# HTML5 and CSS3 for Mobile Applications

Prof. Paul Krause, University of Surrey Laying the Foundations for a Makeover

# Objectives for today

- The new HTML5 elements
- How to make IE < 9 recognise them</li>
- How to reset all browsers to the same foundation
- Starting off the assembly of your very own Boilerplate

```
<!DOCTYPE html>
<html>
<head>
 <title><%= title %></title>
 <%= stylesheet_link_tag "application", :media => "all" %>
 <%= javascript_include_tag "application" %>
 <%= csrf_meta_tags %>
</head>
<body>
   <%= render 'layouts/header' %>
   <%= notice %>
   <%= alert %>
   <%= render 'layouts/navigation' %>
   <section class="sidebar" id="right">
       <% if user_signed_in? %>
           Signed in as <%= current_user.email %>:
           <%= link_to "Sign out", destroy_user_session_path, :method => :delete %>
       <% else %>
           <%= link_to "Sign up", new_user_registration_path %> or <%= link_to "sign in",</pre>
               new_user_session_path %>
       <% end %>
   </section>
   <%= yield %>
   <%= render 'layouts/footer' %>
</body>
</html>
```

# **Ashtead Allotments Association**Healthy Eating and Exercise preserve Life's Balance

- Home
- Events
- Discussion
- Shop

Sign up or sign in

#### AaaCore#index

Find me in app/views/aaa\_core/index.html.erb

© 2012 Ashtead Allotment's Association

# HTML5 - away with 'divitis'

- abbr
- article
- aside
- audio
- canvas
- datalist
- details
- figcaption
- figure
- footer
- header

- hgroup
- mark
- meter
- nav
- output
- progress
- section
- summary
- time
- video

#### ARIA roles

- Accessible Rich Internet Applications
  - part of the Web Accessibility Initiative (WAI)
- Designed to improve the accessibility and interoperability of web content and applications
- ARIA roles can be added to tags to provide more meaning
  - role = "{value}"
- See: <a href="http://www.w3.org/TR/wai-aria/">http://www.w3.org/TR/wai-aria/</a>

Setting Up a Boilerplate

#### Boilerplate = Standardised View Code

- Rails has started us off with "views/layouts/application.rb"
- But it is worth adding to this, and reusing it/refining it across your application portfolio
- Helps ensure consistency across browsers and devices

## Character Encoding

- Rule No. 1
  - "You should always specify a character encoding on every HTML page you serve"
  - http://blog.whatwg.org/the-road-to-html-5-character-encoding
- There are many choices but UTF-8 makes internationalisation easier to deal with and is recommended in the HTML5 spec

## Character Encoding

- Rule No. 1
  - "You should always specify a character encoding on every HTML page you serve"
  - http://blog.whatwg.org/the-road-to-html-5-character-encoding
- There are many choices but UTF-8 makes internationalisation easier to deal with and is recommended in the HTML5 spec

```
<head>
    <meta charset="utf-8">
        <title><%= title %></title>
        <%= stylesheet_link_tag "application", :media => "all" %>
        <%= javascript_include_tag "application" %>
        <%= csrf_meta_tags %>
        </head>
```

#### Joel on Software

- "The Absolute Minimum" to read
  - <a href="http://www.joelonsoftware.com/articles/Unicode.html">http://www.joelonsoftware.com/articles/Unicode.html</a>

# Bit of a Snag:(

- The new HTML5 elements are not supported by IE < 9</li>
- Not too difficult to fix, by using JavaScript to create the new elements
  - The "HTML5 shiv"
- · We will use Modernizr to do this
  - and a shed load of other things



Taking advantage of the new capabilities of HTML5 and CSS3 can mean sacrificing control over the experience in older browsers. Modernizr 2 is your starting point for making the best websites and applications that work exactly right no matter what browser or device your visitors use.

Thanks to the new Media Query tests and builtin YepNope.js micro-library as

Modernizr.load(), you can now combine feature detection with media queries and conditional resource loading. That gives you the power and flexibility to optimize for every circumstance.

Check out the full list of features that Modernizr detects, or learn more about conditional resource loading with Modernizr.

February 6th, 2012

Modernizr 2.5: Supercharged for

Our biggest release yet!

Latest news

**PREVIOUS** 

November 7th, 2011

**How Does Modernizr's Test Suite** 

Work?

September 1st, 2011

**Modernizr and CDNs** 

June 1st, 2011

Modernizr 2

#### Get started with Modernizr

While Modernizr gives you finer control over the experience through JavaScriptdriven feature detection, it is important to continue to use best practices throughout your development process. Use progressive enhancement wherever you can, and don't sacrifice accessibility for convenience or performance.

- · Documentation: Getting started
- · Taking Advantage of HTML5 and CSS3 with Modernizr, Faruk Ates
- How to use Modernizr, Inayaili de León
- · Modernizr: front-end development done right, Ryan Seddon
- [wiki] The Undetectables: features that cannot be detected
- · [wiki] Cross-browser Polyfills

Also check out our Resources section.

Tip: check our Modernizr test suite to quickly test your current browser's features.

