MANIPULATING CONTENT

Unit 5

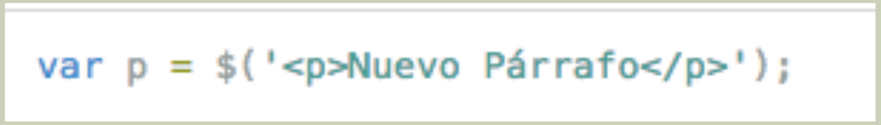
INDEX

1- ITERATING THROUGH ELEMENTS

Filter	Description	Example
size() // length	Number of the elements	$(p).length;
get()	Get a list with the DOM elements	$(p).get();
get(n)	Get the specified DOM element.	$(p).get(3);
find({expression})	Get the elements with the expression specified.	$(body).find(p.classA).css(' border','3px solid black');
each(callback(i))	Run a function in the context of each element. Using of this.	$('li').each(function(i) { var text = this.innerText; console.log('El texto del elemento ' + i + ' es: ' + text); });

2- MANIPULATING CONTENT

- CREATING CONTENT
  - We can create HTML content by passing as an argument the code to the function $()



```
var p = $('<p>Nuevo Párrafo</p>');
```

  - We can also use html() and text() methods to obtain and assign content.

Filter	Description	Example
html()	Get the html of the first selected element	$(”#myimg”).html();
html(htmlString)	Assign the string to all the html found.	$(.blue).html(“My list”);

2- MANIPULATING CONTENT

- MANIPULATING ATTRIBUTES
  - We can manipulate the attributes of one or more attributes using the following functions.

Filter	Description	Example
attr(name)	Get the value of the attribute given	$(#myimg).attr(src);
attr(name,value)	Assign the value to the attribute given.	$(#myimg).attr(src,"http:…");
attr({name:value})	Assign several values to the attributes given. JSON syntax.	$(#myimg).attr({ src: 'http:….', alt: 'my image' });
removeAttr(name)	Removes the given attribute from the element	$(#myimg).removeAttr(alt);

2- MANIPULATING CONTENT

- INSERTING AND MOVING CONTENT
  - We can add and move content to the elements selected using the following functions.
  - Note that we can add already existing content or new one.

Filter	Description	Example
appendTo(selector)	Append at the end of the selection, the content specified.	$(“<li>Second element</li>”).appendTo(ul li:first);
prependTo(selector);	Works as the append but inserting at the beginning.	$(div p).preppend(“<p>hello</p>”);
insertBefore(selector);	Inserts before the specified element.	$(#myimg).insertBefore(ul:first);
insertAfter(selector);	Inserts afterthe specified element.	$(#myimg).insertAfter(ul:eq(2));

2- MANIPULATING CONTENT

- WRAPPING CONTENT
  - Wrapping content means introduce a element inside a element.

Filter	Description	Example
wrap(html)	Wrap each element inside the specified html.	$(.a).wrap("<div style='border:3px solid red'></div>");
wrapAll(html)	Wrap all the selected elements inside the specified html.	$(.a).wrapAll("<div style='border:3px solid red'></div>");
wrapInner(html);	Wrap the selected elements with the html but inheriting the content of the parent.	$(.a).wrapInner("<div style='border:3px solid red'></div>");

2- MANIPULATING CONTENT

- WRAPPING CONTENT – wrap and wrapAll

```
<div class="foo"></div>
<div class="foo"></div>
<div class="foo"></div>
```

```
$('.foo').wrap('<div class="bar" />');
```

```
<div class="bar"><div class="foo"></div></div>
<div class="bar"><div class="foo"></div></div>
<div class="bar"><div class="foo"></div></div>
```

```
$('.foo').wrapAll('<div class="bar" />');
```

```
<div class="bar">
  <div class="foo"></div>
  <div class="foo"></div>
  <div class="foo"></div>
</div>
```

2- MANIPULATING CONTENT

- **WRAPPING CONTENT – wrapInner**

```html
<div class="container">
  <div class="inner">Hello</div>
  <div class="inner">Goodbye</div>
</div>
```

```javascript
$( ".inner" ).wrapInner( "<div class='new'></div>");
```

```html
<div class="container">
  <div class="inner">
    <div class="new">Hello</div>
  </div>
  <div class="inner">
    <div class="new">Goodbye</div>
  </div>
</div>
```

2- MANIPULATING CONTENT

- REPLACING CONTENT
  - With jQuery, we can replace the content of a element by another

Filter	Description	Example
replaceWith(content)	Replace the selected element with the specified content.	$(p:first).replaceWith("<img>…</img>");
replaceAll(selector)	Replace the selected emenets with the selected elements given	$(#foo).replaceAll(div.a);

