jQuery 2.0

Asynchronous vs. Synchronous calls

Why does our `playSong` *method need a callback when complete?*

e.g.: `playSong`, `setTimeout`, `setInterval`, animations, AJAX

Review

http://overapi.com/jquery/

Selectors

Attribute Selectors

```
[name]
[name="value"]
[name!="value"]
[name*="value"]
```

What's it select?

```
<div>
  <a href="http://www.google.com">Link A</a>
  <a href="http://unbounce.com">Link B</a>
  <a title="Open tab">Link C</a>
</div>
```

1. `$('a[href]')`

2. `$('[href*="unbounce"')`

3. `$('[title="Open"'])`

Exercises

1. Select all the links with any "href" attribute
2. Select all the links to "http://www.facebook.com"
3. Highlight all the links to any ".com" URL by setting their class to "highlight".
4. Hide all the links NOT to "http://www.google.com"

Pseudo Selectors

```
:first-child
:last-child
:even
:odd
```

What's it select?

```
<body>
  <ul>
    <li>A</li>
    <li>B</li>
    <li><span>C</span></li>
  </ul>
</body>
```

1. `$(':last-child')`
2. `$(':first-child')`
3. `$('li:even')`

Exercises

1. Select the first table row
2. Select the last table row
3. Remove the even-numbered shapes from the green container
4. Highlight the odd rows on the table
5. Hide the last shape in every container

Attributes

Working with the class

- `addClass`
- `removeClass`
- `toggleClass`
- `hasClass`

Exercises

1. Add the "highlight" class to all links
2. Remove the "highlight" class from all links
3. Toggle the "highlight" class on all links. Toggle again.
4. Add the "my-own-class" class to the "body" element. Then verify it is there using "hasClass".

1. When any container is clicked, add the "highlight" class to all the shapes in that container.
2. When your mouse enters a shape, toggle the "highlight" class.
3. **Shrink Ray:** When any shape is clicked do the following:
   - If the shape has the "small" class, hide it.
   - If it has the "medium" class, remove the "medium" class and add the "small" class.
   - If it has the "large" class, remove the "large" class and add the "medium class."

jQuery Chaining

```
// Without chaining
$('a.search').attr('href', 'google.com');
$('a.search').show();

// With chaining
$('a.search').attr('href', 'google.com').show();
```

Exercises

1. Change the colour of all blue shapes to red, using one line of code.
2. Change all the small red circles into large grey squares, using one line of code.
3. Simplify your Shrink Ray using chaining.

```
val
```

Exercises

1. Set the value of the text field in the form to "Hello World"
2. Get the value of the text field in the form
3. When the form's "Submit" button is clicked, change the contents of "Form Message" to be the text field's value.

Manipulation

append + prepend

Exercises

1. Append a "p" tag containing "Appended" to the "body" element.
2. Prepend a "p" tag containing "Prepended" to the "body" tag.
3. Append a new list item to the List, containing a link to link Amazon.com.
4. Prepend a row to the table with the columns "0" and "-"
5. When the form's submit button is clicked, append the text

Traversal

find + next + prev

Exercises

1. Select all the red shapes in the orange container, using "find".
2. Remove all the shapes just before a small red circle.
3. Hide all shapes just after any grey shape.
4. Make all the circles in the green container black, using "find".

Effects

toggle + fadeOut + fadeIn
slideUp + slideDown

Exercises

1. Toggle the green container.
2. Toggle the green container again.
3. Fade the green container out.
4. Fade the green container in.
5. Slide the green container up.
6. Slide the green container down.

Exercises

1. When "Button 1" is clicked, toggle the green container.
2. When "Button 2" is clicked, fade the "Button Message" out
3. When "Button 3" is clicked, fade the "Button Message" back in.
4. When "Button 4" is clicked, slide the green container up.

Events

Shortcut methods

click, mouseenter, mouseleave

```
// using "on"
$('a').on('click', function() {
  console.log('clicked');
});

// shortcut method
$('a').click(function() {
  console.log('clicked');
});
```

`keypress` **event**

keypress event

```
$(document).on('keypress', function() {
  console.log('key pressed!');
});
```

Which key was pressed?

```
$(document).on('keypress', function(event) {
  // event.which returns the character code of the key pressed.
  var charcode = event.which;
  var key = String.fromCharCode(charcode);
  console.log('key ' + key + ' was pressed');
});
```

Demo

Exercises

1. When the 'b' key is pressed, toggle all (b)lue shapes.
2. When the 'r' key is pressed, toggle all (r)ed shapes.
3. When the 'k' key is pressed, toggle all blac(k) shapes.
4. When the 'g' key is pressed, toggle all (g)ray shapes.
5. When the spacebar key is pressed, append a small blue circle to the green container.

keyup **event**

keyup **event**

```
$('input').on('keyup', function() {
  console.log('Input text changed to ', $(this).val());
});
```

Demo

Log what the user is typing in a field

Exercises

1. As you type in the text input, change the "Form Message" to be the same as what you type.
2. As you type in the text input, change the "Form Message" to be the REVERSE of what you type.
3. Make the "Form Message" show the number of characters remaining (14 characters maximum) as you type in the text input. (e.g. "3 characters remaining").

`submit` **event**

submit event

```
$('form').on('submit', function() {
  console.log('form submitted');
});
```

Exercises

1. When the form is submitted, clear the text in the input field.
2. **Shape Destroyer:** The user can enter a color into the input field. When they click "submit":
    - All shapes matching the given color should be removed.
    - If they enter an invalid color show them an alert message telling them this.
    - Clear the input field.

Advanced Events

Event bubbling

Experiment

1. When a black square is clicked log "Black Square Clicked" to the console.
2. When the orange container is clicked log "Orange Container Clicked"
3. Click the black square in the orange container.
4. What happened? Why?

Demo

Use `event.stopPropagation()` to stop bubbling

```
$('.shape').on('click', function(event) {
  console.log('clicked');

  // Don't bubble up this event to the shape's parents.
  event.stopPropagation();
});
```

Experiment

1. Add `event.stopPropagation()` to the black square's click handler.
2. What happens when you click the black square now?
3. What happens when you click the orange container OUTSIDE the black square?

Default Action

Experiment

1. When a link is clicked, log to the console "Link Clicked"
2. What happens when you click a link? Why?

Discussion

- default action (links and forms)
- `event.preventDefault()`

Use `event.preventDefault()` to prevent the default action

```
$('a').on('click', function(event) {
  console.log('clicked');

  // Don't navigate to the href of this link.
  event.preventDefault();
});
```

Exercise

- Add `event.preventDefault()` to your link click handler.
- What happens when you click a link now?

Delegated events

Demo

1. When a shape is clicked, output "shape clicked" in the console.
2. Add a new shape using the console.
3. Try clicking the new shape.

Delegated events

Problem: Event handlers aren't live.

Solution: Use bubbling to delegate to a container that is always there.

```
$('.container').on('click', '.shape', function() {
  // Works for newly created shapes!
});
```

Exercises

1. Set a delegated click handler on the orange container so that red shapes are removed when you click them.

Resources

- http://api.jquery.com/
- http://overapi.com/jquery/
- http://woorkup.com/wp-content/uploads/2011/12/jQuery-17-Visual-Cheat-Sheet1.pdf
- http://learn.jquery.com (core, events, ajax)

Homework