USED BY Emitter Google Microsoft Economist About.com the knot 6000 posterous



HTML







**CREATED BY** 



Faruk Ateş **Creator & Lead** @KuraFire



@SlexAxton



Lead Developer paul\_irish



Rvan Seddon



Modernizr is the right micro-library to get you up and running with HTML5 & CSS3 today.

## Updating: app/assets/javascripts/application.js

```
FOLDERS
                                   application.js
                   application.html.erb ×

▼ AAA Website

                      // This is a manifest file that'll be compiled into applica
 ▶ .idea

▼ app

                      // listed below.

    assets

   images
                      // Any JavaScript/Coffee file within this directory, lib/as

▼ javascripts

      aaa_core.js.coffee
                   5 // or vendor/assets/javascripts of plugins, if any, can be
                      //
      comments.js.coffee
                      // It's not advisable to add code directly here, but if you
      events.js.coffee
      modernizr-2.5.3.js
                      // the compiled file.
   stylesheets
  controllers
                                     THE FIRST BLANK LINE MARKS THE END OF WHAT'S TO
  helpers
  mailers
                      // GO AFTER THE REOUIRES BELOW.
  models
                  12
  views
                  13 //= require modernizr-2.5.3
 confia
 ▶ db
                  14 //= require jquery
 doc
                  15 //= require jquery_ujs

    lib

                  16 //= require_tree .
 log
 public
                  17
 script
```

## Updating: app/assets/javascripts/application.js

```
FOLDERS
                    application.html.erb ×
                                   application.js

▼ AAA Website

                      // This is a manifest file that'll be compiled into applica
 ▶ .idea

▼ app

                      // listed below.

    assets

   images
                      // Any JavaScript/Coffee file within this directory, lib/as

▼ javascripts

      aaa_core.js.coffee
                   5 // or vendor/assets/javascripts of plugins, if any, can be
                      //
      comments.js.coffee
                       // It's not advisable to add code directly here, but if you
      events.js.coffee
      modernizr-2.5.3.js
                       // the compiled file.
   stylesheets
  controllers
                                      THE FIRST BLANK LINE MARKS THE END OF WHAT'S TO
  helpers
  mailers
                       // GO AFTER THE REQUIRES BELOW.
  models
                  12
  views
                       //= require modernizr-2.5.3
 confia
 ▶ db
                       //= require jquery
 doc
                      //= require jquery_ujs

    lib

                      //= require_tree .
 log
                  16
 public
                  17
 script
```

Normalizing Browsers

# Why Normalize (aka reset)?

- Browsers do not agree on the basic presentation defaults
- Eric Meyer's original reset even removed **bold** from <strong> and *italics* from <em>
  - There is no practical need to go that far
- We will use Normalize.css
  - "preserves useful defaults; normalizes styles for a wide range of elements; corrects bugs and common browser inconsistencies; ..."
  - http://necolas.github.com/normalize.css/

#### app/assets/stylesheets/application.css

```
FOLDERS
                     application.html.erb ×
                                      application.js
                                                        application.css

▼ AAA Website

                         /*
 ▶ .idea

▼ app
                          * This is a manifest file that'll be compiled int

    assets

                          * listed below.
    images
    javascripts

▼ stylesheets

                          * Any CSS and SCSS file within this directory, li
      aaa core.css.scss
                          * or vendor/assets/stylesheets of plugins, if any
      application.css
      comments.css.scss
      events.css.scss
                          * You're free to add application-wide styles to t
      normalize.css
                          * compiled file, but it's generally better to cre
      scaffolds.css.scss
  controllers
                    10
                          *
  helpers
                    11
                          *= require_self
  mailers
                          *= require normalize
                    12
  models
                    13
  views
                         */
 confia
                    14
 ▶ db
```

#### app/assets/stylesheets/application.css

```
FOLDERS
                     application.html.erb ×
                                      application.js
                                                       application.css

▼ AAA Website

                         /*
 ▶ .idea

▼ app
                          * This is a manifest file that'll be compiled int
  assets
                          * listed below.
   images
   javascripts

▼ stylesheets

                          * Any CSS and SCSS file within this directory, li
      aaa core.css.scss
                          * or vendor/assets/stylesheets of plugins, if any
      application.css
      comments.css.scss
      events.css.scss
                          * You're free to add application-wide styles to t
      normalize.css
                          * compiled file, but it's generally better to cre
      scaffolds.css.scss
  controllers
                    10
                          *
  helpers
                    11
                          *= require self
  mailers
                          *= require normalize
  models
                    13
  views
 confia
                    14
 ▶ db
```

Rolling your own Boilerplate

#### Borrowing from HTML5 Boilerplate

- We have fixed some of the issues, but there are many more
- Using a Boilerplate without fully understanding what it all does is not recommended
- So,
- We are going to pull out of HTML5 Boilerplate what we need, in an as needed basis
  - http://html5boilerplate.com/

#### **Ashtead Allotments Association**

#### Healthy Eating and Exercise preserve Life's Balance

Home Events Discussion Shop

Sign up or sign in

#### AaaCore#index

Find me in app/views/aaa\_core/index.html.erb

© 2012 Ashtead Allotment's Association

#### What We Have Done

- Foundation is now in place
  - Modulo pasting over a few more cracks as we go
- Now we are ready to trowel on the styling