



Supporting Opportunistic Programmers with Better Visualizations

VISSOFT 2016

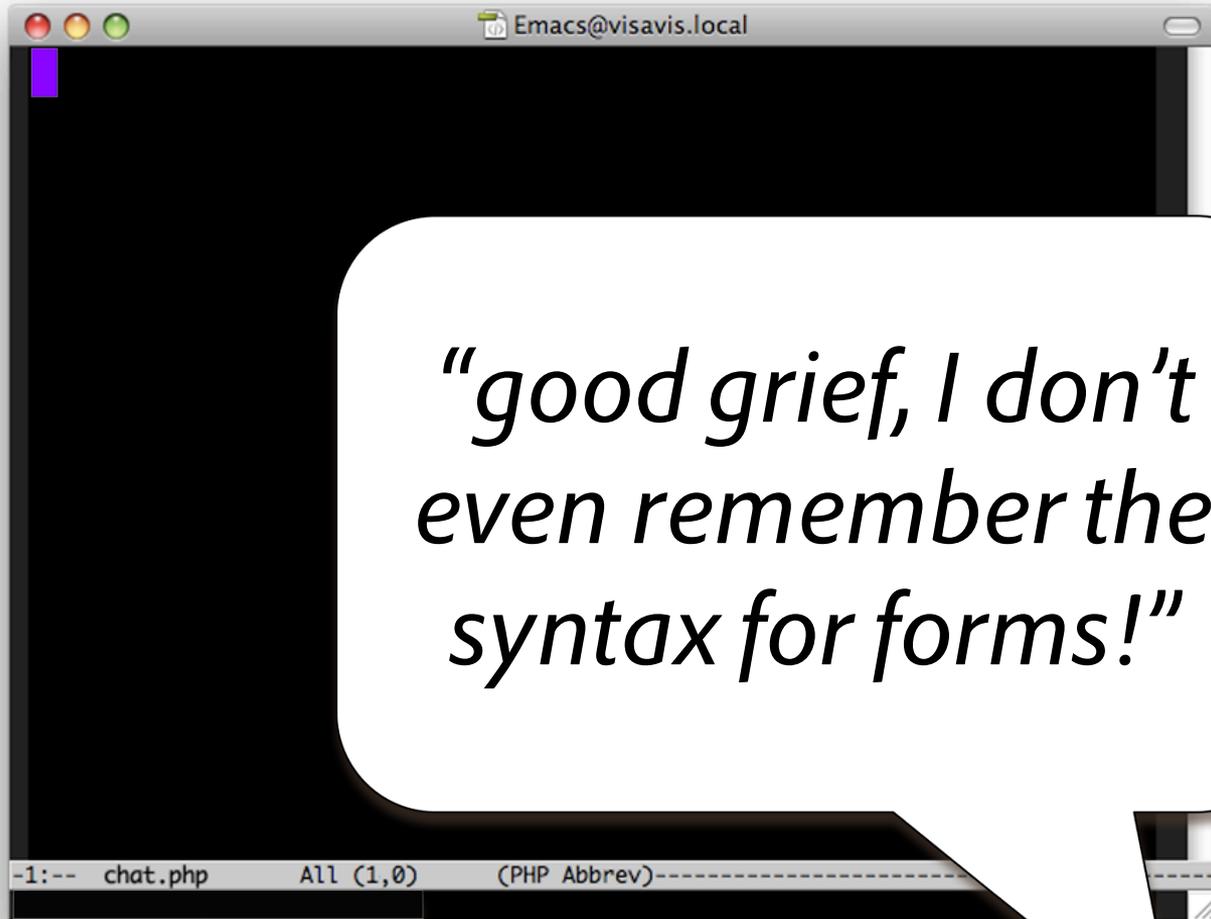
Joel Brandt

Adobe Research

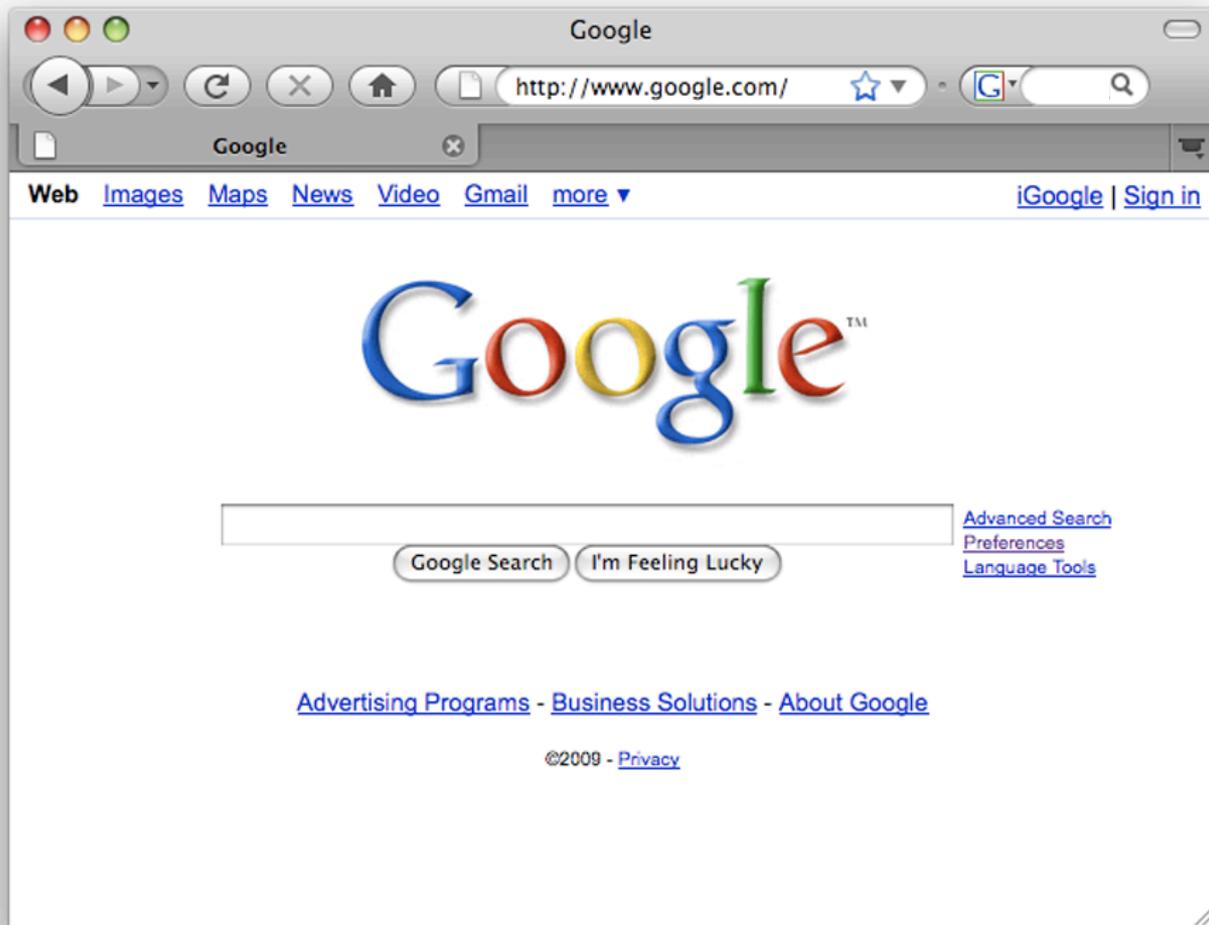
Creative Technologies Lab

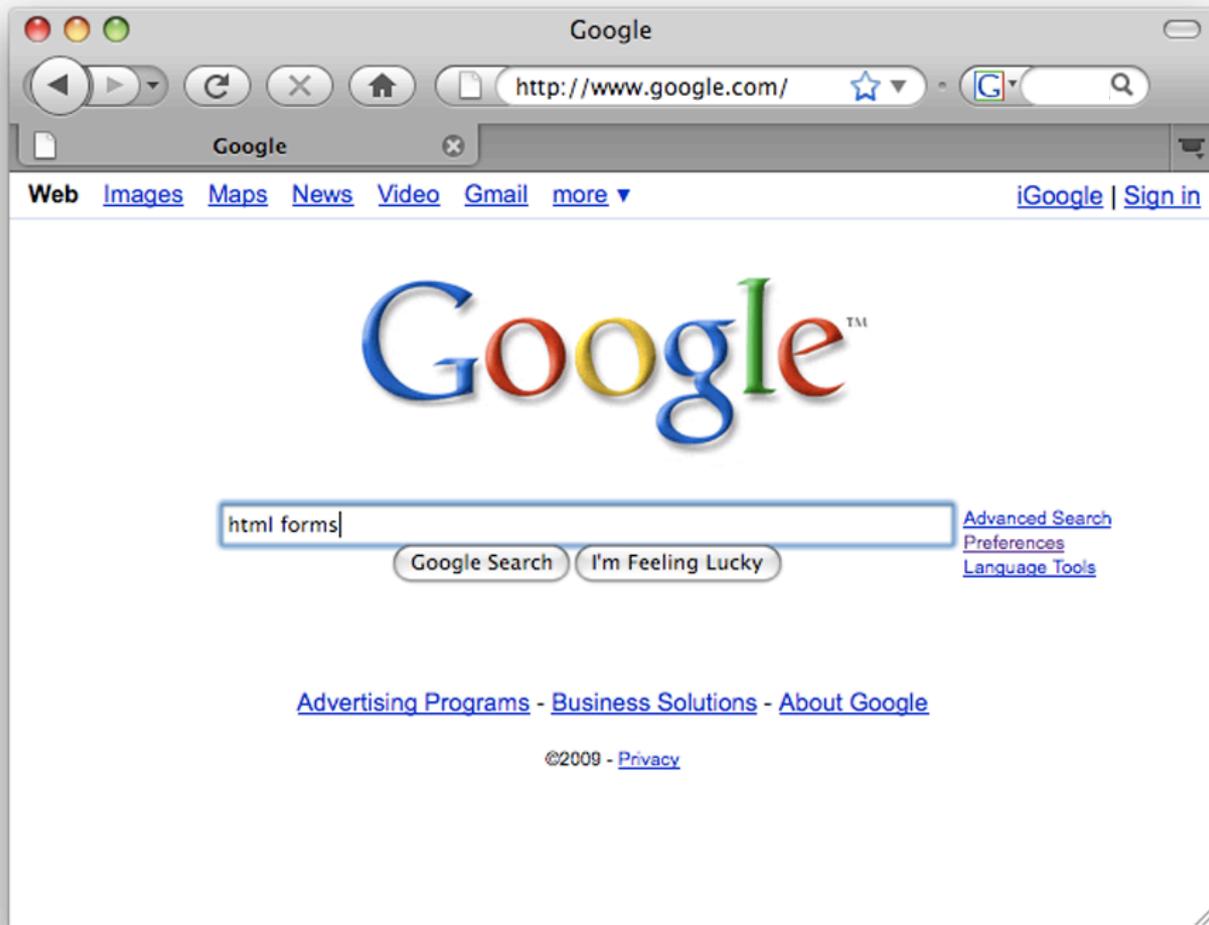
- Open or shared IP, ongoing collaborations with universities
- Graphics, HCI, Vis, ML/AI, Programming Languages
- 22 Researchers, 9 Research Engineers
- ~70 academic publications per year at CHI, UIST, SIGGRAPH, CVPR, ICCV, ICML, etc.
- ~15 major contributions to Adobe products per year
- 50-60 Ph.D.-student-level interns per year
- Expect to grow by about 10 technical staff in 2017

research.adobe.com



*"good grief, I don't
even remember the
syntax for forms!"*





Google

http://www.google.com/

Google

Web [Images](#) [Maps](#) [News](#) [Video](#) [Gmail](#) [more](#) ▾

[iGoogle](#) | [Sign in](#)

Google™

Google Search

I'm Feeling Lucky

[Advanced Search](#)

[Preferences](#)

[Language Tools](#)

[Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2009 - [Privacy](#)

Google

http://www.google.com/search

Google

Web [Images](#) [Maps](#) [News](#) [Video](#) [Gmail](#) [more](#) [Sign in](#)

Google [Advanced Search](#)
[Preferences](#)

Web Results 1 - 10 of about 182,000,000 for [html forms](#). (0.18 seconds)

[HTML Forms and Input](#)
HTML Forms are used to select different kinds of user input. ...
This example demonstrates how to create text fields on an HTML page.
www.w3schools.com/html/html_forms.asp - 34k -
[Cached](#) - [Similar pages](#)

[Forms in HTML documents](#)
An HTML form is a section of a document containing normal content, markup, special elements called controls (checkboxes, radio buttons, menus, etc. ...
www.w3.org/TR/html401/interact/forms.html - 138k -
[Cached](#) - [Similar pages](#)

[HTML Tutorial - Forms](#)
An overview of each HTML form type with examples of code for each type of form.

