CLIENT SIDE WEB DEVELOPMENT TEMA6: JQUERY INTRODUCTION

CFGS - DAW 2

JESUITAS

Sagrado Corazón
Jesuitas · Logroño

Index

- What is a Library?
- Why JQuery?
- Setting Up JQuery
- Fundamentals
- Reading elements
- Changing elements
- Create, Manipulate & Inject

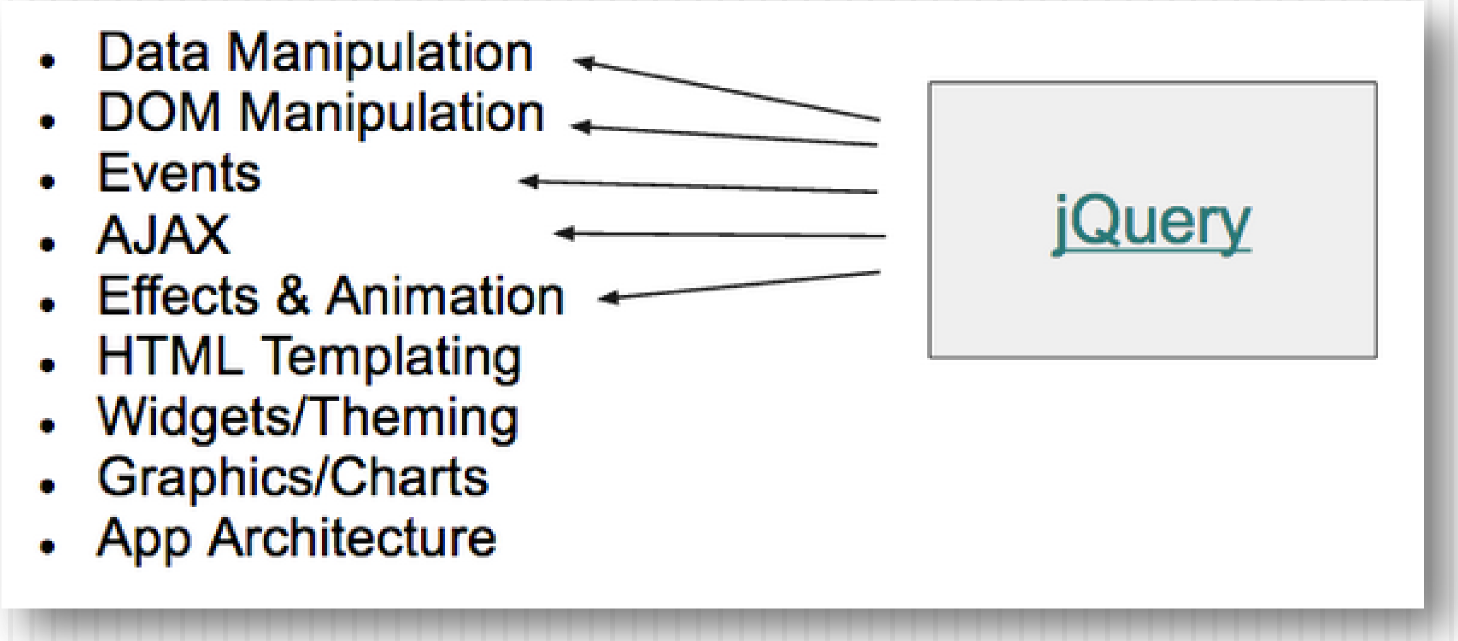
1. What is a Library?

- A collection of reusable methods for a particular purpose.