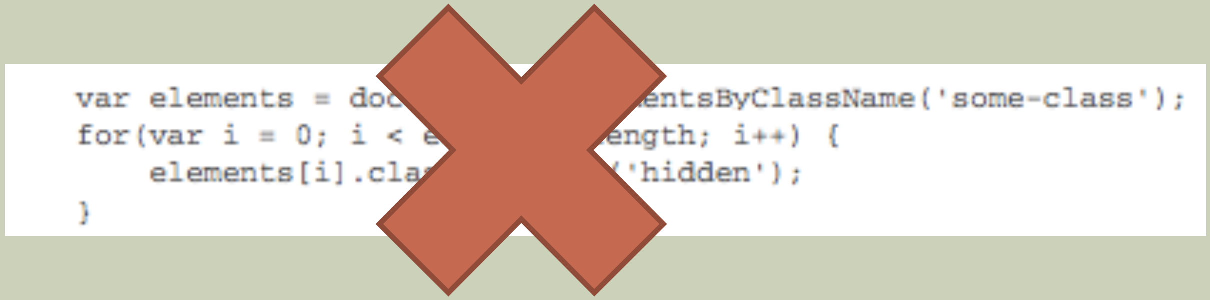
2- MANIPULATING CONTENT

- **REMOVING AND CLONING ELEMENTS**

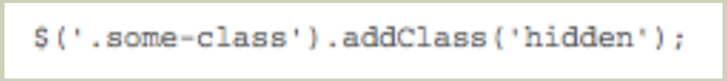
Filter	Description	Example
empty()	Empty the content of a element	$(.foo).empty();
remove()	Remove the selected elements and their content	$(#foo).remove();
clone()	Copy elements and return it to append to other elements	$(#foo).first().clone.appendto(#bar);

2- MANIPULATING CONTENT

- ADDING AND REMOVING CLASS NAMES
  - Adding and removing class names to the elements of a set is an easy operation in jQuery. In JS, we had several problems with this.



```
var elements = doc[illegible]entsByClassName('some-class');
for(var i = 0; i < e[illegible]ength; i++) {
    elements[i].cla[illegible]('hidden');
}
```

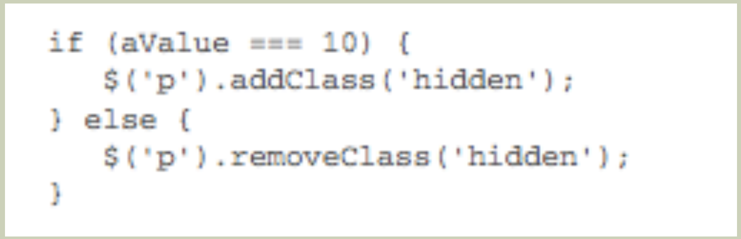


```
$('.some-class').addClass('hidden');
```

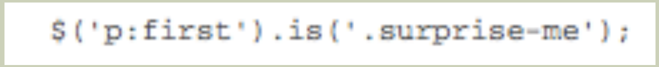
2- MANIPULATING CONTENT

ADDING AND REMOVING CLASS NAMES

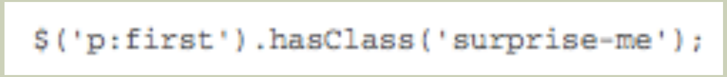
Filter	Description	Example
addClass(className)	Add the given class to the classlist of the element.	$(#foo).addClass('bar');
removeClass(classNa me)	Remove the given class from the classlist of the element.	$(#foo).removeClass('bar');
hasClass()	Determines if any element of the set possesses the passed class name.	$(#foo).hasClass('bar');



```
if (aValue === 10) {
    $('p').addClass('hidden');
} else {
    $('p').removeClass('hidden');
}
```



```
$('p:first').is('.surprise-me');
```

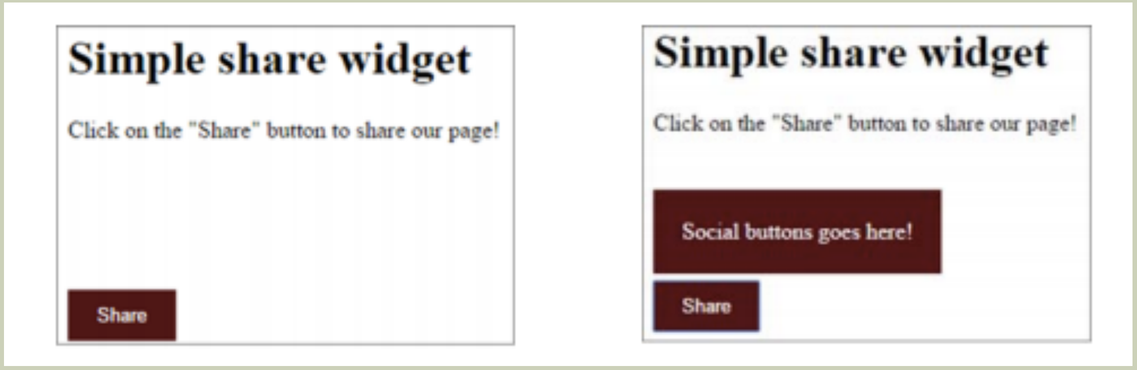


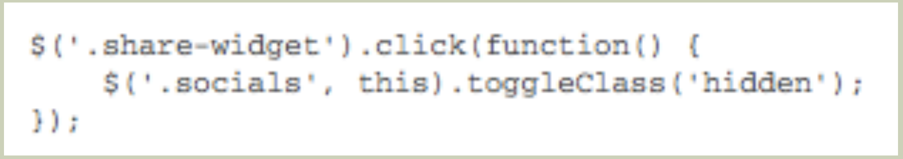
```
$('p:first').hasClass('surprise-me');
```