Sponsored Links

[HTML Sample Forms](#)
Choose From Around 100 Samples. In Industry Since 1998. Contact Now!
www.FormSite.com/HTML

[Make Web Forms in Minutes](#)
Text Areas, Drop Down Boxes & More
No Programming - Free Trial!
www.coffeecup.com

[Easy HTML Form Builder](#)
Build web forms, add validation, & store data. No code required!
www.Ektron.com

The image shows a web browser window with the title "HTML Forms and Input". The address bar contains the URL "http://www.w3schools.com/ht". The browser interface includes standard navigation buttons (back, forward, refresh, home) and a search bar. The page content is organized into a left-hand navigation menu and a main content area.

Navigation Menu (Left):

- HTML Colornames
- HTML Colorvalues
- HTML Quick List
- HTML Advanced**
- HTML Layout
- HTML Frames
- HTML Fonts
- HTML 4.0 Why
- HTML CSS
- HTML Entities
- HTML Head
- HTML Meta
- HTML URLs
- HTML Scripts
- HTML Attributes
- HTML Events
- HTML URL Encode
- HTML Webserver
- HTML Summary
- HTML Examples**
- HTML Examples
- HTML Quiz
- HTML Certificate

Main Content Area:

(You can find more examples at the bottom of this page)

Forms

A form is an area that can contain form elements.

Form elements are elements that allow the user to enter information (like text fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

A form is defined with the `<form>` tag.

```
<form>
.
input elements
.
</form>
```

Input

The most used form tag is the `<input>` tag. The type of input is specified with the `type` attribute. The most commonly used input types are explained below.

Text Fields

HTML Forms and Input

http://www.w3schools.com/ht

The Form's Action Attribute and the Submit Button

When the user clicks on the "Submit" button, the content of the form is sent to the form's action attribute defines the name of the file to send the content to. The form's action attribute usually does something with the received input.

```
<form name="input" action="html_form_submit.asp" method="get">
Username:
<input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

How it looks in a browser:

Username:

If you type some characters in the text field above, and click the "Submit" button will send your input to a page called "html_form_submit.asp". The page will show the received input.

 **More Examples**

HTML Forms and Input

http://www.w3schools.com/ht

The Form's Action Attribute and the Submit Button

When the user clicks on the "Submit" button, the content of the form is sent to the form's action attribute defines the name of the file to send the content to. The form's action attribute usually does something with the received input.

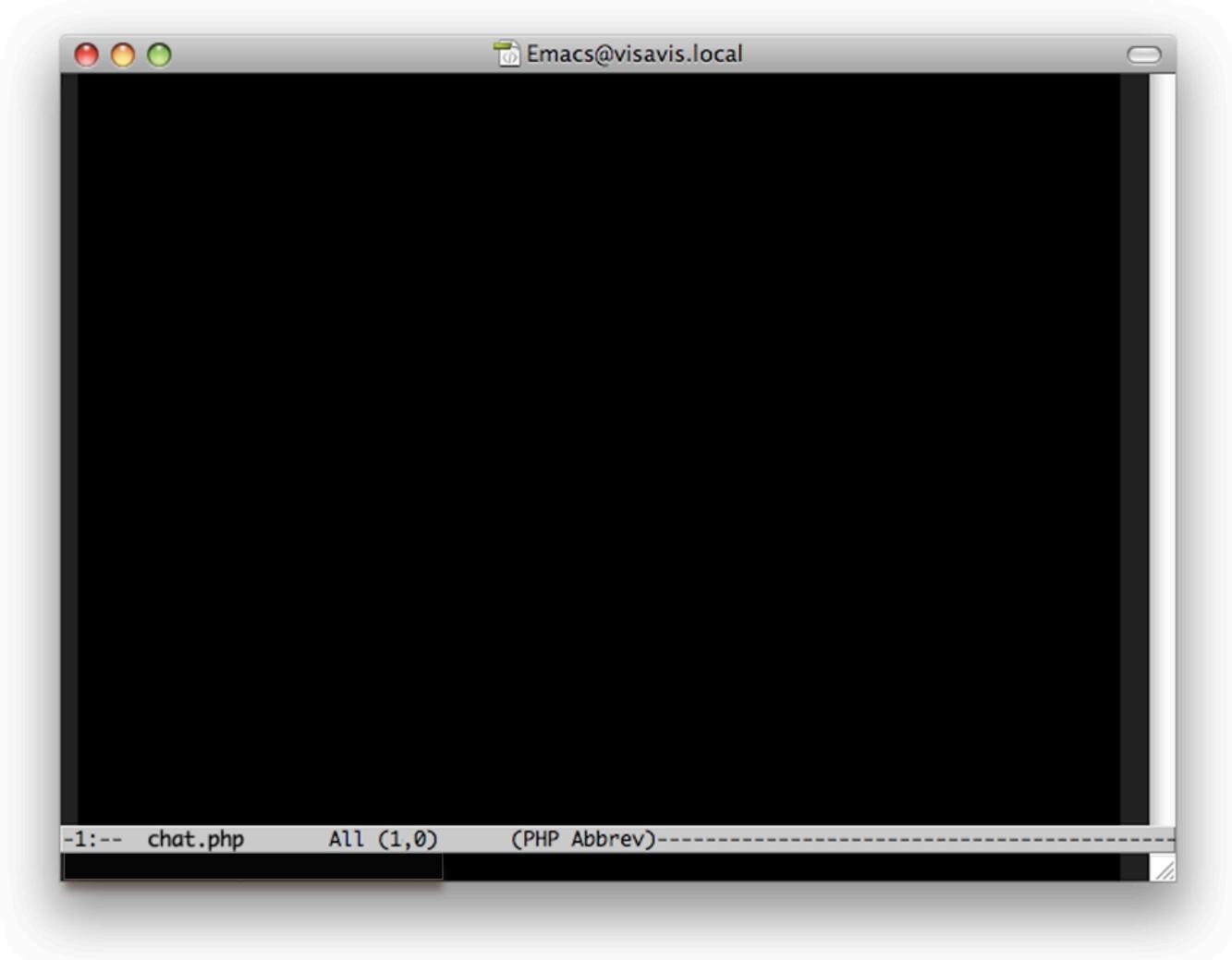
```
<form name="input" action="html_form_submit.asp" method="get"
Username:
<input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

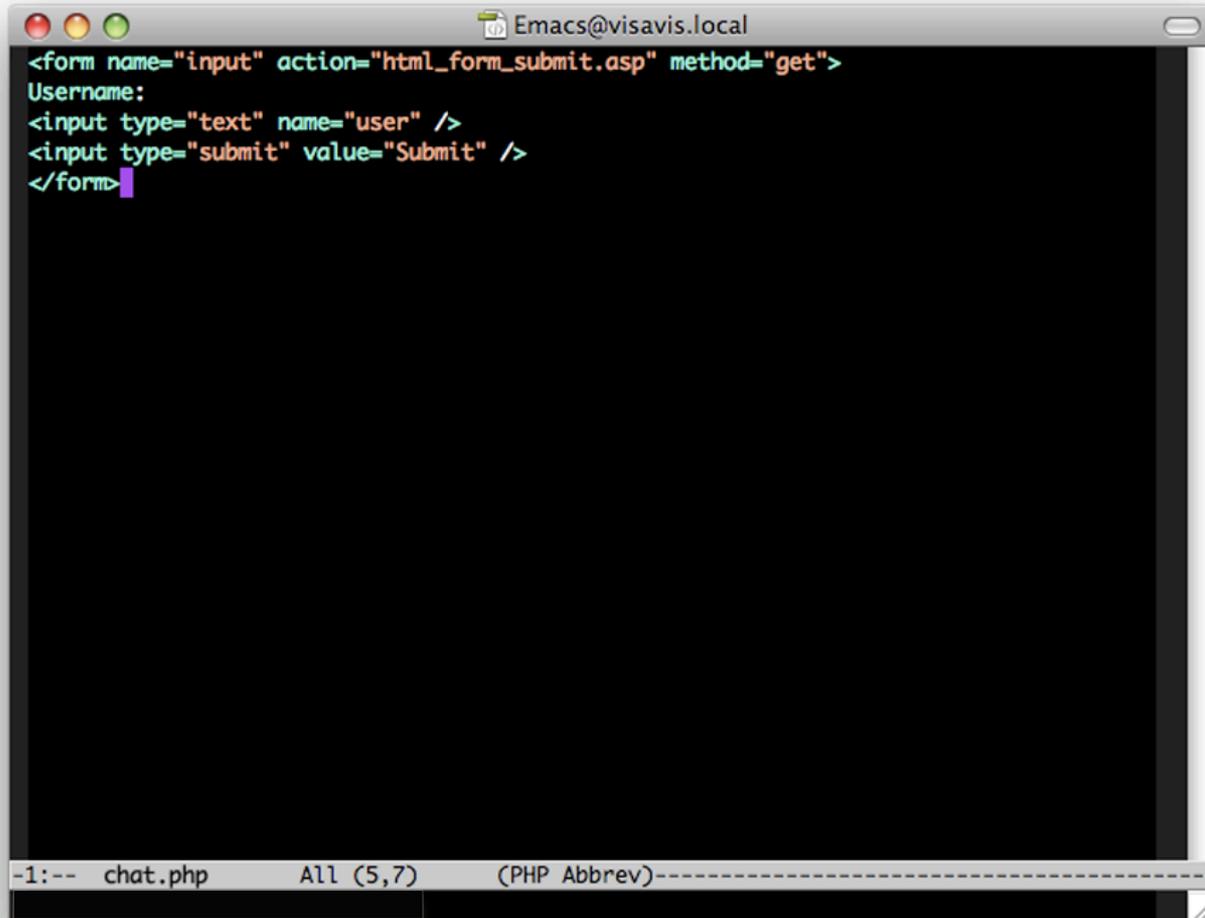
How it looks in a browser:

Username:

If you type some characters in the text field above, and click the "Submit" button will send your input to a page called "html_form_submit.asp". The page will show the received input.

