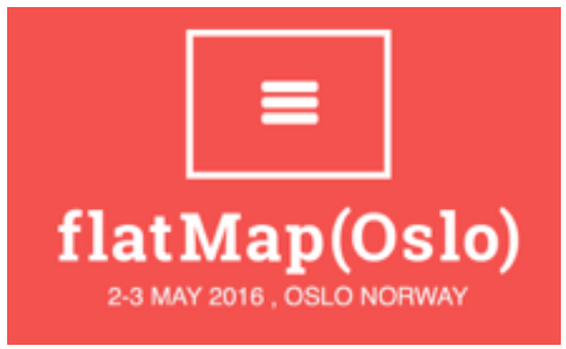


Scala.js workshop

- https://github.com/oyvindberg/scalajs-workshop

Øyvind Raddum Berg
@olvindberg



Who am i?

- Worked professionally with Scala for 4+ years
- Have used Scala.js for a year
- Works for Arktekk

What we will cover today

- Very quick intro
- A few limitations to keep in mind
- Building / development
- Look at a few libraries for typed Html, Dom, Ajax
- Hack a bit on an example project

What is Scala.js?

- Write Scala, compile, run in browser
- Full support for the entire Scala language*
- Full javascript interop
- Fast enough (0.9x - 4x times slower than javascript),
- Small enough (hundreds of kilobytes)
- Production-ready February 2015

What *is* Scala.js?

- Primarily: An output target for the Scala compiler
- An optimising linker (using Google Closure Compiler)
- A Javalib - reimplemented functionality from Java
- A Scalalib - reimplemented functionality from Scala
- A runtime library
- An sbt plugin

Scala.js limitations

- No Java support (source or binary)
- No reflection
- No native
- Browser-sandbox (no file system, etc)
- Slightly differing semantics
  - beware of Floats and Longs,
  - Equality tests and toString for primitives
  - Exceptions from VM (ClassCastException, ArithmeticException, etc)

How?

```
Option("Hello") match {
  case Some(msg) ⇒ dom.window.alert(msg)
  case None      ⇒ dom.window.alert("No Message")
}

var x1 = $m_s_Option$().apply__O__s_Option("Hello");
if ($is_s_Some(x1)) {
  var x2 = $as_s_Some(x1);
  var msg = $as_T(x2.x$2);
  $m_Lorg_scalajs_dom_package$
().window__Lorg_scalajs_dom_raw_Window().alert(msg)
} else {
  var x = $m_s_None$();
  if ((x === x1)) {
    $m_Lorg_scalajs_dom_package$
().window__Lorg_scalajs_dom_raw_Window().alert("No Message")
  } else {
    throw new $c_s_MatchError().init___O(x1)
  }
};
```

Anatomy of a Scala.js project

- Scala.js sbt plugin
- CrossProject
- Dependency handling
- Packaging of Javascript dependencies
- fastOptJS/fullOptJS
- Generate launcher
- Testing / Running (Rhino, Node.js, Phantom.js)

Guided tour of libraries

- scala-js-dom
- scalatags
- uPickle
- autowire

Suggestions

- Try to break it! The compiler generally has your back, and a lot of the pain points from traditional web development are gone, though some remain. By refactoring the application you can get a feeling for what is still brittle

- Play around with scalatags/html templating. For example we now litter the html with hard-coded Bootstrap Css class names, perhaps you could write a Bootstrap wrapper?

- Extend the application to show metadata. Last changed? file size? Right now it's pretty bare bones

- Add support for showing content of files. Such basic functionality missing!. Can you make it happen?

- Breadcrumbs for the parent folders instead of the back button.

- Add support for several file browsers in tabs on the same page. Bootstrap has tabs, and the file browser just needs a DOM element to render to)

- Add support for remembering state. The Local Storage API is defined in dom.localStorage. You probably want to use uPickle for serialization

My take

- Solves **a lot** of headaches for web dev
- Having access to big parts of the Scala ecosystem is **awesome**
- Young frontend ecosystem, still very influenced by javascript, no “one, true way”
- Interplay between ide, sbt, plugins, compilers and dependencies is complex