2. Why JQuery?

- There were a number of reasons web front-end developers started to use JQuery.
- The most important reasons included the following:
  - Minimal and easy coding
  - Readable/clean code
  - Easy CSS handling
  - Animation methods
  - Intuitive function calls
  - Cross-browser
  - Open source, big community

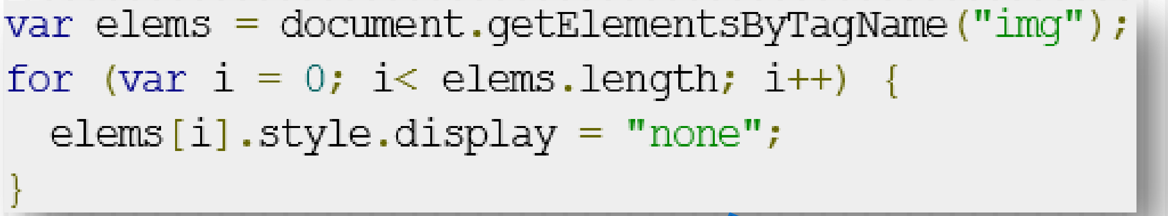
2. Why JQuery?



- Data Manipulation
- DOM Manipulation
- Events
- AJAX
- Effects & Animation
- HTML Templating
- Widgets/Theming
- Graphics/Charts
- App Architecture

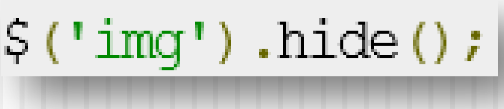
2. Why JQuery?

- Example with JS (no library):



```
var elems = document.getElementsByTagName("img");
for (var i = 0; i< elems.length; i++) {
  elems[i].style.display = "none";
}
```

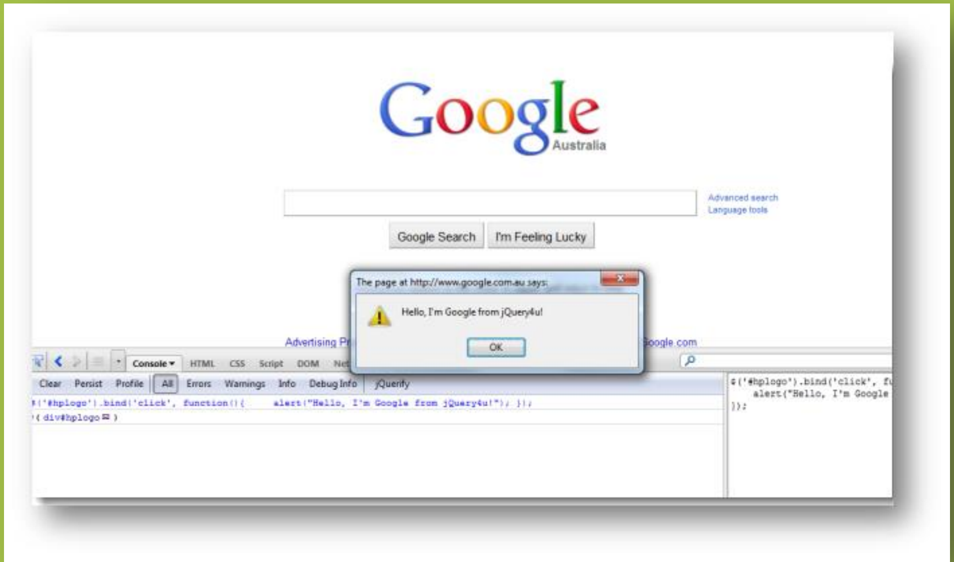
- The same with JQuery:



```
$('img').hide();
```

2. Why JQuery?

A.6.2. Check out this link and configure firebug in order to live debug with JQuery. Try the example given in the guide.



The page at http://www.google.com.au says:

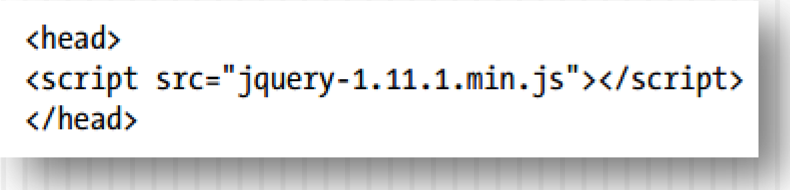
Hello, I'm Google from jQuery4u!

```
$('#hplogo').bind('click', function(){     alert("Hello, I'm Google from jQuery4u!"); });
( div#hplogo )
```

```
$('#hplogo').bind('click', fu
    alert("Hello, I'm Google
});
```

