Web Technologies

Lecture 8

JQuery

JQuery

- *“A fast and concise **JavaScript Library** that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.”* - jQuery.com

- Write less do more

  $("p.neat").addClass(“anim").show("slow");

Advantages

- **Easy of use**
  - Simpler syntax and less lines of code
- **Large library set**
  - Many functions for HTML manipulation, animation, event handling, AJAX, etc.
- **Strong open source community**
  - Many libraries
  - Efficiency and security
- **Great documentation and tutorials**

Disadvantages

- **Functionality <u>may</u> be limited**
  - Despite the many libraries, depending on the website complexity raw Javascript may still be required
- **JQuery file is always required**
  - 25-100KB which need to be loaded with your website
    - Strain on the client
    - Strain on the server (if your website hosts it)
- **Performance**
  - Vanilla Javascript is faster than JQuery
- **Code** is **not always shorter**
  - *$(this).attr("id")* vs. *this.id*

DOM and JQuery

- **Identification**
  - How do I obtain a reference to the node that I want
- **Traversal**
  - How do I move around the DOM tree
- **Node Manipulation**
  - How do I get or set aspects of a DOM node
- **Tree Manipulation**
  - How do I change the structure of the page

Terminology

- **JQuery function**
  - Global jQuery object or the *$ function* depending on the context
- **JQuery object**
  - The object returned by the JQuery function that often represents a group of elements
- **Selected elements**
  - The DOM elements that you have selected, most likely by some CSS selector

JQuery function

- The *$ function* always (even for ID selectors) **returns an array-like objec**t called a JQuery object

  Example:

  document.getElementById("id") == $("#myid")[0];

- The JQuery object **wraps** the originally selected DOM objects
- You can access the actual DOM object by accessing the elements of the jQuery object
- $ **adds** extra **functionality** to DOM elements
- Passing an existing DOM object to $ will give it the jQuery upgrade

  Example:

  var elem = document.getElementById("myelem");

  elem = $(elem);

Node selectors

- Imported from CSS
- **Patterns** used to select the element(s) you want to handle
- http://www.w3schools.com/jquery/jquery_ref_selectors.asp
- http://api.jquery.com/category/selectors/

DOM method	jQuery equivalent
getElementById("id")	$("#id")
getElementsByTagName("tag")	$("tag")
getElementsByName("somename")	$("[name='somename']")
querySelector("selector")	$("selector")
querySelectorAll("selector")	$("selector")

Examples

*	$("*")	All elements
#id	$("#lastname")	The element with id="lastname"
.class	$(".intro")	All elements with class="intro"
.class,.class	$(".intro,.demo")	All elements with the class "intro" or "demo"
element	$("p")	All <p> elements
el1,el2,el3	$("h1,div,p")	All <h1>, and <p> elements

:first	$("p:first")	The first <p> element
:last	$("p:last")	The last <p> element
:even	$("tr:even")	All even <tr> elements
:odd	$("tr:odd")	All odd <tr> elements

parent > child	$("div > p")	All <p> elements that are a direct child of a element
parent descendant	$("div p")	All <p> elements that are descendants of a element
element + next	$("div + p")	The <p> element that are next to each elements
element ~ siblings	$("div ~ p")	All <p> elements that are siblings of a element

Examples

[*attribute=value*]	$("[href='default.htm']")	All elements with a href attribute value equal to "default.htm"
[*attribute!=value*]	$("[href!='default.htm']")	All elements with a href attribute value not equal to "default.htm"
[*attribute$=value*]	$("[href$='.jpg']")	All elements with a href attribute value ending with ".jpg"
[*attribute|=value*]	$("[title|='Tomorrow']")	All elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen
[*attribute^=value*]	$("[title^='Tom']")	All elements with a title attribute value starting with "Tom"
[*attribute~=value*]	$("[title~='hello']")	All elements with a title attribute value containing the specific word "hello"
[*attribute*=value*]	$("[title*='hello']")	All elements with a title attribute value containing the word "hello"

Writing efficient selector

- **Use IDs if possible**
  - **Fast** access
  - $(“#myelement”)
- **Avoid selecting by class only**
  - $(“.myclass”)
  - **Inefficient** in older browsers
- **Keep it simple**
  - No more than **2 or 3 qualifiers** unless you have a very complex HTML
  - **Example:** *$(“p#intro em”)* instead of *$("body #page:first-child article.main p#intro em");*
- **JQuery works from last selector to first**
  - Retrieve best **qualified** selector **first**
  - **Example**: *$("em", $("p#intro"))* instead of *$(“p#intro em”)*

Complex examples

- All p tags that have no children, but only if they don't have a class of ignore

  $(“p:empty:not([class=‘ignore’])”)
- Any p element with the text "REPLACE_ME" in it

  $(“p:contains(‘REPLACE_TEXT’)”)
- All div tags with a child that has a class of special

  $(“div”).children(“.special”)
- All heading elements (h1, h2, h3, h4, h5, h6)

  $(“h1,h2,h3,h4,h5,h6”)
- Every other visible li

  $(“li:even:visible”)

Handling events, CSS, animations

```
$(document).ready(function() {
	$("button").click(function() {
		$("p").css("background-color", "yellow");
	});
});


$("button").click(function() {
	$("div").animate( {
		left: '250px',
		opacity: '0.5',
		height: '150px',
		width: '150px'
	});
});
```

What’s next?

- Server side programming
- Web services
- Cloud computing