2- MANIPULATING CONTENT

TOOGLING CLASSES

Filter	Description	Example
toogleClass(classNam e)	Add the class if is not in the element or remove it if is in the element. i.e. Social button	$(#foo).toogleClass('bar');


Simple share widget
Click on the "Share" button to share our page!
Share
Simple share widget
Click on the "Share" button to share our page!
Social buttons goes here!
Share

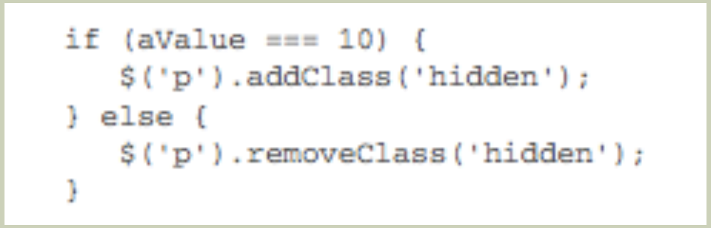


```
$('.share-widget').click(function() {
    $('.socials', this).toggleClass('hidden');
});
```

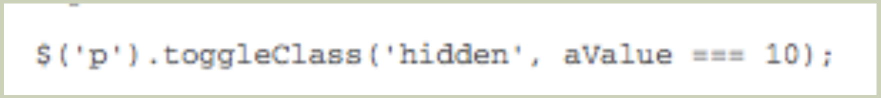
2- MANIPULATING CONTENT

TOOGLING CLASSES

Filter	Description	Example
css(name)	Toogles the class based on the given condition	$(#foo).toogleClass('bar', a===3);



```
if (aValue === 10) {
   $('p').addClass('hidden');
} else {
   $('p').removeClass('hidden');
}
```



```
$('p').toggleClass('hidden', aValue === 10);
```

2- MANIPULATING CONTENT

- GETTING AND SETTING STYLES

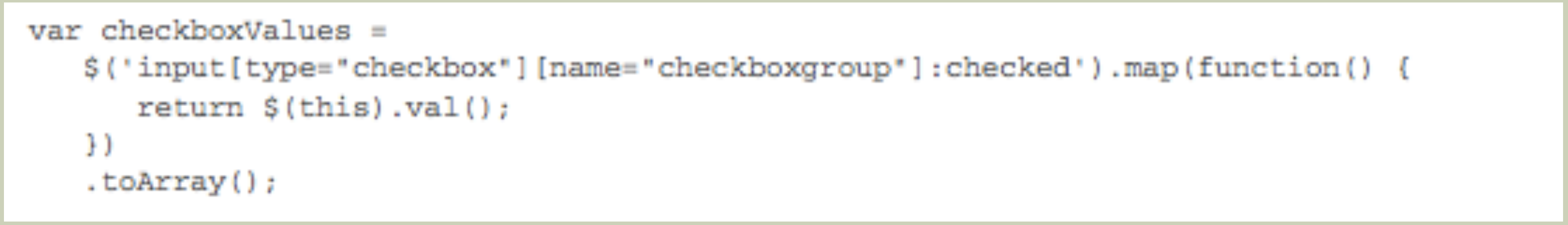
Filter	Description	Example
css(name)	Returns the css property with the given name of the element selected.	$(#foo).css(border);
css(properties)	Assign a set of properties to a element	$(#foo).css({ 'border': '3px solid green', 'background-color': 'red' });
css(property,value)	Assign the specified value to the specified property	$(#foo).css('width','20');

- GETTING AND SETTING DIMENSIONS (activity)
  - width(), height(),innerHeight(), innerWidth(),offset(),position()…

2- MANIPULATING CONTENT

- DEALING WITH FORM ELEMENT VALUES
  - Because form elements have special properties, jQuery contains some functions to getting and setting its values.

Filter	Description	Example
val()	Returns the current value of the first element in the collection.	$('input[type="radio"][name="radio-group"]:checked').val();
val(value)	Sets the passed value as the value of all matched elements	$('input[type=”select"]').val(['one', 'two', 'three']);



```
var checkboxValues =
    $('input[type="checkbox"][name="checkboxgroup"]:checked').map(function() {
        return $(this).val();
    })
    .toArray();
```