jQuery

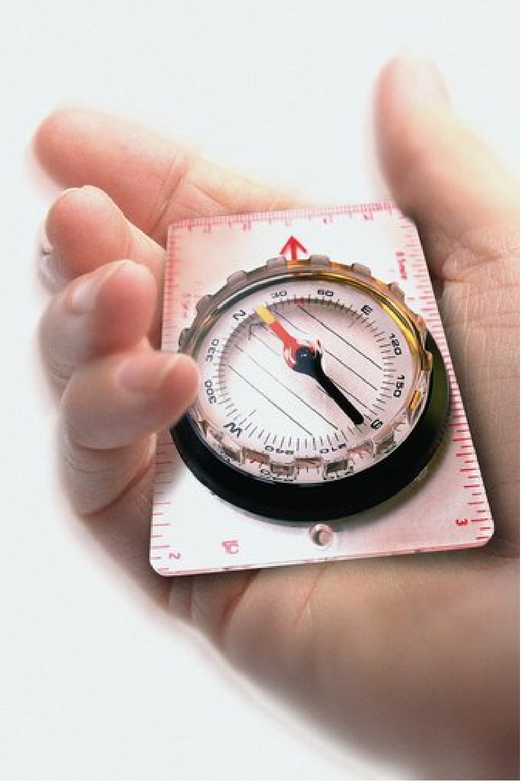
The JavaScript Library

Course objectives

By following this course, you will be able to:

- Enumerate the main advantages of using jQuery
- Manipulate CSS properties
- Design simple JavaScript animations
- Do event programming
- Manipulate content

Course plan



- Discovery of jQuery
- Selectors
- CSS Manipulation
- Event
- Content Manipulation

jQuery

DISCOVERY

Discovery

Presentation

- Library designed to simplify the use of JavaScript
- First release in 2006 by John Resig
  - Actual version: 2.2.x
  - Beta 3.0 released in 2016

- The most popular JavaScript library!
  - Used by 95.9% using a JavaScript library!
  - Used by over 68.6% of all websites

Discovery

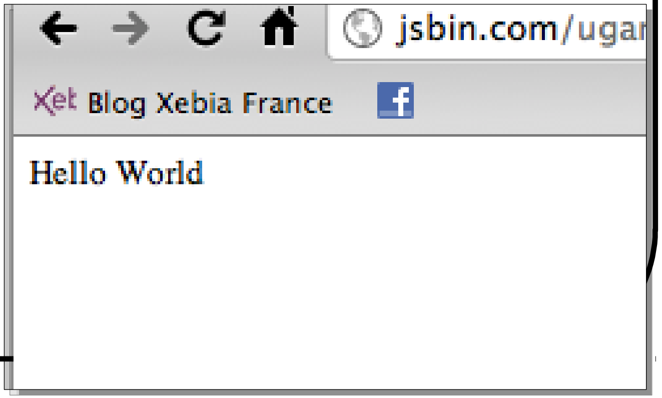
Presentation

- Free and Open Source
- Extendable easily!
- Has many Plugins
  - Image sliders
  - Form validators
  - File upload tools
  - Etc...

Discovery

Hello world

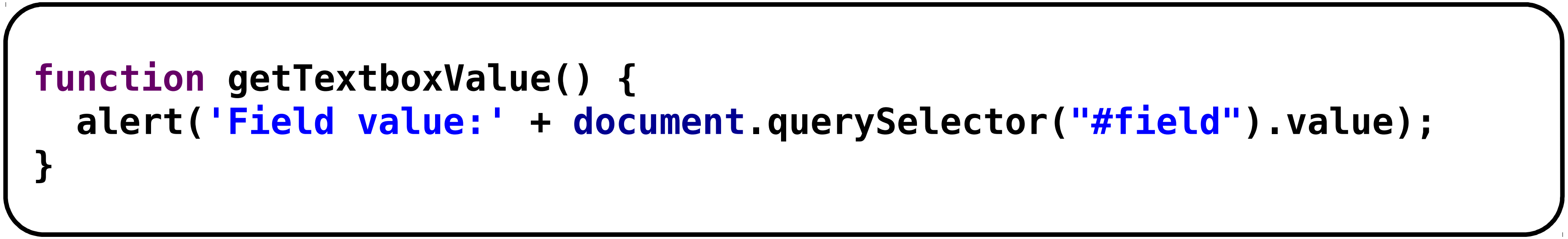
```
<html>
   <head>
      <title>Hello World with jQuery</title>
      <meta charset="utf-8" />
      <script type="text/javascript" src="jquery.js"></script>
   </head>
   <body>
      Hi !
      <script type="text/javascript">
         $('body').html('Hello World');
      </script>
   </body>
</html>
```



