

CS 5142

Scripting Languages

10/11/2013

jQuery

Announcements

- Industry-Academia Workshop
10/18/13:
[http://www.industry-academia.org/
event-cornell-workshop.html](http://www.industry-academia.org/event-cornell-workshop.html)

Closures?

- *A closure* is a function and an environment
- Environment defines non-local variables

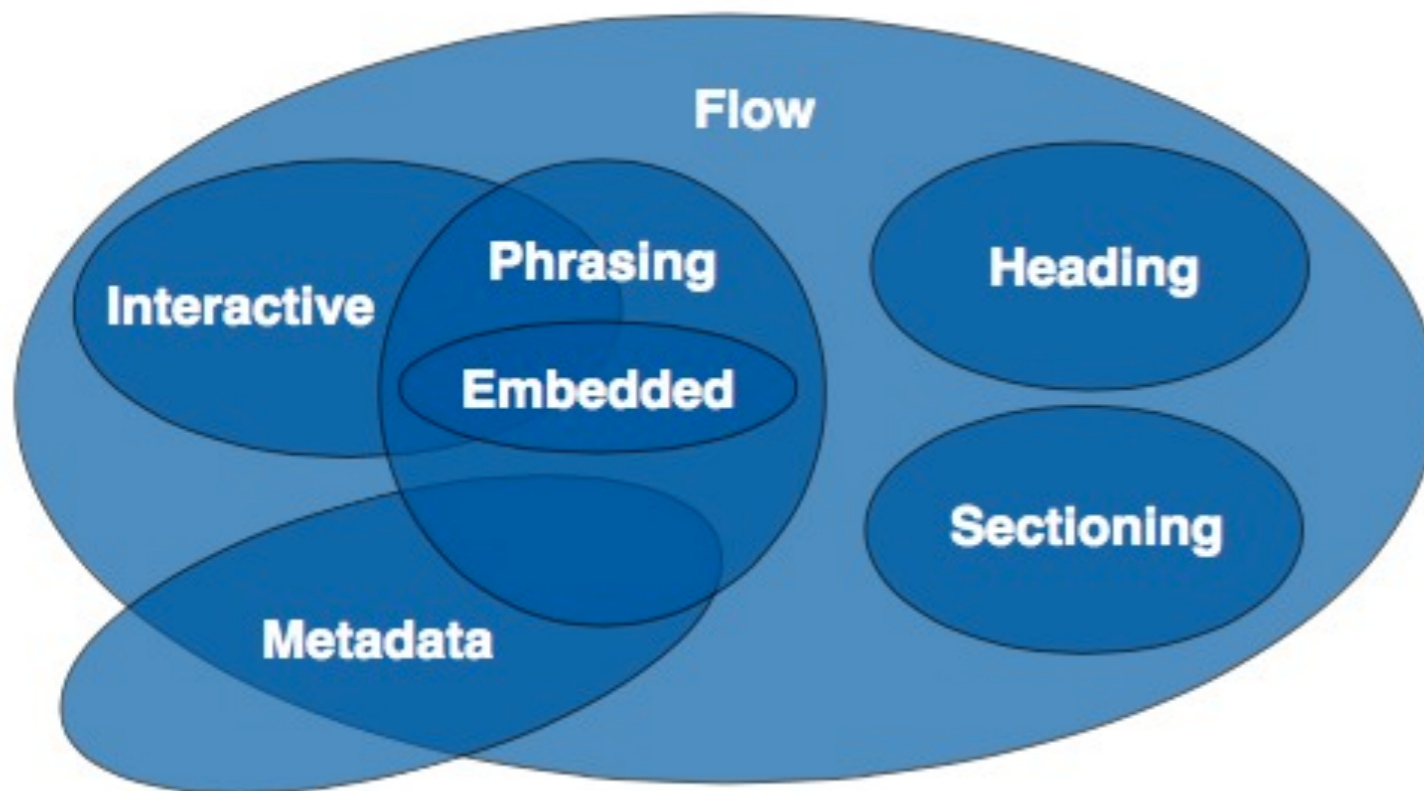
HTML (again)

```
Aquamacs - fruit.html
<html>
  <head>
    <title>Fruitlation</title>
  </head>
  <body>
    <!-- this is a comment -->
    <h1>Translation Table</h1>
    <table border=5>
      <tr><th>English</th> <th>German</th></tr>
      <tr><td>apple</td> <td>Apfel</td></tr>
      <tr><td>pear</td> <td>Birne</td></tr>
    </table>
    <hr size=3 color="red"/>
    Some symbols: &lt;, &gt;, &amp;<br/>
    <a href="index.html">a hyperlink</a>
  </body>
</html>
```

-:-- fruit.html All (18,1) (HTML helper)