 **More Examples**

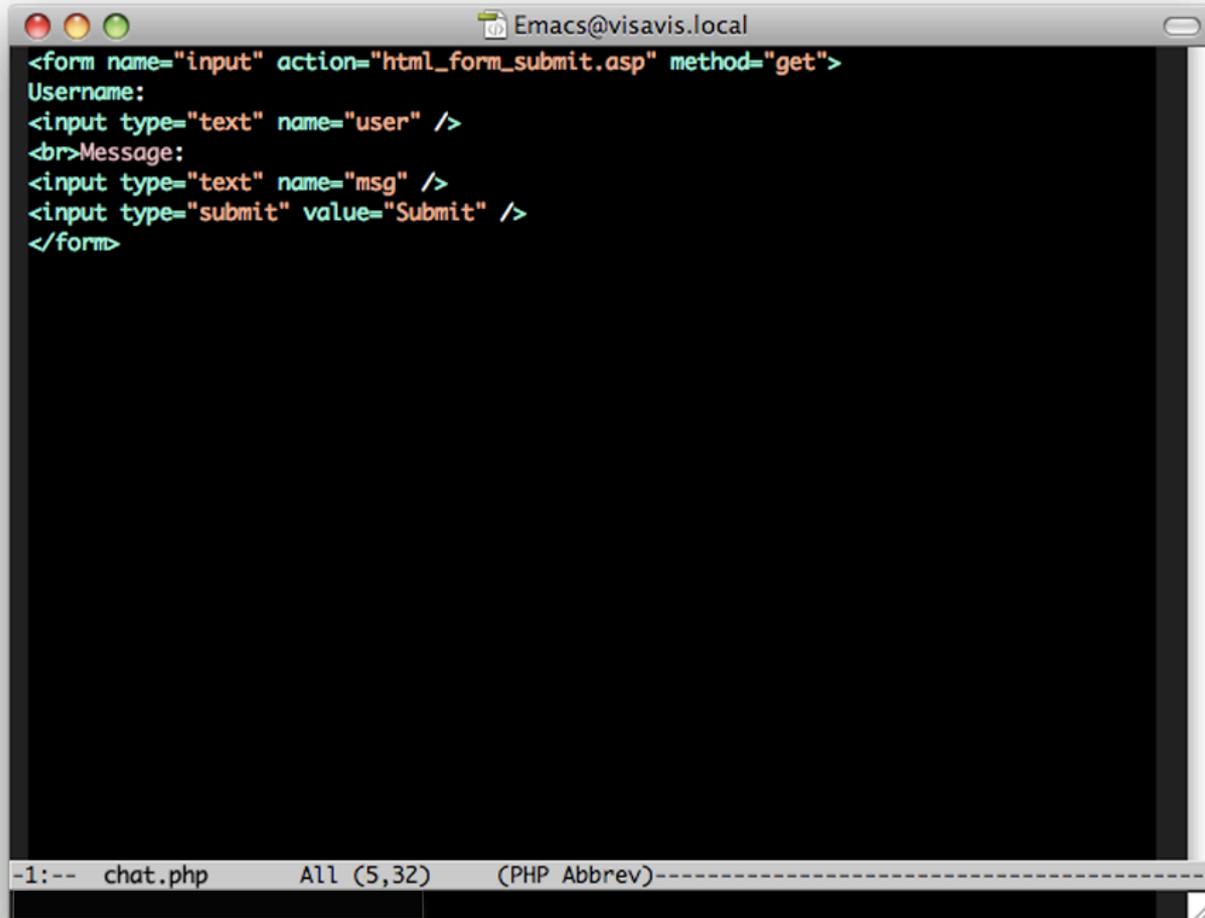




The image shows a screenshot of an Emacs editor window. The title bar at the top reads "Emacs@visavis.local". The main editing area contains the following HTML code:

```
<form name="input" action="html_form_submit.asp" method="get">
Username:
<input type="text" name="user" />
<input type="submit" value="Submit" />
</form>
```

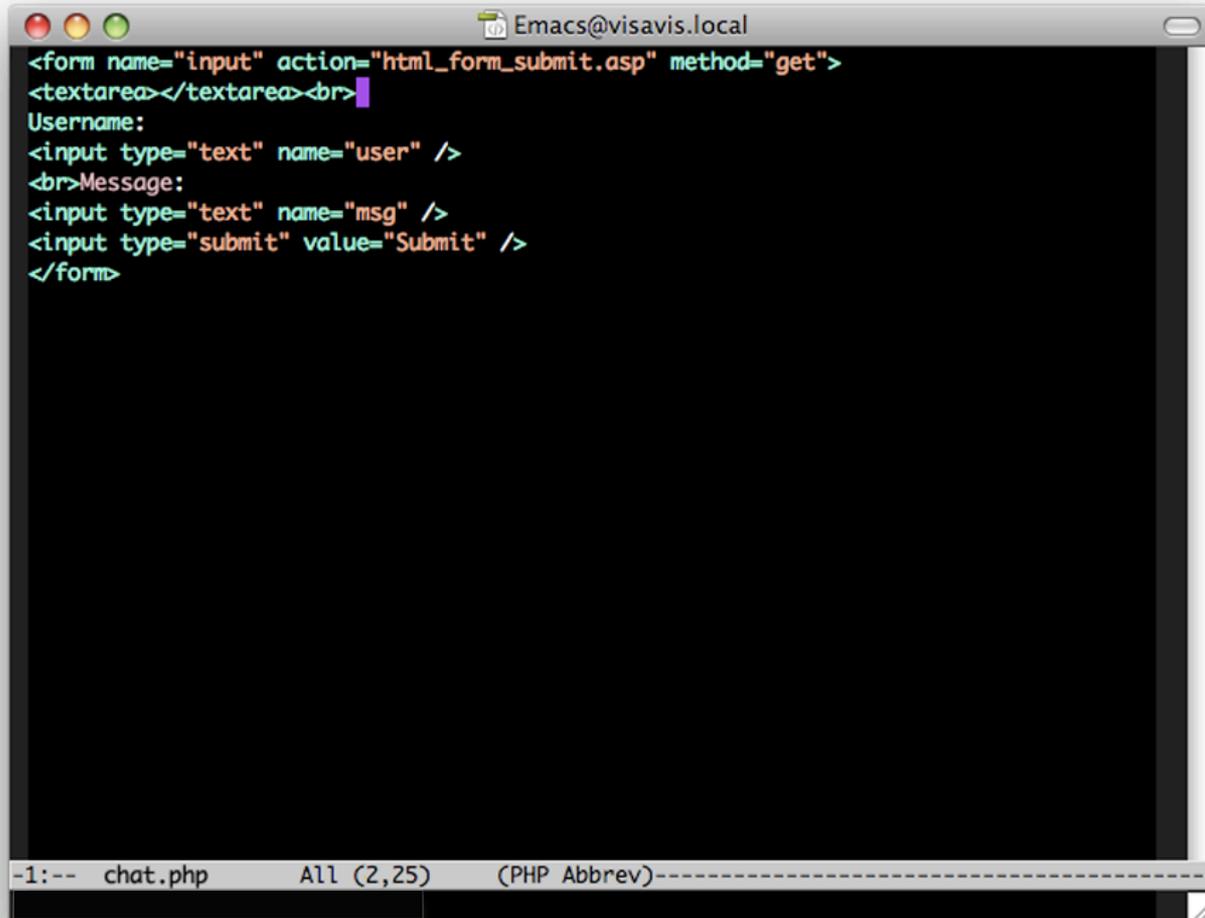
The cursor is positioned at the end of the last line. The status bar at the bottom of the window displays the following information: "-1:-- chat.php All (5,7) (PHP Abbrev)".



The image shows a screenshot of an Emacs editor window. The title bar at the top reads "Emacs@visavis.local". The main editing area contains the following HTML code:

```
<form name="input" action="html_form_submit.asp" method="get">
Username:
<input type="text" name="user" />
<br>Message:
<input type="text" name="msg" />
<input type="submit" value="Submit" />
</form>
```

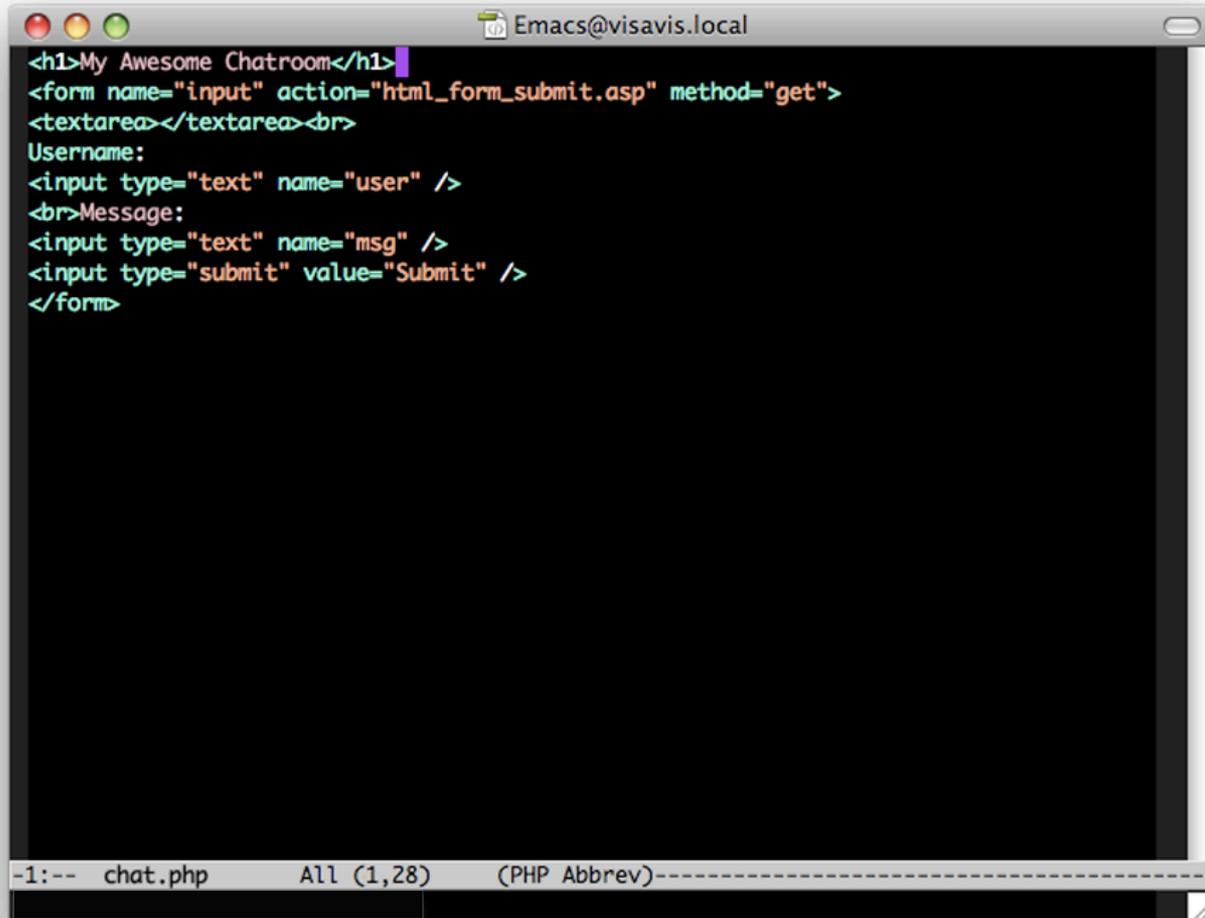
At the bottom of the window, the status bar displays the following information: "-1:-- chat.php All (5,32) (PHP Abbrev)".