Hello World

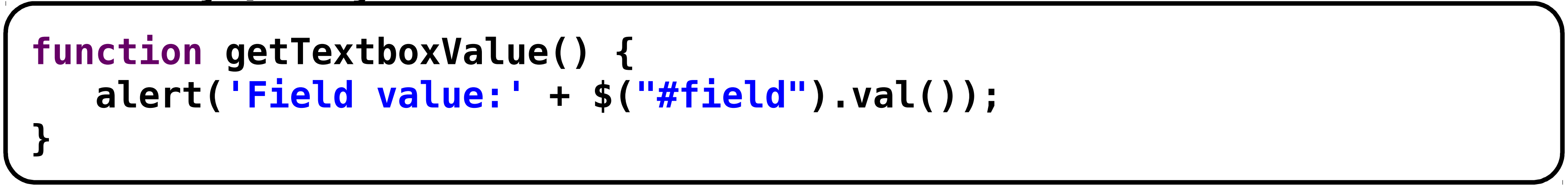
Advantages

- Easier syntax to navigate into a document
  - Simple JavaScript version :



```
function getTextboxValue() {
  alert('Field value:' + document.querySelector("#field").value);
}
```

  - jQuery version :



```
function getTextboxValue() {
   alert('Field value:' + $("#field").val());
}
```

Discovery

Advantages

- JavaScript can render differently on some browsers:
  - Weird behavior
  - Undefined keywords
  - Different values

- Guarantees code compatibility on many browsers

I DON'T ALWAYS DEVELOP WITH JAVASCRIPT
BUT WHEN I DO I USE jQUERY

Simple JavaScript version

```javascript
function fadeIn(el, time = 1000) {
  var op = 1;  // initial opacity
  var timer = setInterval(function () {
    if (op <= 0.1) {
      clearInterval(timer);
      el.style.display = 'none';
    }
    el.style.opacity = op;
    el.style.filter = 'alpha(opacity=' + op * 100 + ")";
    op -= op * 0.1;
  }, time / 10);
}
function fadeOut(el, time = 1000) {
  var op = 0.1;  // initial opacity
  el.style.display = 'block';
  var timer = setInterval(function () {
    if (op >= 1) clearInterval(timer);
    el.style.opacity = op;
    el.style.filter = 'alpha(opacity=' + op * 100 + ")";
    op += op * 0.1;
  }, time / 10);
}
```

Advantages

- JavaScript version (after pasting previous

```
fadeIn(document.querySelector("#myElement"), 500);
fadeOut(document.querySelector("#myElement"), 500);
```

- jQuery version:

```
$("#myElement").fadeIn(500);
$("#myElement").fadeOut(500);
```

So, which one do you prefer?

