

# WORKING WITH LIBRARIES

# WHAT IS A LIBRARY?

A collection of reusable methods for a particular purpose.

A math library might have functions like:

```
math.sum(array);  
math.pow(num, num);  
math.factorial(num);
```

# WHAT IS A LIBRARY?

Include a script tag to the library on your page.

Call functions from [that library](#).

```
<script src="http://mathjs.org/"></script>

<script>
  var answer = math.sum(2, 2);
  alert(answer);
</script>
```

# INCLUDING JAVASCRIPT

Remember to place scripts at the bottom of the body element.

```
<!DOCTYPE html>
<html>
<head>
  <title>My HTML</title>
  <link rel="stylesheet" href="stylesheets.css">
</head>
<body>

  <script src="scripts-go-here.js"></script>
</body>
</html>
```

Read more: [http://developer.yahoo.com/performance/rules.html#css\\_top](http://developer.yahoo.com/performance/rules.html#css_top)

# ACTIVITY: YOUTUBE

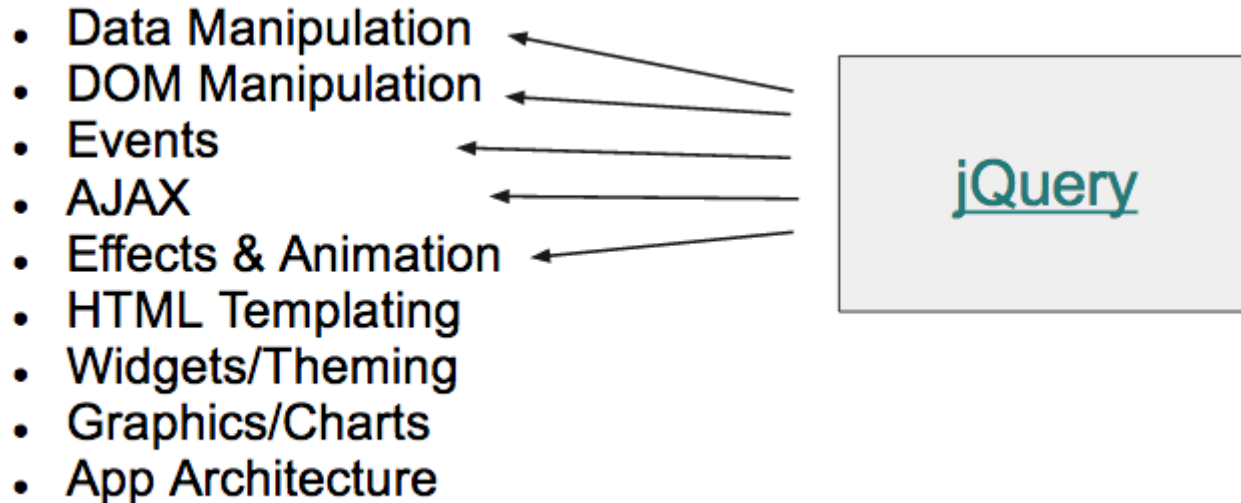
Turn a linked list of titles into a linked list with thumbnails, using JavaScript DOM manipulation and the functions from the youtube.js library.

1. Download the youtube.zip file from Brightspace.
2. Create an array of every link on the page using `document.querySelectorAll(selector)`
3. Iterate through the array. In each iteration of the loop:
  - Find the current href using `element.getAttribute(name)`, and store into a variable
  - Generate a thumbnail URL using `youtube.generateThumbnailUrl(videoUrl)`
  - Create an IMG element using `document.createElement(tagName)`
  - Set the "src" of the IMG element using `element.setAttribute(name, value)`
  - Append the IMG to the link using `element.appendChild(element)`

# WHY USE LIBRARIES?

Don't reinvent the wheel.

Abstract on top of cross-browser differences.



6 Reasons to use Libraries & Frameworks

# JQUERY: WHY?

No library:

```
var p = document.createElement('p');
p.appendChild(document.createTextNode('Welcome!'));
p.style.cssFloat = 'left';
p.style.backgroundColor = 'red';
p.className = 'special';
document.querySelector('div.header').appendChild(p);
```

jQuery:

```
var newP = $('<p>Welcome!</p>');
newP.css({'float' : 'left', 'background-color': 'red'});
newP.addClass('special');
$('div.header').append(newP);
```

# JQUERY: WHY NOT?

Do you really **really** need it?

You Might Not Need jQuery



# JQUERY: INCLUDING

Download or link from an official CDN.

Then include using a `<script>` tag on the page.

```
<script src="js/jquery.min.js"></script>
<script>
    // Your custom code here
</script>
```

Or be super prepared:

```
<script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
<script>
    window.jQuery || document.write('<script src="js/jquery.min.js"></script>')>
</script>
```

# JQUERY: THE BASICS

In HTML:

```
<p>Welcome to jQuery!</p>
```



In JS:

```
$('p').addClass('special');
```



Back to HTML

```
<p class="special">Welcome to jQuery!</p>
```

# JQUERY: THE BASICS

```
$('p').addClass('special');
```

The global jQuery function. Can also be "jQuery".

```
$
```

Finds DOM element(s) according to what's in the quotes. Returns a "jQuery collection."

```
('p')
```

Built-in jQuery method that adds the specified class to the collection.  
[Read the docs here.](#)

```
addClass('special')
```

# JQUERY: FINDING ELEMENTS

All CSS selectors are valid, plus more.

