Web Technologies — Week 9

Mikheil Rukhaia

International Black Sea University,
Tbilisi, Georgia

December 3, 2015

Outline

Introduction to jQuery

Why jQuery?

- jQuery is the most popular JavaScript library on the web.
- There is dozens of plug-ins, tutorials and sample codes to make it much easier to make websites.
- It provides a concise, friendly, human-readable interface.
- Example: can you guess the meaning?

```
$("#fluent").hide("slow")
            .addClass("fun")
            .show("fast");
```

Including the library

- Basic way to get jQuery is to go to `www.jquery.com` and download the library.
- An alternative is to use a public version served by a Content Delivery Network (CDN).
- CDNs are fast servers spread around the world, so sometimes they speed up things for your users.
- Example:

```
<script src="http://ajax.googleapis.com/ajax/
                libs/jquery/1.8.2/jquery.min.js">
</script>
<script src="js/jquery-1.8.2.min.js"></script>
```

The `$` function

- At the heart of every jQuery operation is the `jQuery` object.
- It can be accessed with two different names: `jQuery` or equivalent alias `$`.
- `$` has two different usages: as a function to access HTML elements and as an object to access additional methods and properties of jQuery.
- For readability, it is recommended to use `$()` as function and `jQuery` as object.

The $ function (ctd.)

- `$()` accepts a string representing a CSS selector and returns the HTML element(s) matching the selector.
- Then `text()` method can be used to get/set the HTML element text.
- Note: jQuery automatically changes every element in the matched set.
- Example:

```
$("#hello").text("Hello world from jQuery!");
$("p").text( $("#hello").text() );
```

Start scripts in the smart way

- Frequently you need to manipulate page as soon as it loads.
- For this purpose there is `$(document).ready()` method, taking an anonymous function as argument.
- Example:

```
$(document).ready(fucntion(){
  $("#hello").text("Hello world from jQuery!");
});
```

`text()` vs `html()`

- jQuery provides `html()` method to get or set HTML content of selected element (similar to `innerHTML`).
- The difference between `text()` and `html()` is that while the first one escapes HTML markup, the latter one interprets it properly.
- Example:

```
$("h1").text("<i> Hello world! </i>");
$("h1").html("<i> Hello world! </i>");
```

Selecting elements

- Except CSS selectors, jQuery implements its own selectors as well.

Selector	Selects	Example
`!=`	attribute not equals selector	`$("input[type!='email']")`
`:eq()`	element at the specified index in the returned set	`$("li:eq(2)")`
`:gt()` `:lt()`	all elements greater / less than the specified index in the returned set	`$("li:gt(2)")` `$("li:lt(2)")`

Selecting elements (ctd.)

Selector	Selects	Example
`:first` `:last`	selects the first / last matched element	`$("tr:first")` `$("tr:last")`
`:even` `:odd`	selects even / odd elements	`$("tr:even")` `$("tr:odd")`
`:hidden` `:visible`	all hidden / visible elements	`$("p:hidden")` `$("p:visible")`
`:has()`	elements containing at least one element that matches the specified selector	`$("div:has(p)")`
`:parent`	all parent elements	`$("p:parent")`
`:selected`	all selected elements	`$("option:selected")`
`:animated`	all jQuery animated elements	`$("p:animated")`

Creating elements

- You can create new elements by passing a string of valid HTML to `$()` function.
- Using `append()` method you can add new elements into existing ones.
- Example:

```
div=$("<div id='greetings'><h1>Hi</h1></div>");
h1 =$("<h1>").text("Hello")
              .attr("id","hello");
$(div).append(h1);
$("body").append(div);
```

Creating elements (ctd.)

- `attr()` has capability to set multiple attributes at once using a plain JavaScript object as a map containing a series of attributes and values.

- `attr()` method has another usage as well: if only the name of attribute is specified, then its value is returned.

- Example:

```
$("p").append($("<a>"));
$("p a").attr({
  "href" : "http://www.example.com/",
  "title" : "example",
  "id" : "example"
});
$("a").text($("a").attr("title"));
```

Some useful methods

- `next()` is used to get the next HTML element in the DOM.
- `val()` is used to get the value of the form fields.
- Example:

```
<p> </p>
<input type="text" />
<input type="button" onclick="put()" />

function put() {
  $("p").text($("p").next().val());
}
```

Working with CSS

- `css()` method works analogously to `attr()` and is used to get/set CSS properties to HTML element.

- Example:

```
$("h1").css({
  "font-size" : "150%",
  "color" : "#ffffff",
  "height" : "100px",
  "width" : "500px",
  "background-color" : "#61b7ff",
  "border" : "10px solid #003366"
});
$("p").text($("h1").css("border"));
```

Working with CSS (ctd.)

- Although you can change styles on fly, it is better to keep style information in CSS files and add or remove classes on user interaction.
- For this purpose there are `addClass()`, `removeClass()`, `hasClass()`, and `toggleClass()` methods.
- All four methods accept a string representing the class or list of classes.
- `hasClass()` cannot be chained because it returns boolean value.

Working with CSS (ctd.)

- Example:

```
$("h1").addClass("selected center");
if ($("h1").hasClass("selected")) {
  $("h1").removeClass("selected");
}

$("h1").on("click", function(){
  $("h1").toggleClass("selected center");
});
```

jQuery events

- Before version 1.7 event handling was quite confusing and difficult in jQuery.
- You might see these (deprecated) methods in old codes: `bind()`, `unbind()`, `live()`, `die()`, `delegate()`, `undelegate()`, `click()`, `submit()`, etc.
- Nowadays there is only two methods `on()` and `off()`.
- As names say, `on()` sets events to elements and `off()` reverses the process, removing events from elements.

jQuery events (ctd.)