Install

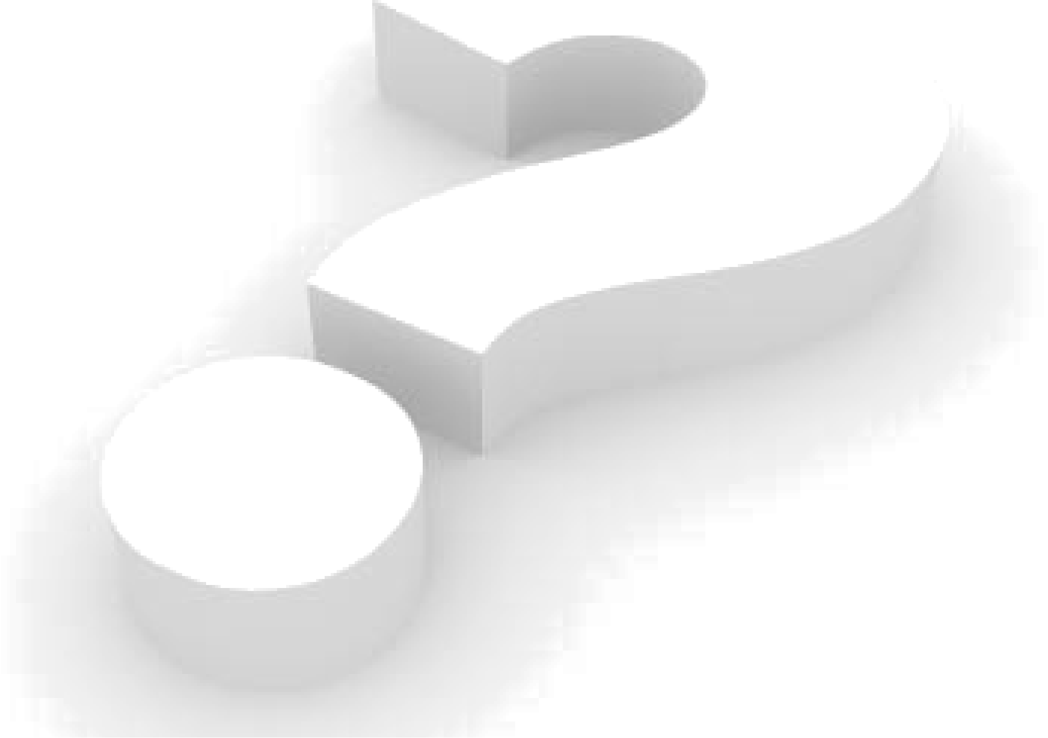
- To use jQuery library, you can download and include it in your page:

```
<script type="text/javascript" src="jquery.js"></script>
```

- If you don’t want to host jQuery file, just use the one hosted by jQuery itself:

```
<script type="text/javascript"
       src="http://code.jquery.com/jquery.min.js"></script>
```

Questions ?



jQuery

SELECTORS

Selectors

Presentation

- Selectors are used to select elements on your page and interact with them.

- Two types of selector can be
  - CSS Selectors
  - Specific jQuery Selectors

CSS selectors

- Take a look at these selectors:

Selector	Returns
*****	All elements
element	All **element** elements
#id	Element with « **id** » as id
.class	Elements with « **class** » as class value
elem[attr]	Elements with « **attr** » attribute specified