The image shows a screenshot of an Emacs editor window titled "Emacs@visavis.local". The editor displays HTML code for a form. The code is as follows:

```
<form name="input" action="html_form_submit.asp" method="get">
<textarea></textarea><br>
Username:
<input type="text" name="user" />
<br>Message:
<input type="text" name="msg" />
<input type="submit" value="Submit" />
</form>
```

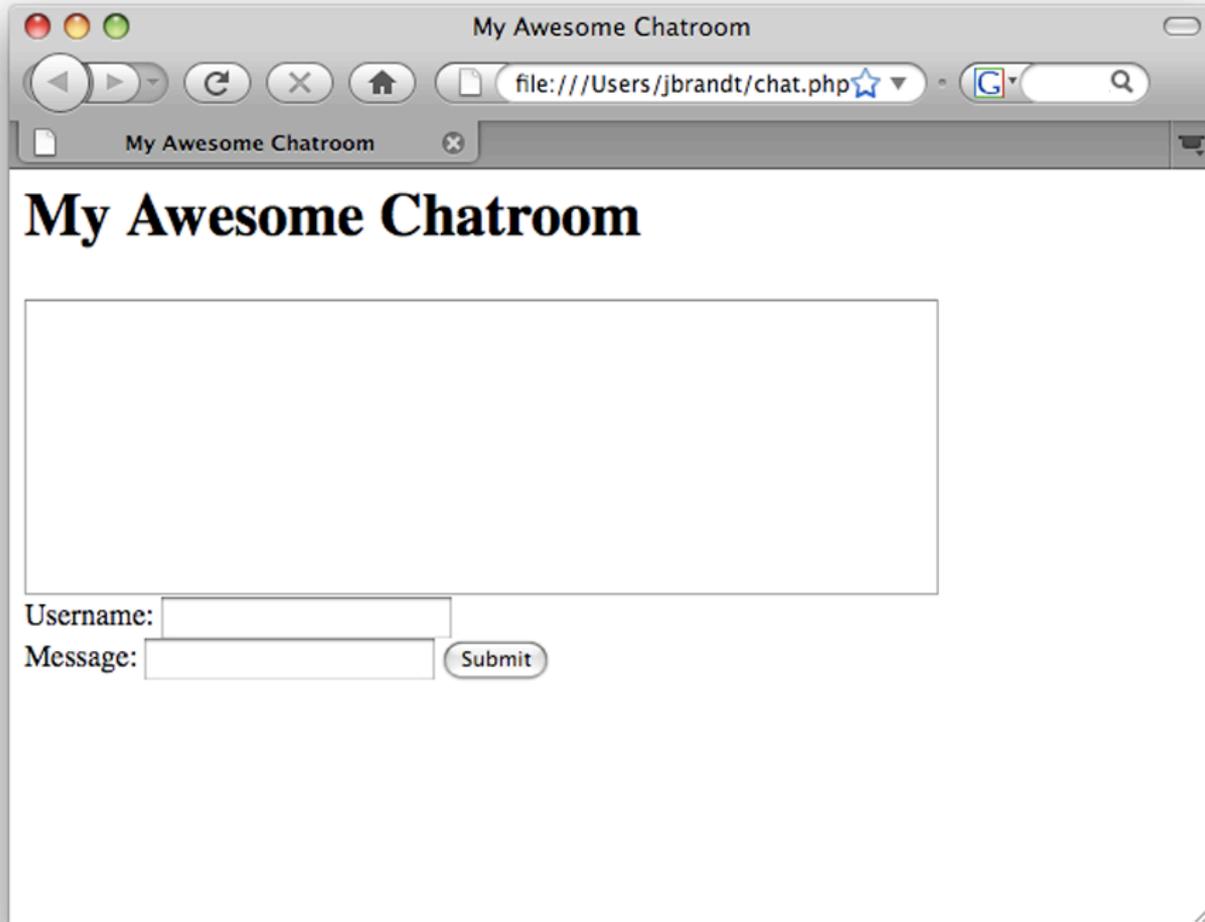
The status bar at the bottom of the window shows the file name "chat.php", the current line and column "All (2,25)", and the encoding "(PHP Abbrev)".



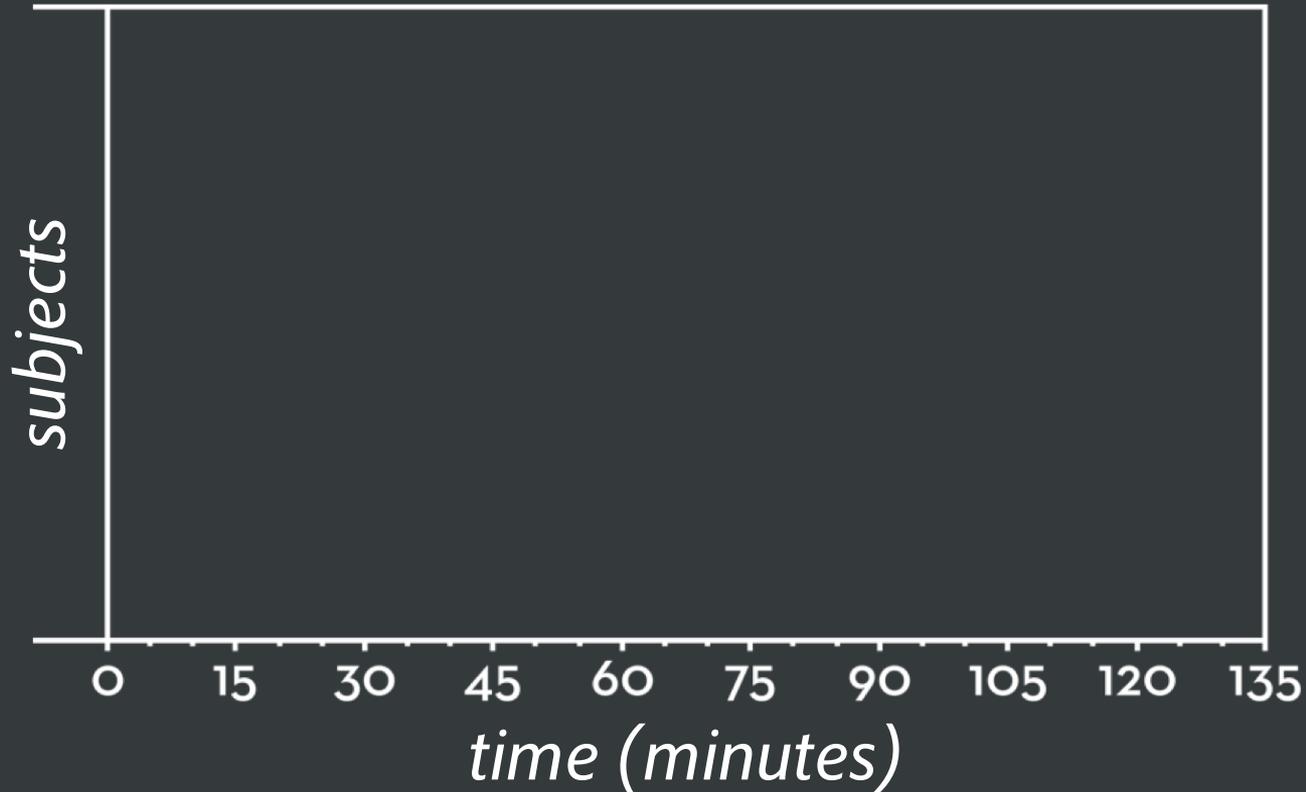
The image shows a screenshot of an Emacs editor window. The title bar at the top reads "Emacs@visavis.local". The main editing area contains the following HTML code:

```
<h1>My Awesome Chatroom</h1>
<form name="input" action="html_form_submit.asp" method="get">
<textarea></textarea><br>
Username:
<input type="text" name="user" />
<br>Message:
<input type="text" name="msg" />
<input type="submit" value="Submit" />
</form>
```

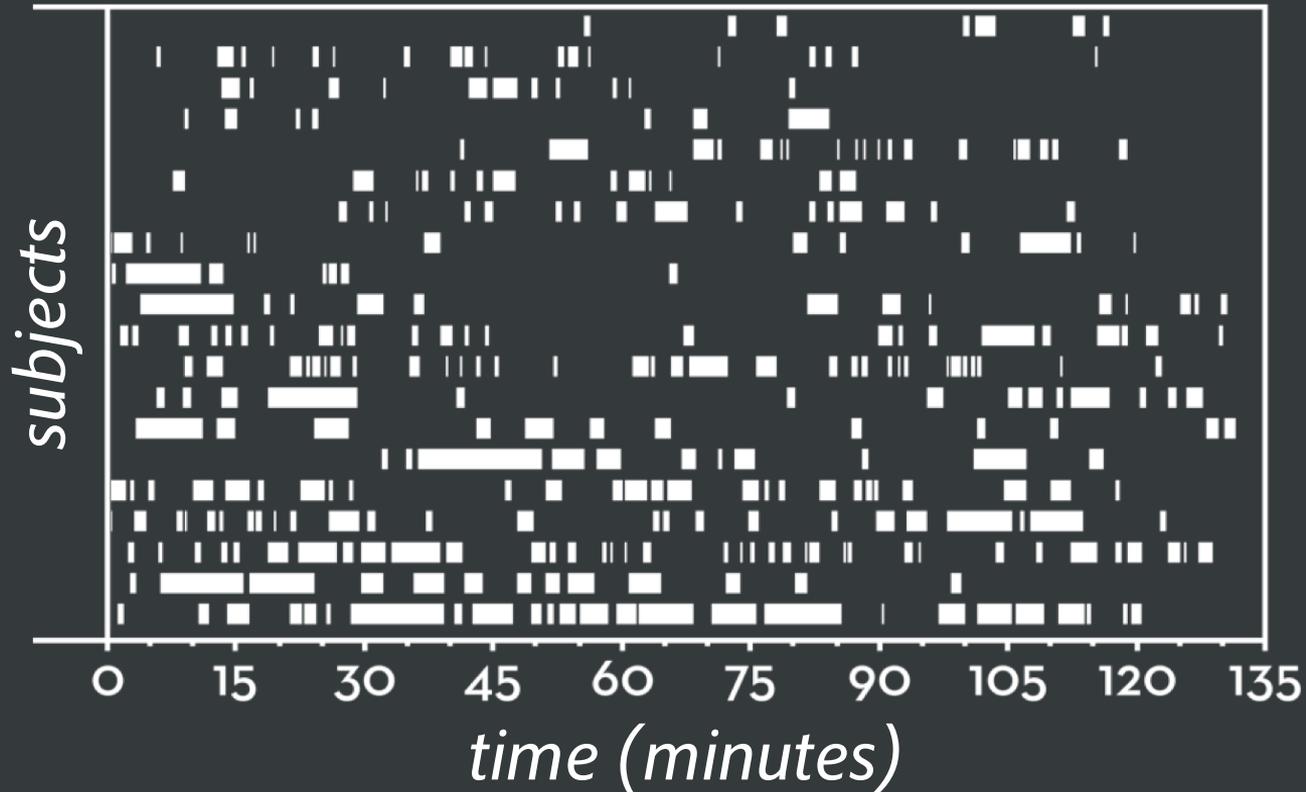
The status bar at the bottom of the window displays: "-1:-- chat.php All (1,28) (PHP Abbrev)".



Frequent Information Access



Frequent Information Access



Frequent Information Access

19% of time spent using Web
($\sigma = 15.1$ minutes)

Used for 18 distinct tasks
($\sigma = 9.1$)

