Web Front-End Development
Week 4: jQuery Effects

Effects

http://api.jquery.com/category/effects/
jQuery includes a lot of cool animation effects.

Basics http://api.jquery.com/category/effects/basics/
.show(): Reveals the matched elements
.hide(): Hides the matched elements
.toggle(): Toggles the visible state of the matched elements using an optional animation

- Duration
  - You can pass in a numerical millisecond value such as 1000, which is one second
  - Predefined speeds
    - “slow” (about 600 milliseconds)
    - “fast” (about 200 milliseconds)
  - Default is 400
- Easing
  - There are two built in easing types in jQuery
  - Swing gives it a little bit of a more natural feel (default)
  - Linear is more of a straight line animation
- For all these functions you can also pass in a completion function, which fires when the effect is finished. This can be a separate function or an anonymous function

https://repl.it/@aileenjp/jQuery-effects

Example:

```
function fadeAnswer() {
  $(this).next('.answer').toggle("slow"); //toggles showing and hiding the answer
  $(this).toggleClass('close'); //toggles adding and removing a class named close to the h2 tag
}

$(document).ready(function () {
  $('.answer').hide(); //selects all divs with class="answer" and hides them
  $('.question').click(fadeAnswer); //toggles fading in and out the answer
}); // end ready
```

Fading http://api.jquery.com/category/effects/fading/
.fadeIn() fades the element to be opaque

- fadeIn() will fade to 100% opacity (opaque, opacity=1)

.fadeOut() hides the element by fading them to be transaprent and removes it from layout

- elements will fade to 0% opacity (transparent, opacity=0)
- when the animation finishes the display CSS property of the element is set to 'none' and removes the element from the page layout

.fadeToggle() toggles the visible state of the matched elements using an optional animation

- Duration and easing parameters same as for .toggle()
- These also can take a completion function

.fadeTo() fades the element to a specified opacity

- In addition to duration, fadeTo() takes a value for opacity between 0 and 1
- Can also take a completion function

Example:
Change toggle to fadeToggle

Sliding http://api.jquery.com/category/effects/sliding/
The sliding functions work the same as the show() and hide() functions but with the sliding effect.
.slideUp(): animates the height of the matched elements to 0 (or whatever CSS min-height is set to)

- when the animation finishes the display CSS property of the element is set to 'none' and removes the element from the page layout

.slideDown(): animates the height of the matched elements to reveal them
.slideToggle(): toggles the state of the matched elements, either sliding them up or down depending on their current state

- Duration is again in milliseconds or the Strings “slow” or “fast”
- Can also take a completion function

Example:
Change toggle to slideToggle

Animation
Custom http://api.jquery.com/category/effects/custom-effects/
If you want to go beyond the basic pre-built jQuery animation functions, there's an animate function that you can use to create custom animation for a wide variety of properties on page elements. Any CSS properties that use numeric values can be animated such as width, height or border thickness.

.animate(properties, duration, easing, completion)provides a way to create smooth animations on the properties of an object.

- The properties parameter is an object with as many property/value pairs as you want.
  - Object notation: {property: value, property: value, …}
  - Property is the name of the property to be animated
  - Value is the value that the animation should move toward
  - Values will be treated as pixels if you don’t give units
  - Or you can specify values using percentage and em
  - You can also specify a += in front of the value that you're animating to for it to animate from whatever the value currently is, and then add that value to it

- The duration, easing, and complete parameters are all optional.
- The duration parameter controls how much time the animation will take. You can specify a numeric value(milliseconds), or the string slow or fast. Defaults to 400 milliseconds.
- The easing parameter can be swing(default) or linear
- The completion parameter is an optional callback function that will be called when the animation sequence completes.

If you want multiple animation effects to happen together, put them all in one animate() function call with all the properties in the properties object.

https://repl.it/@aileenjp/jQuery-simple-animation

Example:

```
function animateImage(image) {
  $("#flatironpic").attr("src", image).css("left", "-750px");
  $("#flatironpic").animate({ left: "50px" }, 2000);
}

$(document).ready(function () {
  $("#summer").click(function () {
    animateImage("images/slide1.jpg")
  });
  $("#winter").click(function () {
    animateImage("images/slide3.jpg")
  });
  $("#fall").click(function () {
    animateImage("images/slide2.jpg")
  });
});
```

- Changes the image src property to the new image and sets left so the image is off the page to the left
- animates the “left” property so the images slide in from the left

The image changes are abrupt, what if we want it to fade out and then the new image fade in as it slides in?

```
function animateImage2(image) {
  $("#flatironpic").attr("src", image).css("left", "-
750px").css("opacity", 0.5);
  $("#flatironpic").animate({ left: "50px", opacity: 1.0 }, 2000);
}
```

```
$(document).ready(function () {
  $("#summer").click(function () {
    $("#flatironpic").animate({ opacity: 0 }, 1000, function () {
animateImage2("images/slide1.jpg") })
  });
  $("#winter").click(function () {
    $("#flatironpic").animate({ opacity: 0 }, 1000, function () {
animateImage2("images/slide3.jpg") })
  });
  $("#fall").click(function () {
    $("#flatironpic").animate({ opacity: 0 }, 1000, function () {
animateImage2("images/slide2.jpg") })
  });
});
```

- animates the current image's opacity to 0 so it fades out
- uses a completion function to then animate the new image
- animates the opacity and the left properties by putting both properties in the properties object { }

If we want different duration we would need to separate out the animation into separate calls and then chain them.

```
  $("#flatironpic").animate({ left: "50px" }, 2000).animate({ opacity: 1.0
}, 1000);
```

Example:
https://repl.it/@aileenjp/jQuery-slideshow
Let's look at how we can create a slideshow of similar sized images that uses a timer instead of the click event.

```
var slides = ["images/slide1.jpg", "images/slide2.jpg",
"images/slide3.jpg"];
var count = 0;

function switchImage() {
  count++;
  $("#flatironpic").attr("src", slides[count]);
}

$(document).ready(function () {
  setInterval(switchImage, 3000); //calls switchImage every 3000
milliseconds
});
```

- setInterval() is a JavaScript function that calls a function based on a value in milliseconds
- why does the count variable have to be declared outside of the function?
- Issues
    - Stops after the 3 images, doesn't loop
    - Image change is abrupt

Let's get the images to loop and have the images fade so the change isn't so abrupt

```
function switchImage2() {
  $("#flatironpic").animate({opacity: 0.0 }, 500, function () {
    if (count < 2) {
      count++;
    } else {
      count = 0;
    }
    $("#flatironpic").attr("src", slides[count]).animate({opacity: 1.0 },
500);
  });
}
```

- Animates the current image to opacity 0 over 500 milliseconds
- Sets count to the next image or back to 0
- Changes image src and animates opacity to 1 over 500 milliseconds.