- Comment `<!-- ... -->`
- Element `<table>...</table>`
- Start tag `<table>`
- Attribute `border=5`
- Contents (between matching tags)
- End tag `</table>`
- Empty element `<hr .../>`
- Entity `&`

HTML Content



- **Heading:** h1, h2
- **Sectioning:** article, nav, section
- **Phrasing:** abbr, img, textarea, cite, button
- **Embedded:** audio, canvas, iframe, math
- **Interactive:** a, button, label, select
- **Metadata:** base, link, meta, style
- **Flow:** most elements in the <body>

Source: <http://www.w3.org>

HTML 5

- Replaces XML, XHTML, older HTML
- How do we tell the browser we are using HTML 5?
`<!DOCTYPE html> <html lang="en">`
- Usually must escape special characters
- Use `<meta>` tag!
 - Don't know the character set, so be explicit.
 - If UTF8, can use most characters directly
 - literal less-than '`<`', greater-than '`>`', and ampersand '`&`' characters must *always* be encoded as entities:
`<`, `>`, and `&`;

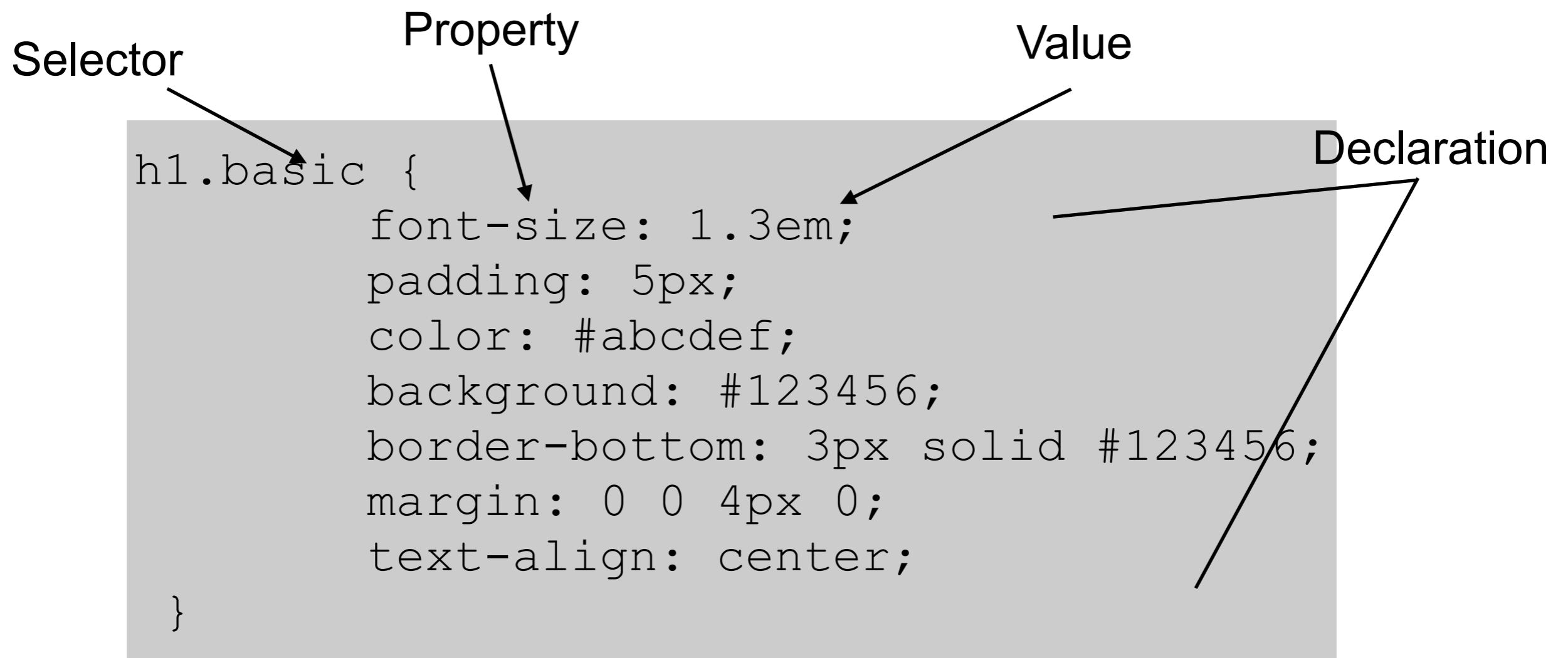
HTML 5

- In previous versions of HTML, block elements (<h1>, <p>, <table>) could not appear inside inline elements (, <td>, <a>).
- In HTML5, there are more kinds of content, so rules are more complicated.
- Can add custom attributes to HTML elements (prefix with 'data-').

```
<li data-spacing="10cm">Apple</li>
```

Cascading Style Sheets

CSS changes the look/format of an HTML document



HTML + CSS

```
<!DOCTYPE HTML><html lang="en">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>CSS Hello World</title>
<link href="hello.css" rel="stylesheet" type="text/css"/>
</head>
<body>
<h1 class="basic">Hello World</h1>
</body>
</html>
```

CSS Selectors

Patterns used to select the elements you want to style

Selector	Example	Description
.class	.intro	elements with class="intro"
#id	#firstname	elements with id = "firstname"
element	p	all p elements
e,e	div,p	all div and all p elements
e > e	div>p	all p elements where parent is div element
e e	div p	all p elements inside div elements

Back to Scripting

- What if I want to programmatically change the behavior/look of my web site?
- JavaScript can be used to modify the DOM, but....