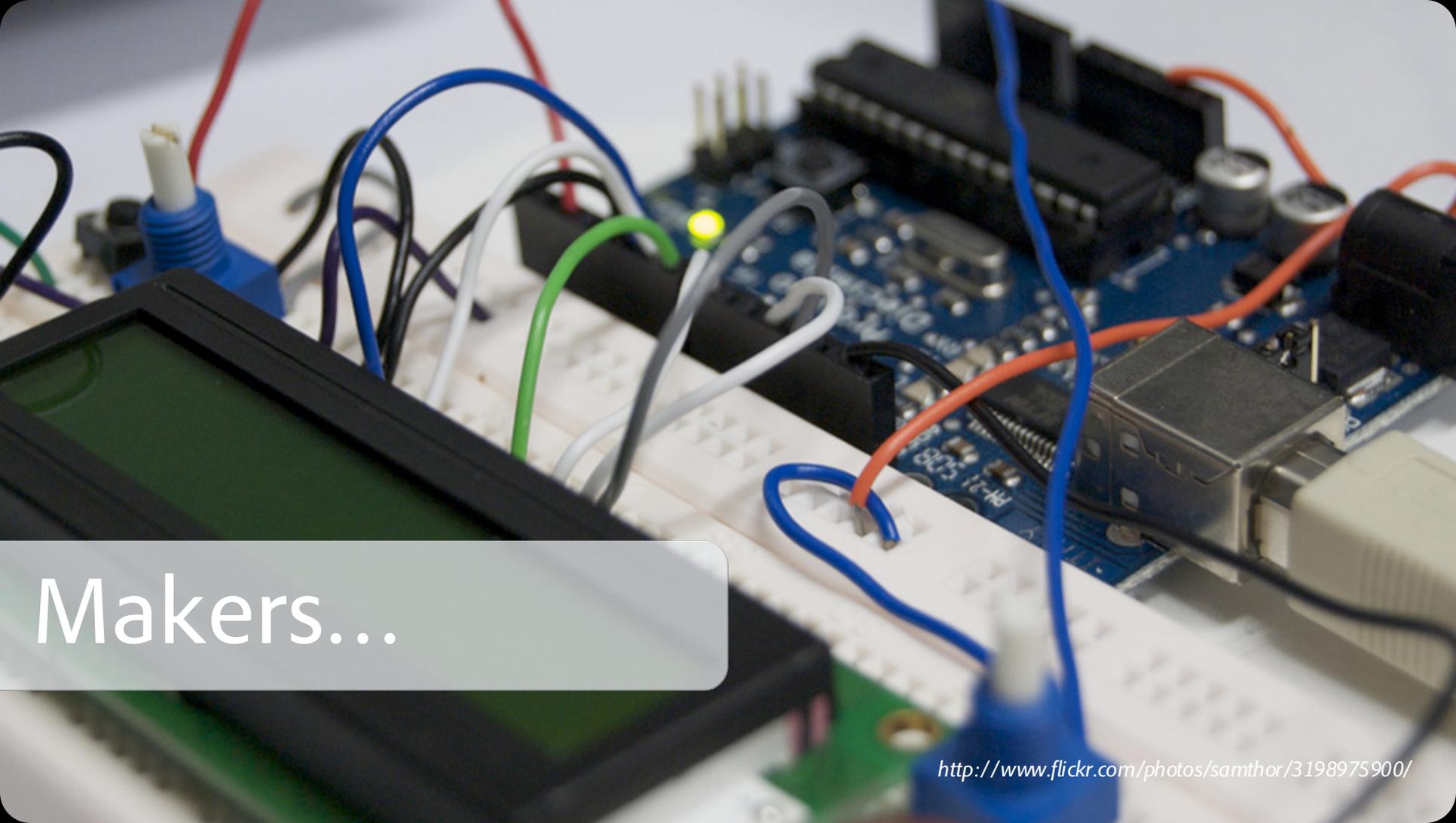
Opportunistic Programmers



Designers...



Scientists...



Makers...

By 2012:

13 million who program as
part of their job...

...but only 3 million who are
professional programmers

*Vast increase in
information resources*

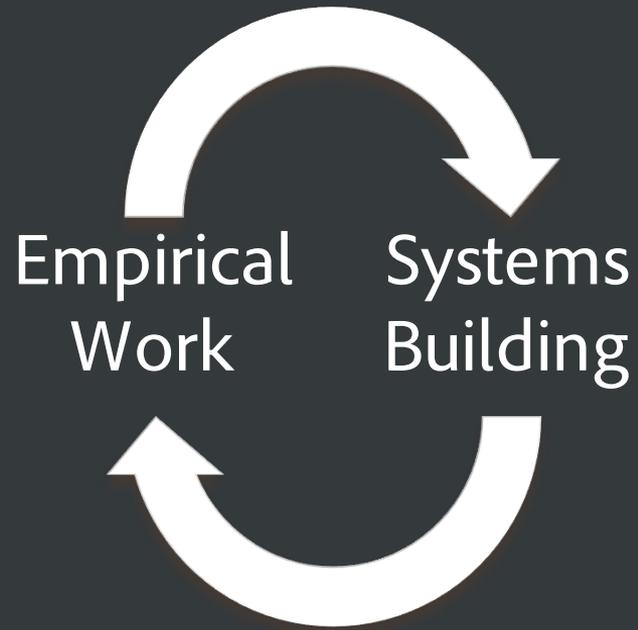
*Democratization of
access and distribution*

*How do we redesign
the programming experience?*

just-in-time learning is common

examples are important

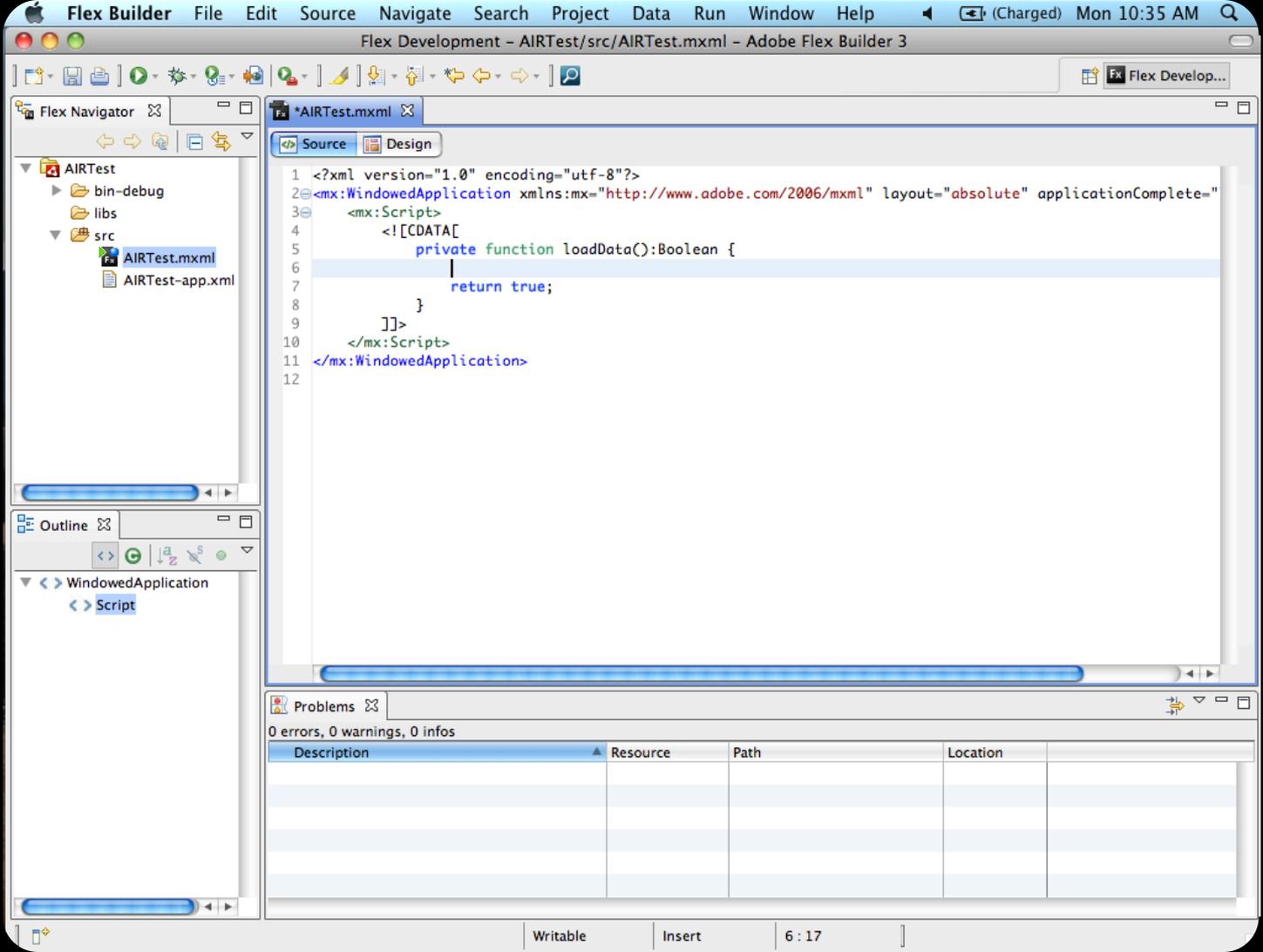
[Brandt CHI 2009]



Blueprint

Code Search in the Editor

[Brandt CHI 2010]



Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4     <![CDATA[
5         private function loadData():Boolean {
6             busy cursor|
7         }
8     ]]>
9 </mx:Script>
10 </mx:WindowedApplication>
11
12
```

Problems

0 errors, 0 warnings, 0 infos

Description

Resource

Path

Location

Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4     <![CDATA[
5         private function loadData():Boolean {
6             busy cursor
7             return true
8         }
9     ]]>
10 </mx:Script>
11 </mx:WindowedApplication>
12
```

busy cursor



Problems

0 errors, 0 warnings, 0 infos

Description

Resource

Path

Location

```
1 <?xml version="1.0"
2 <mx:WindowedApplicat
3 <mx:Script>
4 <![CDATA[
5     private
6     busy
7     retu
8     }
9     ]]>
10 </mx:Script>
11 </mx:WindowedApplica
12
```

busy cursor

busy cursor

Flex 3 - Using a busy cursor

http://livedocs.adobe.com/flex/3/html/cursormgr_4.html

You can modify the example in [Creating and removing a cursor](#) to use the default busy cursor. The following example shows:

```
<?xml version="1.0"?>
<!-- cursors\DefBusyCursorApp.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">

    <mx:Script>
        <![CDATA[
            import mx.managers.CursorManager;
            import flash.events.*;

            private function onEnter(event:Event):void {
                CursorManager.setBusyCursor();
            }

            private function onExit(event:Event):void {
                CursorManager.removeBusyCursor();
            }
        ]]>
    </mx:Script>
</mx:Application>
```

Source Design

```
1 <?xml version="1.0"
2 <mx:WindowedApplicat
3 <mx:Script>
4 <![CDATA[
5     private
6     busy
7     retu
8     }
9     ]]>
10 </mx:Script>
11 </mx:WindowedApplica
12
```

Problems 0 errors, 0 warnings, 0 infos
Description

busy cursor

busy cursor

Flex 3 - Using a busy cursor

http://livedocs.adobe.com/flex/3/html/cursormgr_4.html

You can modify the example in Creating and removing a cursor to use the default busy cursor following example shows:

```
<?xml version="1.0"?>
<!-- cursors\DefBusyCursorApp.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">

    <mx:Script>
        <![CDATA[
            import mx.managers.CursorManager;
            import flash.events.*;

            private function loadComplete(event:Event):void {
                CursorManager.setBusyCursor();
            }

            private function loadComplete(event:Event):void {
                CursorManager.removeBusyCursor();
            }
        ]]>
    </mx:Script>
</mx:Application>
```

Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4   <![CDATA[
5     private function scrollData():Boolean {
6       //@query:'busy cursor'
7       //@source: http://livedocs.adobe.com/flex/3/html/cursormgr_4.html
8       CursorManager.setBusyCursor();
9     }
10    return true;
11  }
12  ]]>
13 </mx:Script>
14 </mx:WindowedApplication>
15
```

Problems

0 errors, 0 warnings, 0 infos

Description

Resource

Path

Location

Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4     <![CDATA[
5         private function loadData():Boolean {
6             //@query:'busy cursor'
7             //@source: http://livedocs.adobe.com/flex/3/html/cursormgr_4.html
8             CursorManager.setBusyCursor();
9
10
11
12             return true;
13         }
14     ]]>
15 </mx:Script>
16 </mx:WindowedApplication>
17
```