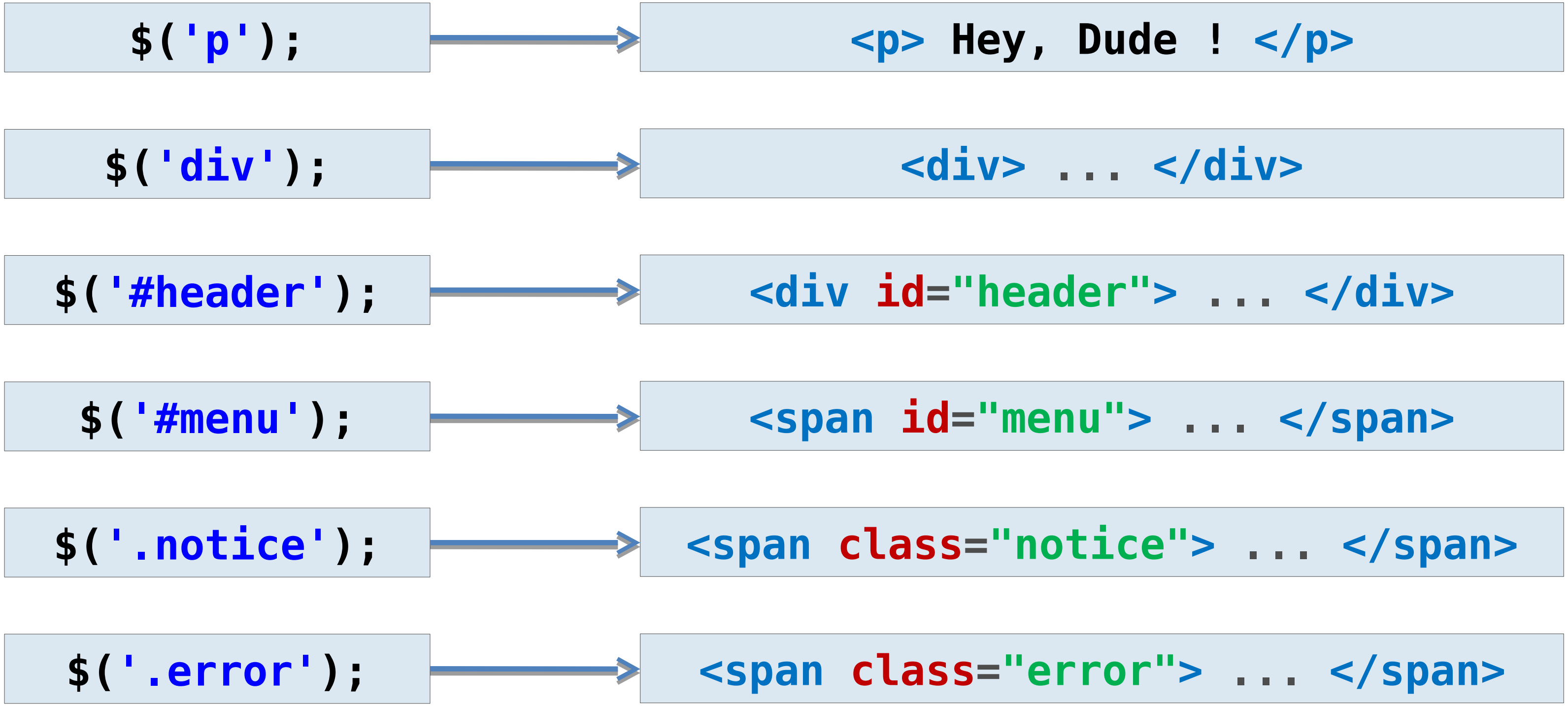
CSS selectors

- Take a look at these selectors:

Selector	Returns
elem[attr=" val"]	Elements with « **attr** » attribute specified with the value « **val** »
elem1 elem2	All « **elem2** » elements contained in all « **elem1** » elements
elem1 > elem2	All « **elem2** » elements directly inside all « **elem1** » elements
elem1 + elem2	All « **elem2** » elements immediatly preceded by « elem1 » elements
elem1 ~ elem2	All « elem2 » elements preceded by « elem1 » elements

Selectors

Example



`$('p');`	`<p> Hey, Dude ! </p>`
`$('div');`	`<div> ... </div>`
`$('#header');`	`<div id="header"> ... </div>`
`$('#menu');`	`<span id="menu"> ... </span>`
`$('.notice');`	`<span class="notice"> ... </span>`
`$('.error');`	`<span class="error"> ... </span>`

Specific jQuery selectors

- Take a look at these selectors:

Selector	Returns
:hidden	Hidden/invisible elements
:visible	Visible elements
:has("selector2")	Elements containing elements selected by the selector « **selector2** »
:contains("text")	Elements containing the text « **text** »
:not("selector2")	Elements not selected by the selector « **selector2** »