- In its most basic form `on()` accepts two arguments: the event to listen for and a function to fire when the event occurs.
- `off()` accepts the event and removes all listeners for this event.
- Example:

```
$("button").on("click", function() {
  $("h1").toggleClass("selected center");
});
$("h1").on("click", function() {
  $("button").off("click");
});
```

jQuery events (ctd.)

- `on()` has optional selector and data arguments which give context to event binding.
- Selector enables you to bind events to elements that might not exist when the page is created.
- Data is passed to the handler in the `event.data` property each time an event is triggered.
- Data can be any type, but if a string is used the selector must either be provided or explicitly passed as `null`.
- Best practice is to use a plain object so that multiple values can be passed as properties.

jQuery events (ctd.)

- Example:

```
function toggler(event) {
  $("#hello").toggleClass(event.data.style);
}
$("form").on("click", "#center",
             {"style" : "center"}, toggler);
$("form").on("click", "#selected",
             {"style" : "selected"}, toggler);
$("#hello").on("click", function(){
  if (!$("#hello").hasClass("selected"))
    $("#selected").trigger("click");
  $("form").off("click", "#selected", toggler);
});
```

Basic Animations and Effects

Visibility

- `show()`, `hide()`, and `toggle()` methods control the visibility of HTML elements.
- Their values can be `"slow"` (600ms), `"fast"` (200ms), or a number of milliseconds, which is how long the animation should take.
- Example:

```
$("#hello").show("fast");
$("#hello").hide("slow");
$("#toggle").on("click", function(){
  $("#hello").toggle(1000);
});
```

Animate opacity

- `fadeIn()`, `fadeOut()`, and `fadeToggle()` methods operate only on the element opacity.
- They can also take speed arguments, but unlike previous methods, default speed is 400ms.
- Example:

```
$("#hello").fadeIn("fast");
$("#hello").fadeOut("slow");
$("#toggle").on("click", function(){
  $("#hello").fadeToggle(1000);
});
```

Sliding doors

- `slideDown()`, `slideUp()`, and `slideToggle()` methods animate against the height of an element.

- Example:

```
$("#hello").slideDown("fast");
$("#hello").slideUp("slow");
$("#toggle").on("click", function(){
  $("#hello").slideToggle(1000);
});
```

- Note: this effect works well only with block-level elements.

Menu example

```
<h2> Menu 1 </h2>
<nav id="menu1">
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
  </ul>
</nav>
```

```
<h2> Menu 2 </h2>
<nav id="menu2">
  <ul>
    <li>item 1</li>
    <li>item 2</li>
    <li>item 3</li>
  </ul>
</nav>
```

Menu example (ctd.)

```
$(document).ready(function() {
  $("nav").hide();
  $("h2").on("click", function() {
    if (!$(this).hasClass("selected")) {
      $(".selected").removeClass("selected")
                    .next().slideUp("fast");
      $(this).addClass("selected")
             .next().slideDown("slow");
    }
  });
});
```

jQuery UI

Getting jQuery UI

- jQuery UI is a library enhancing the functionality of jQuery.
- jQuery UI provides a set of widgets to make development of rich and interactive websites easier.
- To get jQuery UI, go to `http://jqueryui.com` and download the latest stable version containing `.js` and `.css` files.
- Include both of them in your HTML file.

Getting jQuery UI (ctd.)

- `.css` file provides a theme for the widgets.
- If you want to customize their look, you can use the Themeroller, which is a web-based tool available at `http://jqueryui.com/themeroller/`

Functionality

- To see all features of jQuery UI including source code examples, go to `http://jqueryui.com/demos/`

- Interactions:

Method	Description
`draggable()`	takes no parameters.
`droppable()`	takes a plain JavaScript object where filed `drop` should have a function, defining on-drop behavior.
`resizable()`	takes no parameters.
`selectable()`	takes no parameters.
`sortable()`	takes no parameters, but needs call of `disableSelection()` as well.

Functionality (ctd.)

- Widgets:

Method	Description
`accordion()`	takes no parameters; creates an interface that shows or hides blocks of content based on user interaction.
`autocomplete()`	takes a plain JavaScript object where filed `source` should be specified to feed the hints.
`button()`	takes no parameters.
`datepicker()`	takes no parameters.
`dialog()`	takes no parameters.
`menu()`	takes no parameters.

Functionality (ctd.)

- Widgets: (ctd.)

Method	Description
`progressbar()`	takes a plain JavaScript object where filed `value` should be specified.
`slider()`	takes no parameters, but a function should be assigned on `slidechange` event.
`spinner()`	takes no parameters.
`tabs()`	takes no parameters.
`tooltip()`	takes no parameters.

Laboratory Work

Exercises

- Create a menu similar to the one given on the example.
- Using `window.location.hash` test the hash of the URL and expand only the corresponding menu item.
- Associate each item in the menu with a paragraph of text, which will be shown on the page if the item is clicked.
- Create a test HTML file and try all the jQuery UI elements.
- Modify your previous lab works to use jQuery UI elements (e.g. use tabs, accordion, menu, etc.).

Discussion?!