Problems

0 errors, 0 warnings, 0 infos

Description

Resource

Path

Location

Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4   <![CDATA[
5     private function loadData():Boolean {
6       //@query:'busy cursor'
7       //@source: http://livedocs.adobe.com/flex/3/html/cursormgr_4.html
8       CursorManager.setBusyCursor();
9
10      URL
11
12      return true;
13    }
14  ]]>
15 </mx:Script>
16 </mx:WindowedApplication>
17
```

Problems

0 errors, 0 warnings, 0 infos

Description

Resource

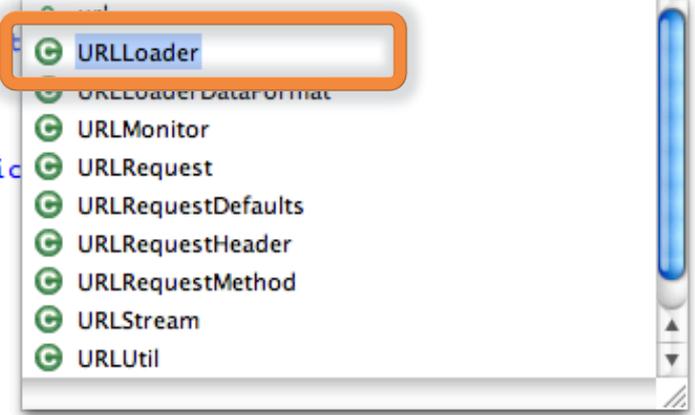
Path

Location

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4     <![CDATA[
5         private function loadData():Boolean {
6             //@query:'busy cursor'
7             //@source: http://livedocs.adobe.com/flex/3/html/cursormgr_4.html
8             CursorManager.setBusyCursor();
9         }
10     ]>
11     URL
12     ret
13     }
14 ]>
15 </mx:Script>
16 </mx:WindowedApplic
17
```

- url
- URLLoader
- URLLoaderDataFormat
- URLMonitor
- URLRequest
- URLRequestDefaults
- URLRequestHeader
- URLRequestMethod
- URLStream
- URLUtil

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4   <![CDATA[
5     private function loadData():Boolean {
6       //@query:'busy cursor'
7       //@source: http://livedocs.adobe.com/flex/3/html/cursormgr_4.html
8       CursorManager.setBusyCursor();
9     }
10  }
11  URL
12  re
13  }
14  ]]>
15 </mx:Script>
16 </mx:WindowedApplic
17
```



- URLLoader
- URLLoaderDataFormat
- URLMonitor
- URLRequest
- URLRequestDefaults
- URLRequestHeader
- URLRequestMethod
- URLStream
- URLUtil

Source Design

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:WindowedApplication xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute" applicat
3 <mx:Script>
4     <![CDATA[
5         private function loadData():Boolean {
6             //@query:'busy cursor'
7             //@source: http://livedocs.adobe.com/flex/3/html/cursormgr_4.html
8             CursorManager.setBusyCursor();
9
10            URLLoader|
11
12            return true;
13        }
14    ]]>
15 </mx:Script>
16 </mx:WindowedApplication>
17
```

Problems

0 errors, 0 warnings, 0 infos

Description

Resource

Path

Location

```

1 <?xml version="1.0"
2 <mx:WindowedApplicat
3 <mx:Script>
4 <![CDATA[
5     private
6         //@q
7         //@s
8         Curs
9
10        URLL
11
12        retu
13    }
14    ]]>
15 </mx:Script>
16 </mx:WindowedApplica
17

```

URLLoader

URLLoader

flash.net.URLLoader

<http://livedocs.adobe.com/flex/3/langref/flash/net/URLLoader.html>

Flex 3 - Working with external data

http://livedocs.adobe.com/flex/3/html/17_Networking_and_communications_3.html

Warning: The code example below may contain security vulnerabilities. To learn more about writing Flash/Flex code, please read the [following article](#)

For example, to upload an XML packet to a server-side script, you could use the following ActionScript 3.0 code:

```

var secondsUTC:Number = new Date().time;
var dataXML:XML =
    <login>
        <time>{secondsUTC}</time>
        <username>Ernie</username>
        <password>guru</password>
    </login>;
var request:URLRequest = new URLRequest("http://www.yourdomain.com/login");
request.contentType = "text/xml";
request.data = dataXML.toXMLString();
request.method = URLRequestMethod.POST;
var loader:URLLoader = new URLLoader();
try

```

```

1 <?xml version="1.0"
2 <mx:WindowedApplicat
3 <mx:Script>
4 <![CDATA[
5     private
6     //@q
7     //@s
8     Curs
9
10    URLL
11
12    retu
13    }
14    ]>
15 </mx:Script>
16 </mx:WindowedApplica
17

```

URLLoader

URLLoader

Doc

Flex 3 - Working with external data

http://livedocs.adobe.com/flex/3/html/17_Networking_and_communications_3.html

Warning: The code example below may contain security vulnerabilities. To learn more about writing Flash/Flex code, please read the [following article](#)

The following code demonstrates how setting the `URLLoader.dataFormat` property to `URLLoaderDataFormat.VARIABLES` allows you to automatically parse loaded data into a `URLVariables` object:

```

package
{
    import flash.display.Sprite;
    import flash.events.*;
    import flash.net.URLLoader;
    import flash.net.URLLoaderDataFormat;
    import flash.net.URLRequest;

    public class URLLoaderDataFormatExample extends Sprite
    {
        public function URLLoaderDataFormatExample()
        {
            var request:URLRequest = new URLRequest("http://www.[yourdo
            var variables:URLVariables = new URLVariables();

```

Source Design

```
1 <?xml version="1.0"
2 <mx:WindowedApplicat
3 <mx:Script>
4 <![CDATA[
5     private
6         //@q
7         //@s
8         Curs
9
10    URLL
11
12    retu
13    }
14    ]]>
15 </mx:Script>
16 </mx:WindowedApplica
17
```

URLLoader

URLLoader

```
public class ExternalDocs extends Sprite
{
    public function ExternalDocs()
    {
        var request:URLRequest = new URLRequest("http://www.[yourdo
        var loader:URLLoader = new URLLoader();
        loader.addEventListener(Event.COMPLETE, completeHandler);
        try
        {
            loader.load(request);
        }
        catch (error:ArgumentError)
        {
            trace("An ArgumentError has occurred.");
        }
        catch (error:SecurityError)
        {
            trace("A SecurityError has occurred.");
        }
    }
    private function completeHandler(event:Event):void
    {
        var dataXML:XML = XML(event.target.data);
        trace(dataXML.toXMLString());
    }
}
```

Problems

0 errors, 0 warnings, 0 infos

Description

busy_cursor

back

Dock this query

gr_4.html

You can modify the example in Creating and removing a cursor to use the default busy cursor, as the following example shows:

```
<?xml version="1.0"?>
<!-- cursors\DefBusyCursorApp.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">

  <mx:Script>
    <![CDATA[
      import mx.managers.CursorManager;
      import flash.events.*;

      private function loadImage(event:Event):void {
        CursorManager.setBusyCursor();
        image1.load("img_4.png");
      }

      private function loadComplete(event:Event):void {
        CursorManager.removeBusyCursor();
      }
    ]]>
  </mx:Script>

  <mx:VBox>
    <!-- Image control to load the image. -->
    <mx:Image id="image1"
      height="50"
    />
  </mx:VBox>
</mx:Application>
```

busy cursor

← back

busy cursor Dock this query

```
width="100"
scaleContent="true"
complete="loadComplete(event);"/>

<!-- Button triggers the load. -->
<mx:Button id="myButton" label="Show" click="initImage(e"
</mx:VBox>
</mx:Application>
```

The executing SWF file for the previous example is shown below:



The image shows a pixelated hourglass cursor icon centered on a blue background. The hourglass is black with white details, and it has a small blue speech bubble above it. The entire scene is enclosed in a thick orange border.

Write code faster in directed tasks...

57 vs. 121

seconds to first code paste

Wilcoxon-Mann-Whitney statistic = 2.38, $p < .01$,

Task completion time strongly
correlated with time of first paste

Spearman correlation coeff. = 0.52, $p < .01$

...and the code is better!



Wilcoxon-Mann-Whitney statistic = 2.15, $p < .02$

● *blueprint* ● *control*

milliseconds matter

[Brutlag 2009, O'Hara 1998]

Codelets

Interactive Examples

*In collaboration with Stephen Oney and Brad
Myers at Carnegie Mellon University*

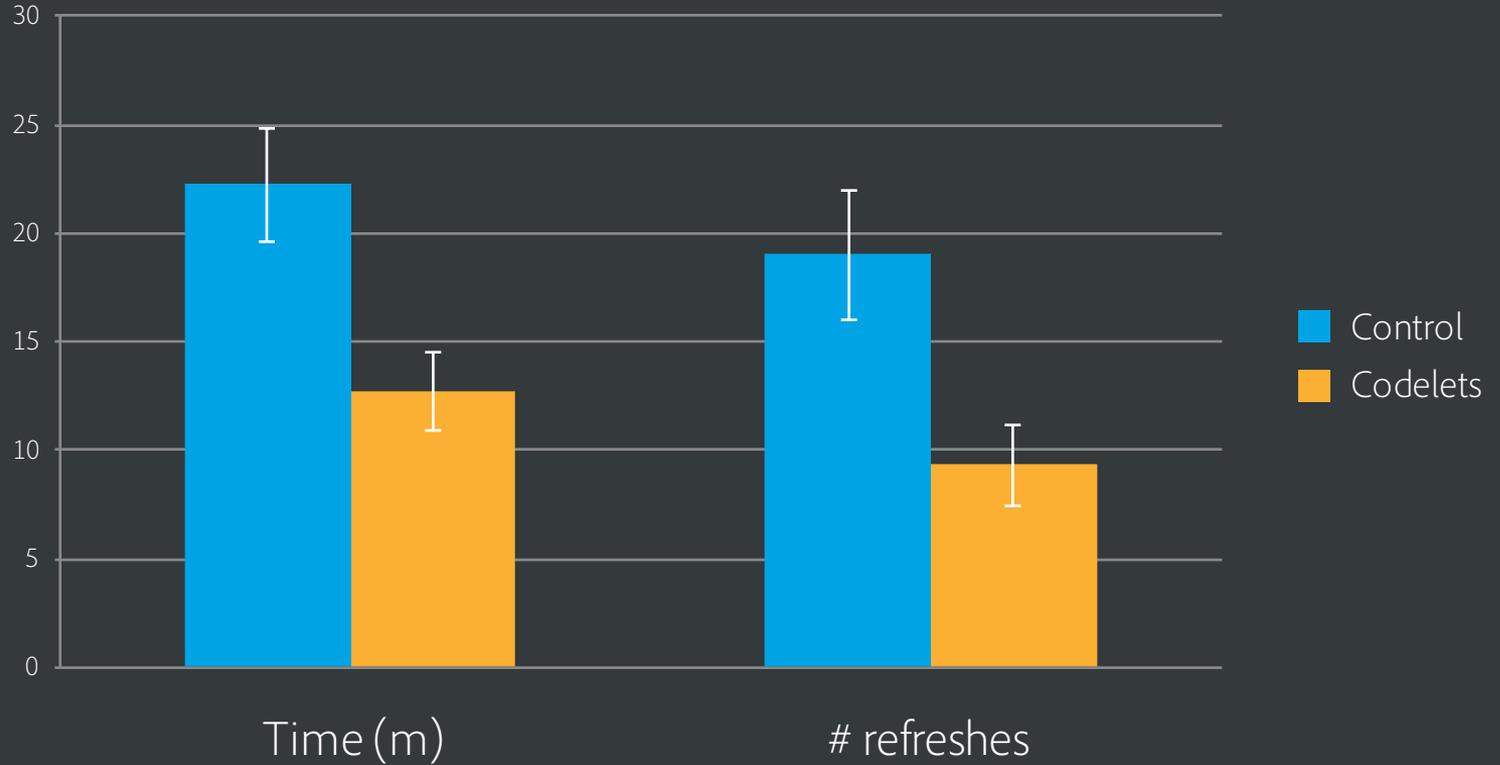
```
1 <html>
2   <head>
3
4   </head>
5   <body>
6     |
7   </body>
8 </html>
```