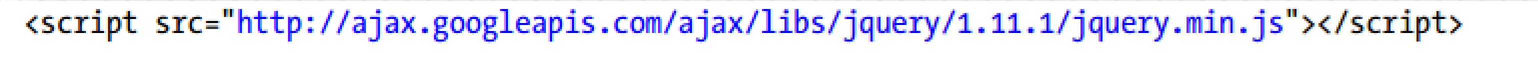
3. Setting Up JQuery

- There are different ways to start using JQuery on your web page
  1. Download the JQuery Library from JQuery.com
     - The JQuery library is a single JavaScript file, and now you have to reference it with a <script> tag in your code. For example:



```
<head>
<script src="jquery-1.11.1.min.js"></script>
</head>
```

  2. Include JQuery from some content delivery network (CDN), such as Google



```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.1/jquery.min.js"></script>
```

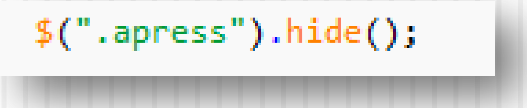
4. Fundamentals

JQuery Syntax

- In the most basic of the scenarios, you would perform an action in JQuery:

$(selector).action();

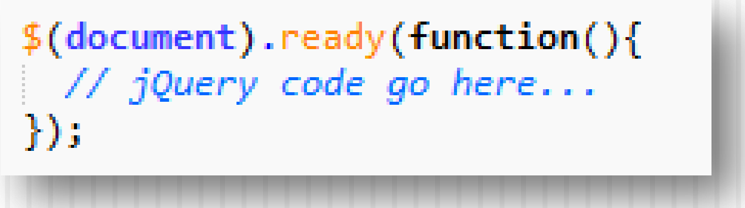
1. The $ (dollar) sign is used to define and access JQuery. It would be the same as *JQuery(selector).action();*
2. *(selector)* is used to query (or find) an HTML element.
3. *action()* is the JQuery action that must be performed on an element. Already implemented in the library for you. Check the API documentation.



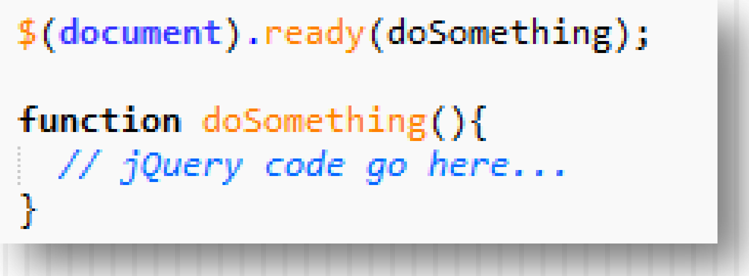
```
$(".apress").hide();
```

4. Fundamentals

- **Document Ready Event**
  - The document ready event is important, considering that it provides a time line for certain actions to be executed once the document is downloaded completely to the client web browser.

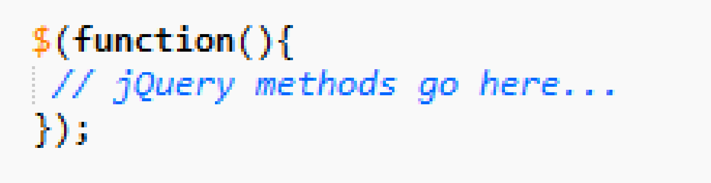


```
$(document).ready(function(){
    // jQuery code go here...
});
```



```
$(document).ready(doSomething);

function doSomething(){
    // jQuery code go here...
}
```