Get Selected Text

```
function getSelectedText() {  
    if (window.getSelection)  
        return window.getSelection().toString();  
    else if(document.getSelection)  
        return document.getSelection();  
    else if(document.selection)  
        return document.selection.createRange().text;  
}
```

- IE before 9: document.selection
- IE 9: document.getSelection()
- Safari, Firefox : window.getSelection()

How Many Children?

```
<book>
  <title>Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
```

```
xmlDoc=loadXMLDoc("books.xml");
x=xmlDoc.documentElement.childNodes;
document.write("Number of child nodes: " + x.length);
```

- IE9 says 4 child nodes
- IE10 and other browsers say 9 child nodes (whitespace is significant!)

Example source: http://www.w3schools.com/dom/dom_mozilla13vs_ie.asp

JavaScript vs. DOM

- JavaScript core API is standardized
 - ECMAScript version 5.1, June 2011
- DOM implementations vary:
 - Different browsers
 - Different browser versions
- Client-side API becomes very complicated

jQuery

- JavaScript library, open-source
- Released in January 2006 by John Resig
- September 2008, Microsoft and Nokia support
- June 2012, used by*:
 - 54.7% of top 10,000 websites
 - 42.4% of top 1,000 sites
 - 24.7% of top 100 sites

*<http://www.pingdom.com>

Install and Use:

Download from <http://jquery.com>, include in the script element:

```
<script src="jquery-1.10.1.min.js"></script>
```

Point to a content distribution network:

```
<script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>  
<script src="http://code.jquery.com/jquery-migrate-1.2.1.min.js"></script>
```


jQuery Function

- *A factory* method for a jQuery object

- pass CSS selector:

```
var objs = $(":header");
```

- pass DOM object:

```
var objs = $(document);
```

- pass HTML:

```
var objs = $("<html>...</html>");
```

- pass a function:

```
jQuery(function() { ... });
```

jQuery Object

- `$()` returns a jQuery Object
- It is an array-like object with properties: `length`, `selector`, `context`, and `jquery`
- Convert to an actual array with `toArray()`

```
$("body").length // 1
```

```
var bodyscript = $("script", document.body);  
bodyscript.selector // "script"  
bodyscript.context // document.body  
bodyscript.jquery // 1.10.2
```

Query Results

```
$(":header").each(function() { alert(this.id); });
```

- The each method expects a *callback* as its argument
- Callback is invoked as a method on each matched element
- In the callback, `this` refers to an Element object
- If callback returns `false`, then early exit (like `break`)

Query Results

```
var id_array = $( ":header" ).map(  
    function() { return this.id; }).toArray();  
  
for (var i=0;i<id_array.length;i++) {  
    // do something  
}
```

- Works like EcmaScript map
- Callback as argument, invoked once for every element
- Result might not be jQuery objects

Attribute Getters and Setters

- Only one method: `attr()`
- When used as a setter, sets every element in a jQuery object
- When used as a getter, queries only the first element

```
$("form").attr("action"); // get the action attribute  
                        // from the form
```

```
$("#icon").attr("src", "icon.gif"); // set the src  
                                // attribute
```

Element Content Getters and Setters

- `text()` gets plain text content
- `html()` get html content

```
var title = $("head title"); // doc title  
var headline = $("h1").html(); // first h1 element
```

CSS Attribute Setter and Getter

```
var c = $("h1").css("background-color");  
$("h1").css("background-color", "gray");
```

- Similar to attr()
- Returns the current, or “computed” attribute
- Cannot change compound styles (e.g., can change font-weight, but not font)

Event Handlers

Make sure that the DOM is ready for manipulation:

```
$(document).ready(function() { ... });
```

Register simple event handler:

```
$("#h1").click(function() { $this.css("background-color", "gray"); });
```


Generate Events

Behave like a button submit:

```
$( "p" ).submit( );
```

Trigger a generic event:

```
$( "button:first" ).trigger( "click" );
```

Evaluating jQuery

- Uniform interface to DOM
- Widely available, widely used
- Write cross-browser client-side code
- AJAX support (future lectures)

Last Slide

- Next week: Databases!