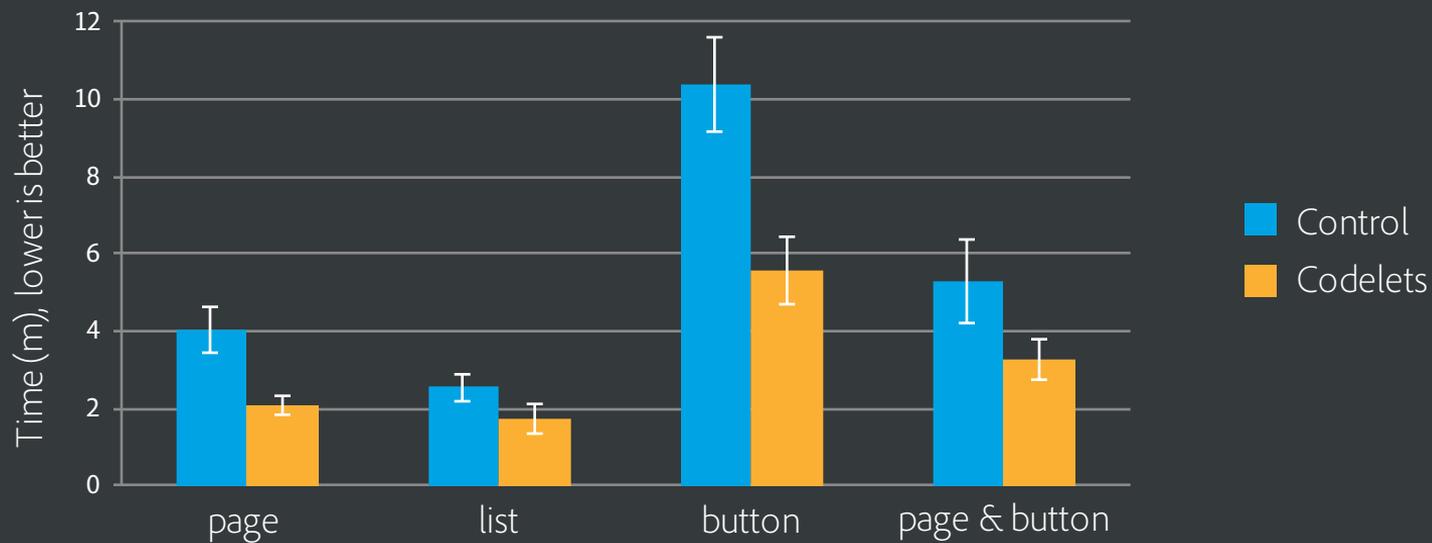
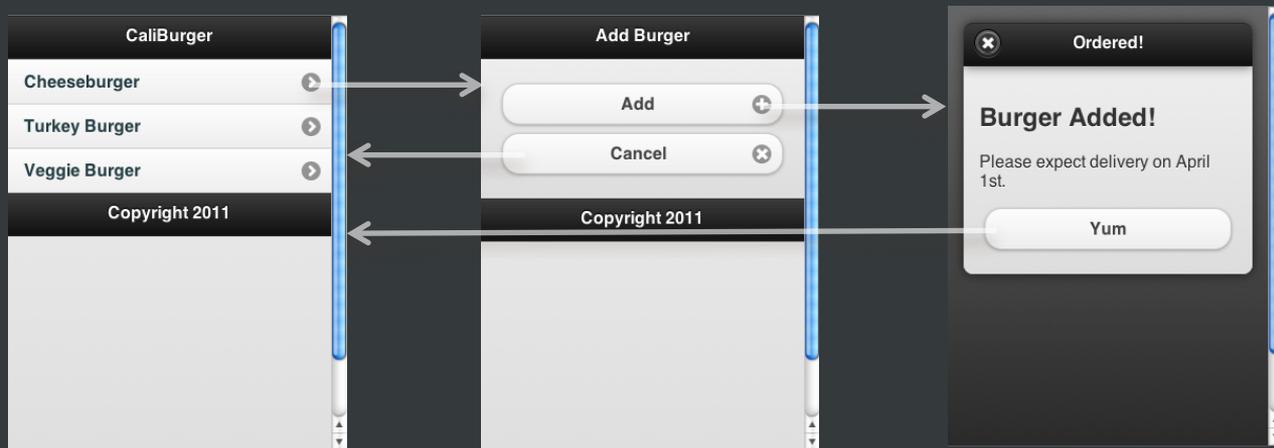


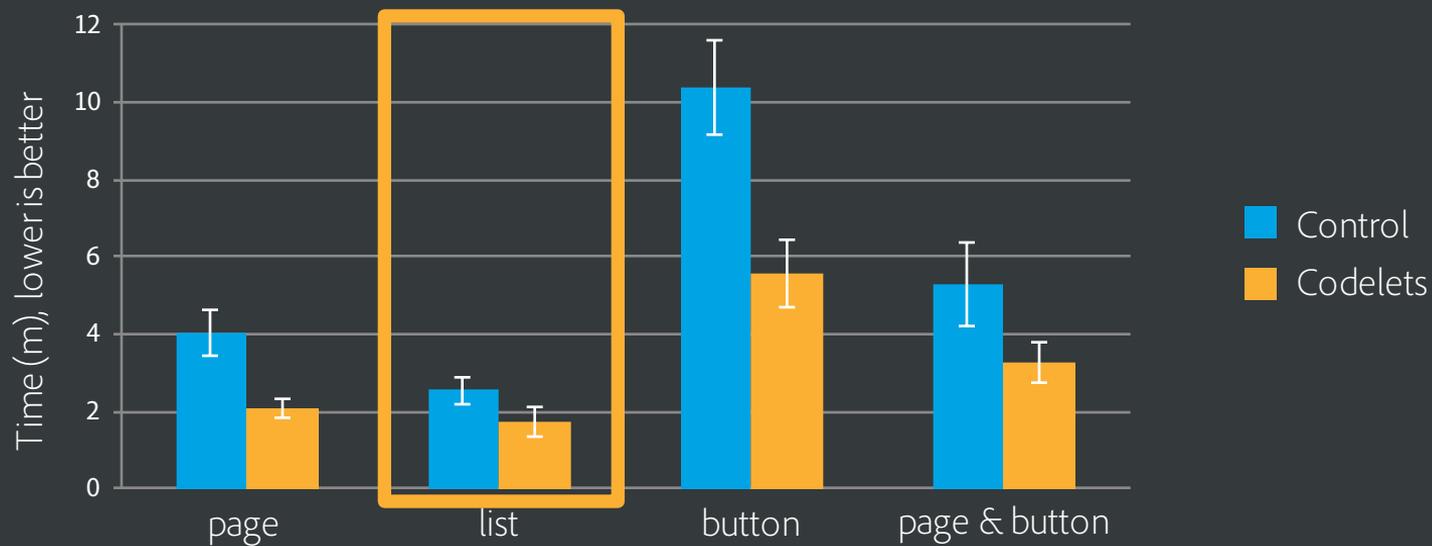
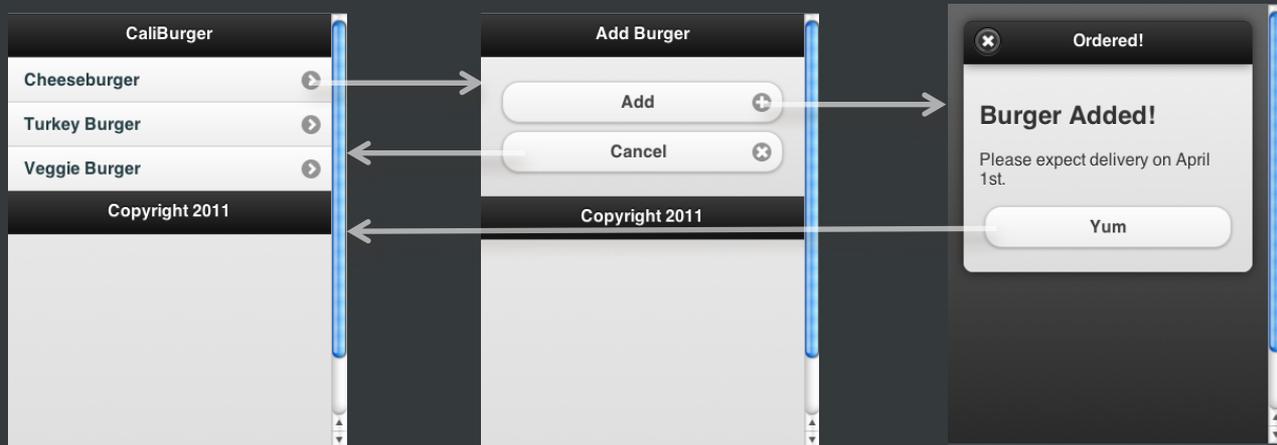
User Study: Method

- 20 participants
- conducted remotely
- between subjects
 - control group: pointed to jQM docs
 - treatment group: used exemplar examples
- two parts (will only present part one)

User Study: Results



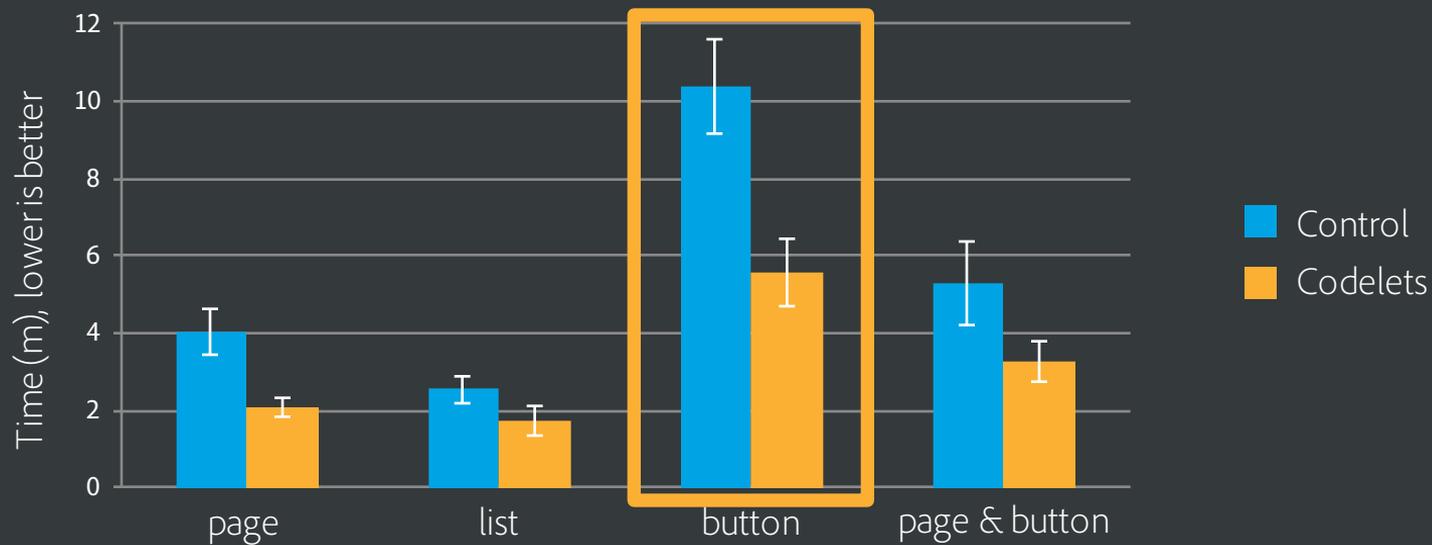
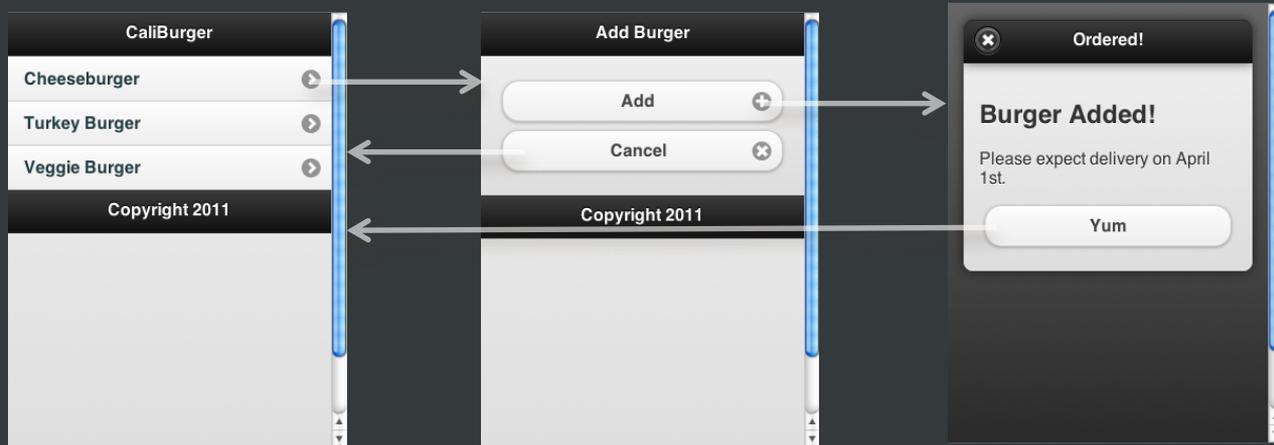




```
<ul data-role="listview">
  <li><a href="acura.html">Acura</a></li>
  <li><a href="audi.html">Audi</a></li>
  <li><a href="bmw.html">BMW</a></li>
</ul>
```