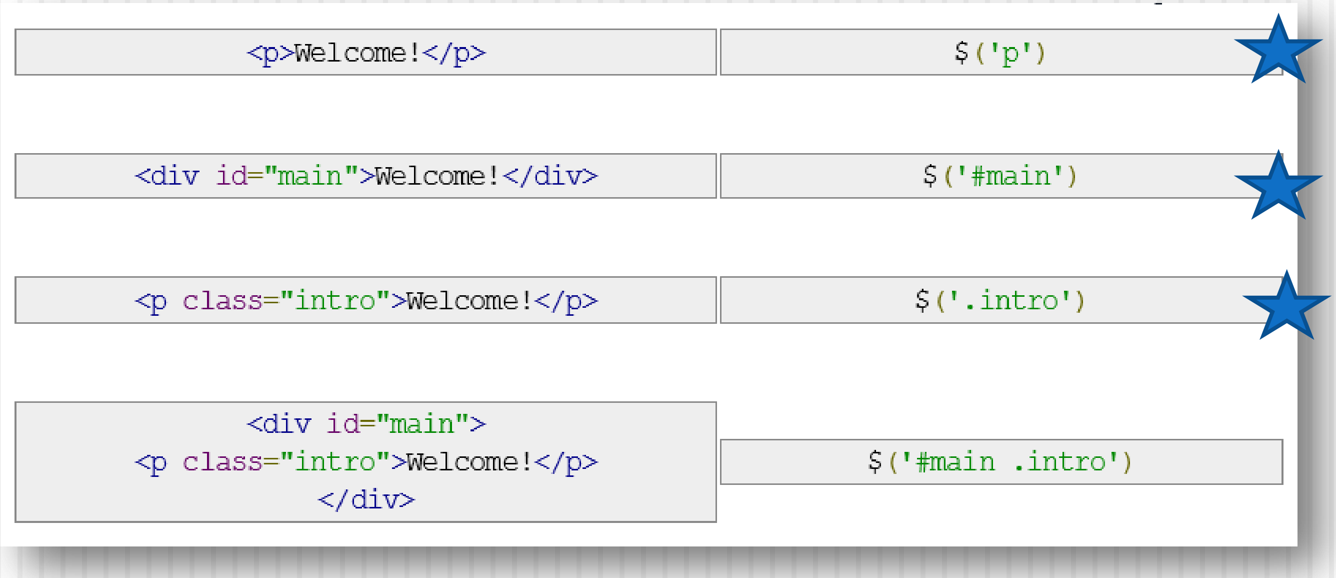


```
$(function(){
    // jQuery methods go here...
});
```

4. Fundamentals

Finding elements: Selectors

- A JQuery selector is a functionality that makes use of expressions to find a matching element from the DOM, based on certain criteria.
- All CSS selectors are valid, and more.
- Some examples:

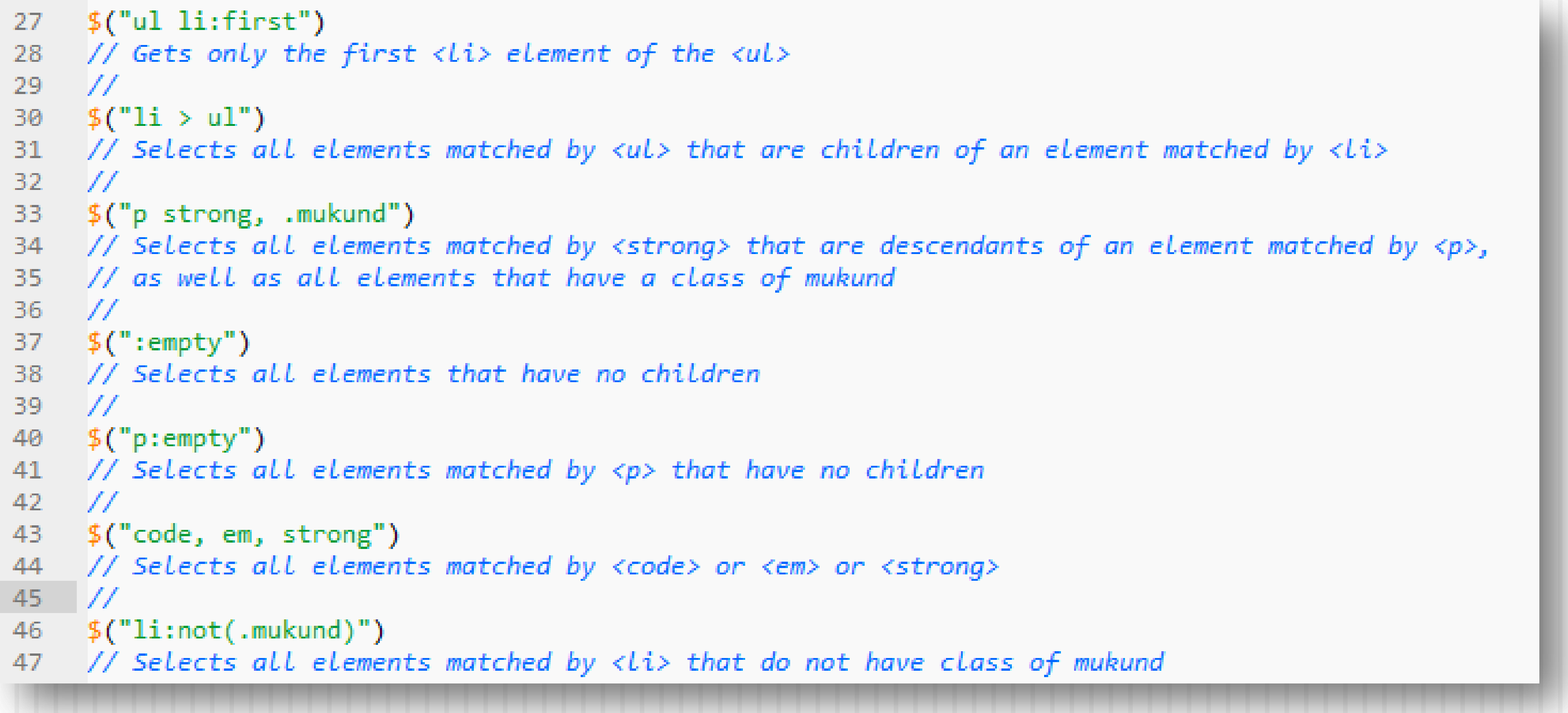


`<p>Welcome!</p>`	`$('p')`
`<div id="main">Welcome!</div>`	`$('#main')`
`<p class="intro">Welcome!</p>`	`$('.intro')`

`<div id="main">`
`<p class="intro">Welcome!</p>`
`</div>`
	`$('#main .intro')`

4. Fundamentals

- **Finding elements: Selectors**
  - Other examples:



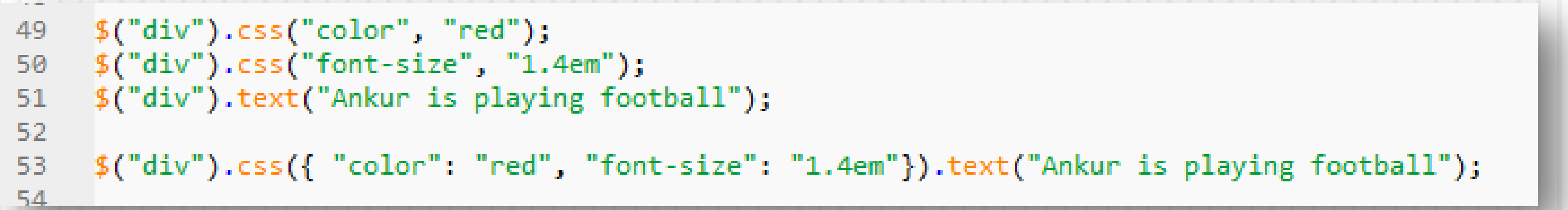
```
$("ul li:first")
// Gets only the first <li> element of the <ul>
//
$("li > ul")
// Selects all elements matched by <ul> that are children of an element matched by <li>
//
$("p strong, .mukund")
// Selects all elements matched by <strong> that are descendants of an element matched by <p>,
// as well as all elements that have a class of mukund
//
$(":empty")
// Selects all elements that have no children
//
$("p:empty")
// Selects all elements matched by <p> that have no children
//
$("code, em, strong")
// Selects all elements matched by <code> or <em> or <strong>
//
$("li:not(.mukund)")
// Selects all elements matched by <li> that do not have class of mukund
```

4. Fundamentals

- <u>Some comments about selectors:</u>
  1. To make a selector query fast, you must be specific about the right-hand side of your selector.
     - Example:
       - If you write ***$('.data tr td.mukund')***, it would produce the same effect as ***$(".data td.mukund")***, the reason being straightforward and simple: all TDs come under TRs.
       - ***$(".data td.mukund")*** would have better performance.

