

JQuery

Javascript That We'll Use

Including jQuery in your webpage

- jQuery source code can be downloaded from the jQuery website, use minimized version to take less time to download
 - Once it is downloaded, save it in the same folder as the rest of the website, then you can use
 - `<script src="jquery-version-number"></script>`
 - Source is also hosted by google at
<https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js> (we'll use this)

jQuery Syntax

- jQuery can select elements on the webpage using the same selectors as CSS, then can perform various operations on those elements selected
 - `$` is used to let javascript know that jQuery is being used in that line, basic line is:
 - `$(selector).action();`
 - `$("p").hide();` //Hides all p elements, makes them invisible on the webpage
 - `$(".test").hide();` //hides elements with class test
 - `$("#test").hide();` //hides THE element with id test
 - At the top of every javascript file, you should include `$(document).ready(function(){});`, then put the entire rest of the file inside the middle brackets, `{}`.
 - This doesn't try to do things to the DOM until the entire webpage is saved, which allows script tags to be included in the head (with `async/defer`) and makes sure everything works as intended

More jQuery syntax

- Adding on to the previous slide, (and to remind you of CSS selectors), all of these are valid selectors
 - `$(“p.intro”)` //p's with class intro
 - `$(“p:first”)` //first p element
 - `$(“ul li”)` // all li's inside ul
 - `$(“ul>li”)` // all li's directly inside ul
 - `$(“ul li:first-child”)` //first li element within the ul element
 - `$(“tr:even”)` //selects all even <tr> elements

jQuery Events

- Events are actions that visitors can perform on elements on the page, when this happens, events are said to be “fired”
 - Common events below:

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

jQuery Event Syntax

- The parameter to an event is a function
 - An example is the `$(document).ready(function(){});` from earlier, passes function to ready
 - `$("#p").click(function(){});` //makes the page do something when any `<p>` is clicked
 - More robust example:
 - ```
$("#p").click(function(){
 $(this).hide();
});
```

<http://codepen.io/cgodfrey/pen/MbbJqx>

Other relevant events are `hover()`, `mousedown()`, `mouseup()`, `hover()`

# Animations! (now it gets fun)

- These actions dynamically change the elements in some way
  - `hide()/show()` will do what they say, only if it is not already true
  - Optional parameter is speed (in milliseconds) or a string “slow”, “medium”, “fast” (default 0 ms)
  - `toggle()` will `hide()` if it is shown and `show()` if it is hidden, also takes speed parameter
- Fading
  - Four different actions, `fadeIn()`, `fadeOut()`, `fadeToggle()`, `fadeTo()`, all take speed parameter too
  - `fadeIn()/fadeOut()/fadeToggle()` - basically identical to the actions above (only minor change)
  - `fadeTo()` will fade to a specific opacity, first parameter speed, second opacity
  - <http://codepen.io/cgodfrey/pen/wooJoR>
- Slide
  - Three functions, `slideUp()`, `slideDown()`, `slideToggle()`. Take speed parameter like others
  - `slideUp()` hides, `slideDown()` displays, `slideToggle()` toggles it
  - <http://codepen.io/cgodfrey/pen/BQQWgX>

# Animation action

- Animate() action
  - Required parameter is an object that represents the CSS property (or properties) being animated, optional parameter is speed
  - `$("#div").animate({"left": "250px"}, 500);` would move all the divs to the RIGHT (increasing space to the left). Note that this is the absolute location, not relative location
    - Note that by default all HTML elements are static (not movable), so you have to define its CSS property first
    - <http://codepen.io/cgodfrey/pen/jVVmqr>
  - You can list a bunch of animations within and jQuery will perform them one after another

```
$("#button").click(function() {
 var div=$("#animate");
 div.animate({"opacity": "0.5"}, 500);
 div.animate({"left": "400px"}, 200);
 Etc.
})
```



# Stopping Animations Early

- Users will want to skip the animation to save time, `stop()` action does that
  - `stop()` has two optional parameters, `stopAll`, and `goToEnd`
    - `stopAll`, if true will not execute queued animations (like previous slide) and just stop everything. Default value is false
    - `goToEnd`, if true will jump to the end of the animation, otherwise just stop where it is in the animation. Default is false
  - <http://codepen.io/cgodfrey/pen/ObbxdX>

# Callback

- All of the actions (including the ones we covered earlier) have another optional parameter we didn't cover earlier
  - The parameter is called callback, it takes a function that is called after the action is completed.

```
$("#button").click(function(){
 $("#p").hide("slow", function(){
 alert("The paragraph is now hidden");
 });
});
```