jQuery Mobile Lists

A list view is coded as a simple unordered list containing linked list items with a `data-role="listview"` attribute. jQuery Mobile will apply all the necessary styles to transform the list into a mobile-friendly list view with right arrow indicator that fills the full width of the browser window. When you tap on the list item, the framework will trigger a click on the first link inside the list item, issue an AJAX request for the URL in the link, create the new page in the DOM, then kick off a page transition. Here is the HTML markup for a basic linked list.



```
<a data-role="button" href="" data-icon='plus' data-iconpos='right' data-rel='dialog' >Add</a>
```



jQuery Mobile Buttons

Syntax
Preview

Buttons that are used for navigation should be coded as anchor links, and those that submit forms as button elements - each will be styled identically by the framework.

Icon:



Placement:

Left Right Top Bottom

Text: Hidden

Open as:

Dialog Reverse Transition

Codelets users

- did not read accompanying text
- had two usage patterns
 - leaving codelet open
 - closing codelet and then editing code

Codelets API

- Editor exposes API
- 3rd parties implement examples
 - Not implemented by editor
- Uses HTML & JS
 - What most examples are already in

let's implement one:

```
1  
2  
:  
:  
:  
3
```

```
int x = 1;
```

Setting a Variable

Name :

Value:

```
int v = 2;
```

```
<codelet>
```

```
<head>
```

meta-information & interactivity (Javascript)

```
</head>
```

```
<body>
```

example code & documentation (HTML)

```
</body>
```

```
</codelet>
```

```
<codelet>
```

```
<head>
```

```
<title>    Creating an int </title>
```

```
<keywords> int          </keywords>
```

```
<type>    block         </type>
```

```
<lang>    c             </lang>
```

(meta-information)

interactivity (Javascript)

```
</head>
```

```
<body>
```

example code & documentation (HTML)

```
</body>
```

```
</codelet>
```

<codelet>

<head>

```
<title>    Creating an int </title>
<keywords> int          </keywords>
<type>    block        </type>
<lang>    c            </lang>                (meta-information)
```

interactivity (Javascript)

</head>

<body>

```
<example>
  int <mark id="name" /> = <mark id="value" />;
</example>                (example code)
```

documentation (HTML)

</body>

</codelet>

<codelet>

<head>

```
<title>    Creating an int </title>
<keywords> int          </keywords>
<type>    block         </type>
<lang>    c             </lang>                (meta-information)
```

interactivity (Javascript)

</head>

<body>

```
<example>
  int <mark id="name" /> = <mark id="value" />;
</example>                (example code)
```

```
<page>
  Name : <input type="text" id="name_inp" /> <br />
  Value: <input type="text" id="val_inp" />
</page>                    (documentation)
```

</body>

</codelet>

<codelet>

<head>

```
<title>    Creating an int </title>
<keywords> int          </keywords>
<type>    block         </type>
<lang>    c              </lang>                (meta-information)
```

```
<script>$(function() {
    attach_input_to_mark($(“input#name_inp”), “name” );
    attach_input_to_mark($(“input#val_inp ”), “value” );
});</script>                                (interactivity)
```

</head>

<body>

```
<example>
    int <mark id=“name” /> = <mark id=“value” />;
</example>                                (example code)
```

```
<page>
    Name : <input type=“text” id=“name_inp” /> <br />
    Value: <input type=“text” id=“val_inp” />
</page>                                (documentation)
```

</body>

</codelet>

```
<codelet>
```

```
  <head>
```

```
    <title>    Creating an int </title>
```

```
    <keywords> int          </keywords>
```

```
    <type>     block        </type>
```

```
    <lang>     c            </lang>
```

```
    <script>$(function() {
```

```
      attach_input_to_mark($(“input#name_inp”), “name” );
```

```
      attach_input_to_mark($(“input#val_inp ”), “value” );
```

```
    });</script>
```

```
  </head>
```

```
  <body>
```

```
    <example>
```

```
      int <mark id=“name” /> = <mark id=“value” />;
```

```
    </example>
```

```
    <page>
```

```
      Name : <input type=“text” id=“name_inp” /> <br />
```

```
      Value: <input type=“text” id=“val_inp” />
```

```
    </page>
```

```
  </body>
```

```
</codelet>
```

In the lab, participants complete tasks involving example code 43% faster

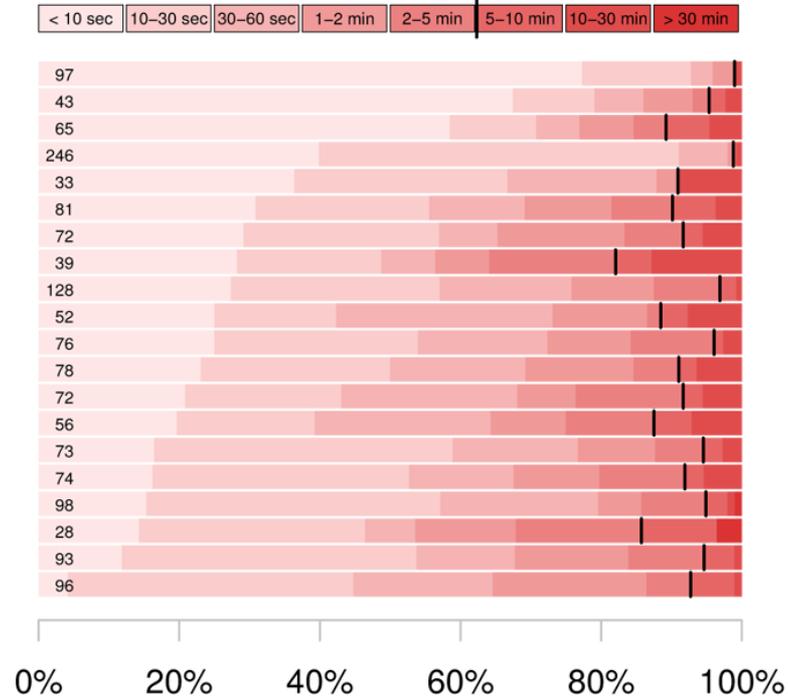
Examples have a lifetime: they're useful throughout the entirety of a project

Theseus

Always-On JavaScript Debugging

In collaboration with Thomas Lieber and Rob Miller at MIT

Programmers iterate rapidly



Search:

name-directory ▾

- index.html
- jquery-1.7.js
- spinner.gif

```

1 <!doctype html>
2 <html>
3 <head>
4 <title>Company Directory</title>
5
6 <script src="jquery-1.7.js?prebug=no"></script>
7 <script>
8 (function () {
9   'use strict';
10  var resultsDiv;
11  var spinnerDiv;
12
13  var searchUrl = "http://127.0.0.1:3000/search";
14
15  var search = function (term, callback) {
16    $.getJSON(searchUrl, {s: term}, function () {
17      callback.apply(this, arguments);
18    });
19  };
20
21  var hideSpinner = function () {
22    spinnerDiv.hide();
23  };
24
25  var term = "";
26  var keyPressed = function () {
27    if ($(this).val() !== term) {
28      var thisTerm, i;
29      term = $(this).val();
30
31      if (term.length > 0) {
32        spinnerDiv.show();
33        resultsDiv.empty();
34        search(term, function (results) {
35          if (term === thisTerm) {
36            hideSpinner();
37            results.forEach(function(result) {
38              $("<li />").appendTo(resultsDiv).text(result);
39            });
40          }
41        });
42      } else {
43        resultsDiv.empty();
44      }
45    }
46  });

```

Line 15, Column 12 64 Lines HTML Spaces 4



index.html

async-fs.js

sync-fs.js

main.js

config.js

async-filesystem ▾

async-fs.js

config.js

index.html

▶ lib

main.js

mockfs.js

output.js

style.css

sync-fs.js

async-fs.js



```
1  /*jshint vars: true, plusplus: true, devel: true, nomen: true, indent: 4, maxerr:
2  /*global $, define */
3
4  /*
5   * async-fs.js
6   */
7
8  define(function (require, exports, module) {
9      "use strict";
10
11     var fs    = require("mockfs");
12     var output = require("output");
13     var config = require("config");
14
15     function listDirectory() {
16         var result = [];
17
18         function statAndInsertHelper(filename, position) {
19             var sp = fs.statPromise(filename);
20
21             sp.fail(function (error) {
22                 result[position] = {name: filename, error: error};
23                 output.log(result);
24             });
25
26             sp.done(function (type) {
27                 result[position] = {name: filename, type: type};
28                 output.log(result);
29             });
30         }
31     }
32
33     var lp = fs.listPromise(config.TEST_DIRECTORY);
34
35     lp.fail(function (error) { console.error(error); });
36
37     lp.done(function (filenames) {
38         filenames.forEach(statAndInsertHelper);
39     });
40
41 }
42
43 exports.listDirectory = listDirectory;
44
45 });
46
```

Debugging is hard because user's mental model is different than program logic

Always-on visualizations help programmers build & maintain an accurate mental model

Vesta

Using Runtime Traces to Improve Documentation and Testing

*In collaboration with Jan-Peter Krämer and Jan Borchers
at RWTH Aachen University*



```
1
  Add some documentation!

7 function swap (arr, a, b) {
8   var tmp = arr[a];
9   arr[a] = arr[b];
10  arr[b] = tmp;
11
12  return arr;
13 }
14
  Add some documentation!

20 function sort (arr) {
21   var change;
22   do {
23     change = false;
24     for (var i = 1; i<arr.length; i++) {
25       if (arr[i-1] > arr[i]) {
26         change = true;
27         swap(arr, i-1, i);
28       }
29     }
30   } while (change);
31
32   return arr;
33 }
34
35 var theArray = [5,3,5,2,3];
36 console.log("Unsorted: " + theArray);
37 var result = sort(theArray);
38 console.log("Sorted: " + result);
39
40
41 exports.sort = sort;
42 exports.swap = swap;
43
```

Autogenerated Suites for Current Document ▾

Current Function ▾

Spec Closure

beforeEach

+ Add Test

afterEach



All documentation created with Vesta was correct,
compared to only 60% in control condition

Vesta could generate significant portions of all unit tests,
including tests that were written in control condition

Blueprint

Codelets

Theseus

Vesta

Growing importance on fast information
access and just-in-time skill acquisition

In the future, knowledge work may become less about knowing how to do a task, and more about knowing how to find the right answer.

In the future, knowledge work may become less about knowing how to do a task, and more about knowing how to find the right answer.

*How can the
software visualization community
help opportunistic programmers?*

Build interfaces that assume the programmer knows nothing about the codebase

Help professionals build software
that is easy for non-professionals to
augment

Make opportunistic programmers'
building blocks first-class citizens
of their development tools

Focus collaboration tools on
learning rather than *building and*
maintaining software

Remember that you're not
designing for yourself



Adobe