4. Fundamentals

2. Don’t repeat selectors: computational cost → For every selector that you specify, the JQuery engine has to perform a lookup in the entire DOM tree



```
$("div").css("color", "red");
$("div").css("font-size", "1.4em");
$("div").text("Ankur is playing football");

$("div").css({ "color": "red", "font-size": "1.4em"}).text("Ankur is playing football");
```

3. Try to avoid using this type of selector as much as possible.

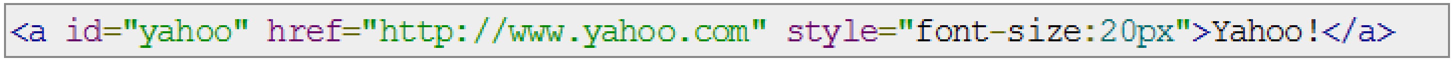
4. Fundamentals

E.2.1 Task - Selectors

Khan course: “Welcome to JQuery” + “DOM access with JQuery”

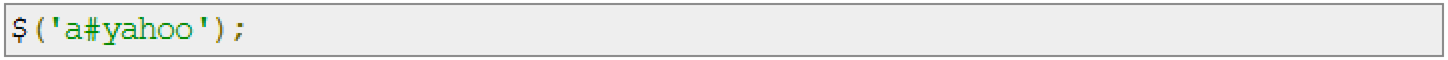
5. Reading elements

- If we have this HTML:



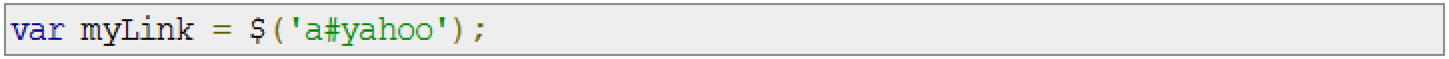
```
<a id="yahoo" href="http://www.yahoo.com" style="font-size:20px">Yahoo!</a>
```

- We can find it:



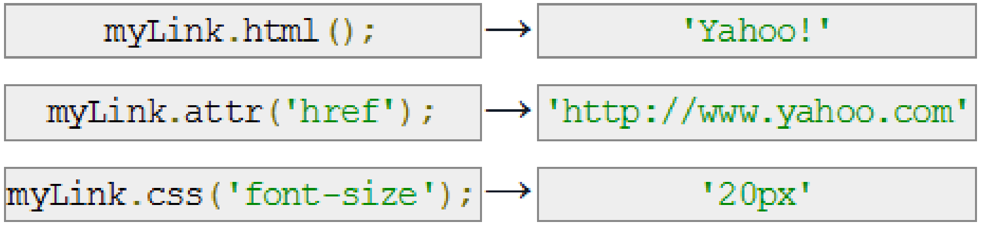
```
$('a#yahoo');
```

- Store it:



```
var myLink = $('a#yahoo');
```

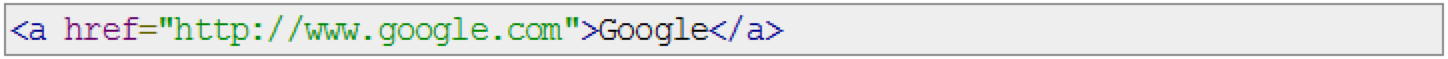
- Read information:



```
myLink.html();               → 'Yahoo!'
myLink.attr('href');         → 'http://www.yahoo.com'
myLink.css('font-size');     → '20px'
```