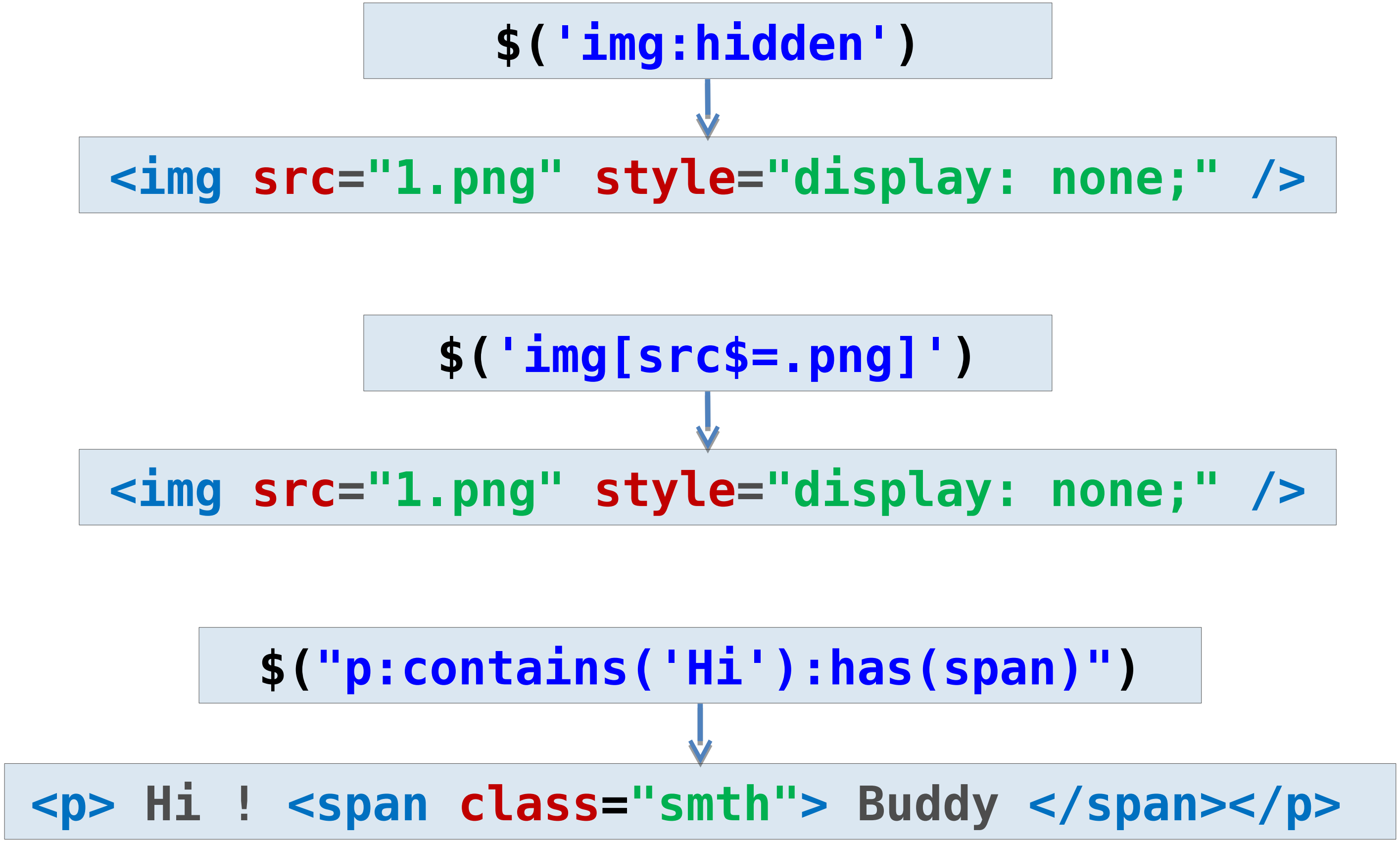
Specific jQuery selectors

- Take a look at these selectors:

Selector	Returns
:eq(number)	Returns the element at index « **number** » (zero-based)
:first	The first element. Same as :eq(0)
:last	The last element
:even	Elements with even index
:odd	Elements with odd index

Selectors

Example



```
$('img:hidden')
```

```
<img src="1.png" style="display: none;" />
```