[Read the docs.](#)

With this HTML...

```
<p>Welcome!</p>
```

→

```
$( 'p' );
```

```
<div id="main">Welcome!</div>
```

→

```
$( '#main' );
```

```
<p class="intro">Welcome!</p>
```

→

```
$( '.intro' );
```

```
<div id="main">  
<p class="intro">Welcome!</p>  
</div>
```

→

```
$( '#main' ).find( '.intro' );  
// or  
$( '#main .intro' );
```

We find it this way:

# JQUERY: CHANGING ELEMENTS

If we start with this HTML:

```
<a href="http://www.google.com">Google</a>
```

We can use this jQuery:

```
$( 'a' ).html( 'Yahoo!' )  
$( 'a' ).attr( 'href', 'http://www.yahoo.com' )  
$( 'a' ).css( { 'color': 'purple' } )
```

And we'll get this:

```
<a href="http://www.yahoo.com" style="color:purple">Yahoo!</a>
```

# JQUERY: READING ELEMENTS

If we start with this HTML...

```
<a href="http://www.yahoo.com" style="color: purple;">Yahoo!</a>
```

...We can find out lots of things about it:

```
console.log($('a').html()); // Yahoo!
```

```
console.log($('a').attr('href')); // http://yahoo.com
```

```
console.log($('a').css('color')); // purple
```

# JQUERY: LOOPING

Yahoo  
Google



```
// Get the links and assign them to variable links
var links = $('a');

// Use the jQuery each method to loop through each link.
links.each(function() {

    // Take each link and assign it to the variable link
    var link = $(this);

    // Set the link text to the link text + '!'
    link.text(link.text() + '!');
});
```



Yahoo!  
Google!

# THE DOM-READY EVENT

Use it to when you need to wait for the DOM to be ready

```
$(document).ready(function() {  
    // Do everything  
});
```

Is the same as:

```
$(function() {  
    // Do everything  
});
```

Read more: <http://learn.jquery.com/using-jquery-core/document-ready/>



# ACTIVITY: YOUTUBE, PART 2

Append video thumbnails like before but with jQuery instead of native JavaScript.

1. Download the [jQuery library](#) and save it into the same directory as your project files.
2. Add a `script` tag to the page which loads in the jQuery library you just downloaded.
3. Use jQuery to append video thumbnails to your website. If you're not sure how to do something with jQuery, google for "X with jQuery" and you will likely find the jQuery docs or StackOverflow questions.

# EVENTS

## Non-jQuery Way:

```
element.addEventListener('click', function(event) {  
    // code to be executed when user clicks  
});
```

## jQuery Way:

Bind a named function:

```
function clickHandler() {  
    console.log('clicked!');  
}  
  
$('#button').on('click', clickHandler);
```

Or bind an anonymous function:

```
$('#button').on('click', function() {  
    console.log('clicked!');  
});
```

# THE EVENT OBJECT

The 'event' parameter provides a return value that is treated in a special way, telling us a variety of things about our event, such as the type or element clicked.

```
$( 'a' ).on( 'keypress', function(event) {  
  
    // outputs event object  
    console.log(event);  
  
    // outputs value for key pressed  
    console.log(event.which);  
  
    // outputs element clicked  
    console.log(event.target);  
});
```

[jQuery Event Object documentation](#)

# EVENTS

## Preventing default actions

```
$('form').on('submit', function(event) {  
  event.preventDefault();  
  
  console.log('Not submitting the form!');  
  // Maybe I want to do stuff first.  
});
```

[event.preventDefault\(\) vs. return false; on CSS-Tricks](#)

# EFFECTS & ANIMATION

```
$( '#error' ).toggle(1000);  
$( '#error' ).hide();  
  
$( '#error' ).fadeIn(1000);  
  
$( '#error' ).show(1000, function(){  
    $(this).addClass( 'alert' )  
});
```

[jQuery Effects Documentation](#)

# CHAINING

```
var banner = $('#banner');  
banner.css('color', 'red');  
banner.html('Welcome!');  
banner.show();
```

Is the same as:

```
banner.css('color', 'red').html('Welcome!').show();
```

Is the same as:

```
banner.css('color', 'red')  
  .html('Welcome!')  
  .show();
```

# ACTIVITY: EVENTS AND CHAINING

Let's bind some events and play with chaining.

Start with [this basic html page](#).

Try one of the following or create your own activity.

- Create a counter when you click the button.
- Change the background when you click the button.
- Prevent default actions on the submit button.
- Create an error message and fade it in if the form isn't filled out.