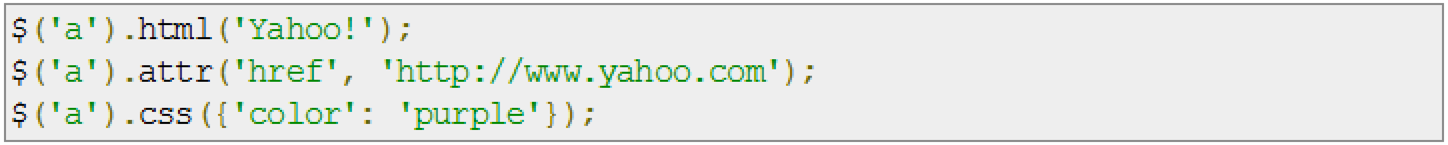
6. Changing elements

- If we start with this HTML:



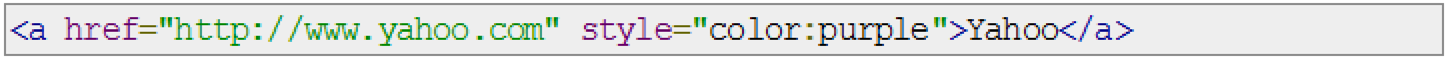
```
<a href="http://www.google.com">Google</a>
```

- We can apply this JQuery:



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

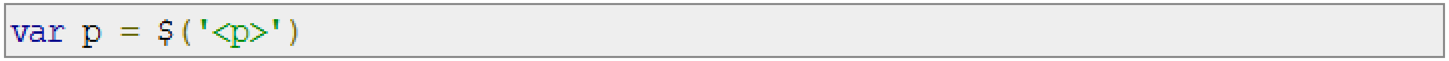
- And the result would be:



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

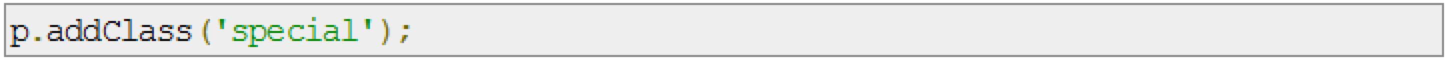
7. Create, Manipulate & Inject

1. Create element and store a reference



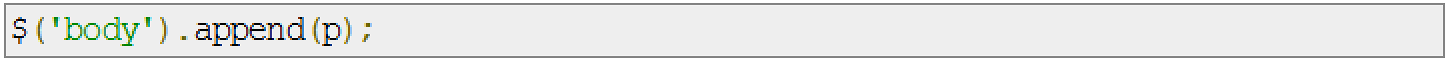
```
var p = $('<p>')
```

2. Use a method to manipulate



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

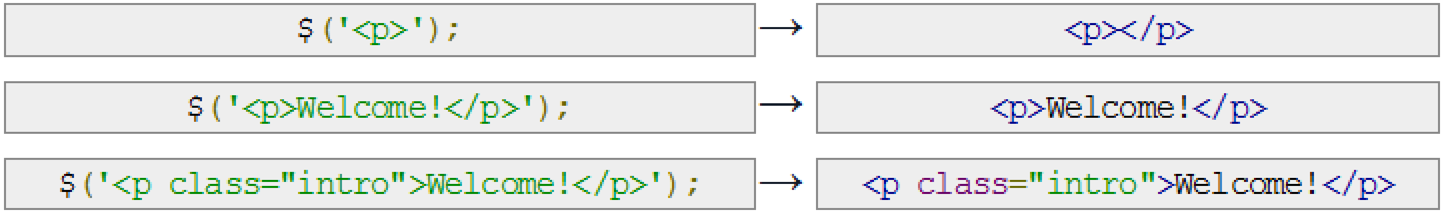
3. Inject into your HTML



```
$('body').append(p);
```

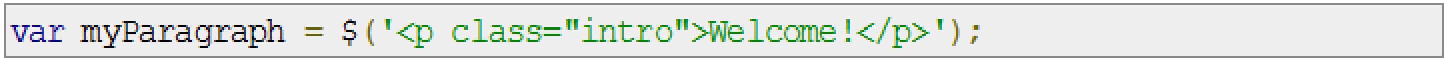
7. Create, Manipulate & Inject

- **Create and store:**
  - Pass in any HTML string and JQuery will create it and return it as a collection:



```
$('<p>');                                  → <p></p>
$('<p>Welcome!</p>');                      → <p>Welcome!</p>
$('<p class="intro">Welcome!</p>');        → <p class="intro">Welcome!</p>
```

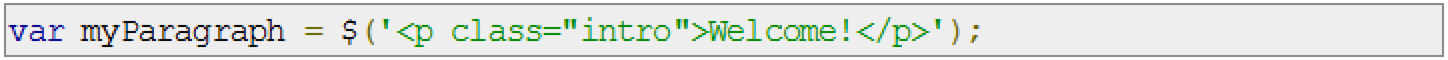
  - We can store a reference to our new element in memory:



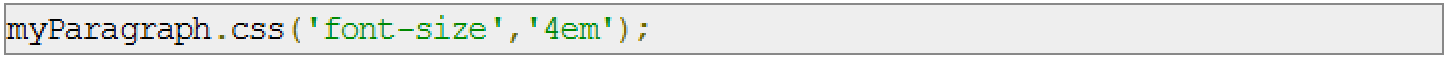
```
var myParagraph = $('<p class="intro">Welcome!</p>');
```

7. Create, Manipulate & Inject

- **Manipulate:**
  - Now that we've stored a reference, we can manipulate it:

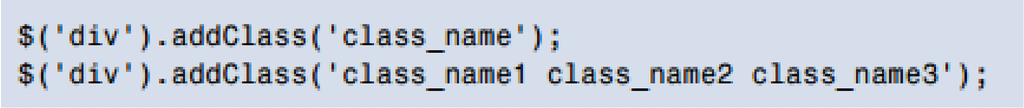


```
var myParagraph = $('<p class="intro">Welcome!</p>');
```



```
myParagraph.css('font-size','4em');
```

  - Classes can also be easily added or removed:

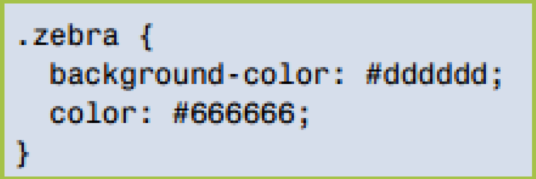


```
$('div').addClass('class_name');
$('div').addClass('class_name1 class_name2 class_name3');
```

7. Create, Manipulate & Inject

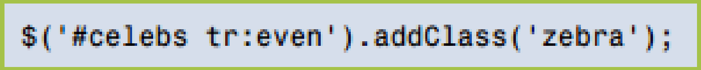
A6.1. Follow these steps:

1) Create an HTML with a table in the body.
2) Define a “zebra” class style like this:



```
.zebra {
  background-color: #dddddd;
  color: #666666;
}
```

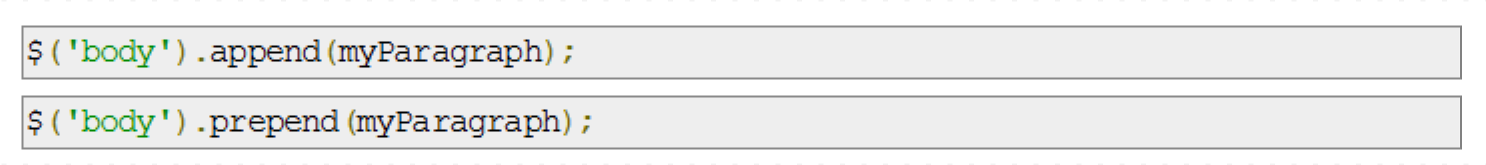
3) Using JQuery add zebra class to even:



```
$('#celebs tr:even').addClass('zebra');
```

7. Create, Manipulate & Inject

- **Inject:**
  - We can take our stored reference to myParagraph and inject it somewhere:



```
$('body').append(myParagraph);
```

```
$('body').prepend(myParagraph);
```

7. Create, Manipulate & Inject

A6.2. Change the given example that use DOM API, in order it to use JQuery.