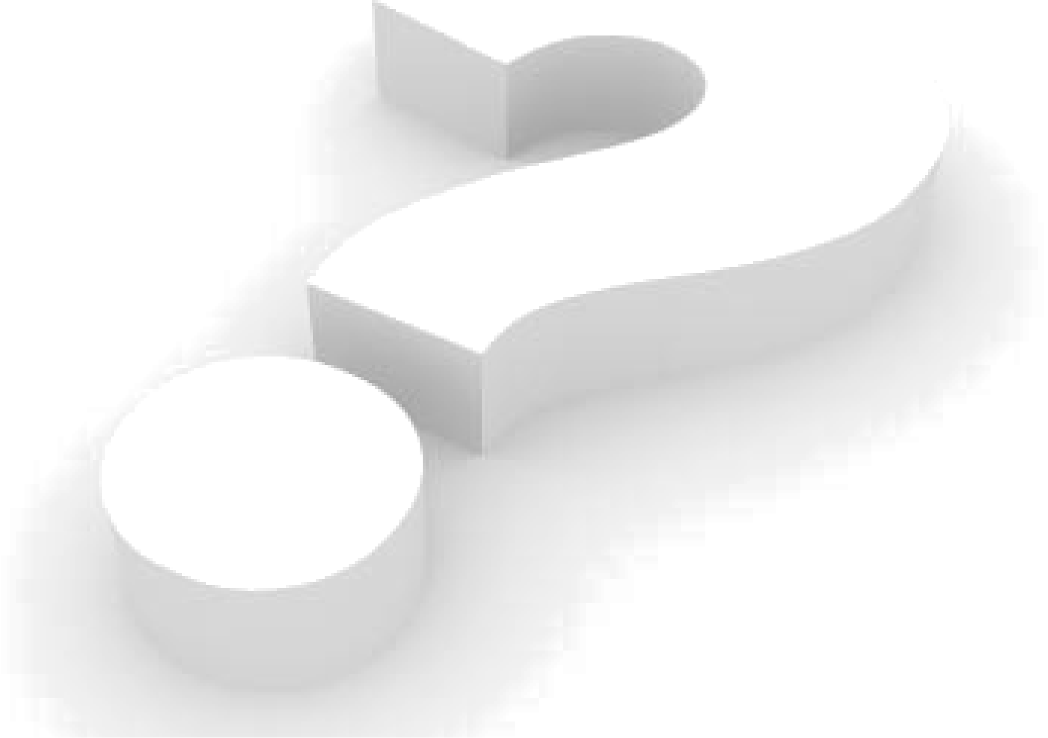
```
$('img[src$=.png]')
```

```
<img src="1.png" style="display: none;" />
```

```
$("p:contains('Hi'):has(span)")
```

```
<p> Hi ! <span class="smth"> Buddy </span></p>
```

Questions ?



jQuery

CSS MANIPULATION

CSS Functions

- Easy CSS manipulation thanks to functions below:
  - .css(propertyName)
    - Get the CSS property value

```
var color = $("div").css("background-color");
console.log(color); // outputs rgba(0, 0, 0, 0)
```

  - .css (properyName, value)
    - Set the CSS property

```
$("div").css("background-color", "cyan");
var color = $("div").css("background-color");
console.log(color); // outputs rgb(0, 255, 255)
```

CSS Manipulation

CSS Functions

- Easy CSS manipulation thanks to functions below:
  - .css (map)
    - Set all the CSS properties contained in the map

```
$("p").css({
    "background-color" : "cyan",
    "font-weight" : "bolder",
    "cursor" : "pointer"
});
```

CSS Manipulation

Show/Hide

- Show an element:

```
$("#myElement").show();
```

- Hide it:

```
$("#myElement").hide();
```

- Toggle (hide if visible, show if hidden)

```
$("#myElement").toggle();
```

CSS Manipulation

Fading

- Show an element gradually:

```
$("#myElement").fadeIn();
```

- Hide an element gradually:

```
$("#myElement").fadeOut();
```

- Toggle gradually (hide if visible, show if hidden)

```
$("#myElement").fadeToggle();
```

CSS Manipulation

Sliding

- Show an element by sliding:

```
$("#myElement").slideDown();
```

- Hide an element by sliding:

```
$("#myElement").slideUp();
```

- Toggle by sliding (hide if visible, show if hidden)

```
$("#myElement").slideToggle();
```

Animations

- If the previous animations didn’t fit your needs:
  - *.animate(properties, [duration])*
  - All animated properties should be a single numeric value
    - Properties like **width**, **height**, or **left** can be animated but **background-color** can’t
    - In addition to numeric values, each property can take « **show** », « **hide** » and « **toggle** » as values
    - Animated properties can also be relative thanks to leading **+=** and **-=** operators
  - Duration value is in milliseconds or one of the

Animations

- If the previous animations didn’t fit your needs:
  - *.animate(properties, [duration])*

```
$("#test").animate({
    left: '+=50',
    opacity: 'show'
}, 500);
```

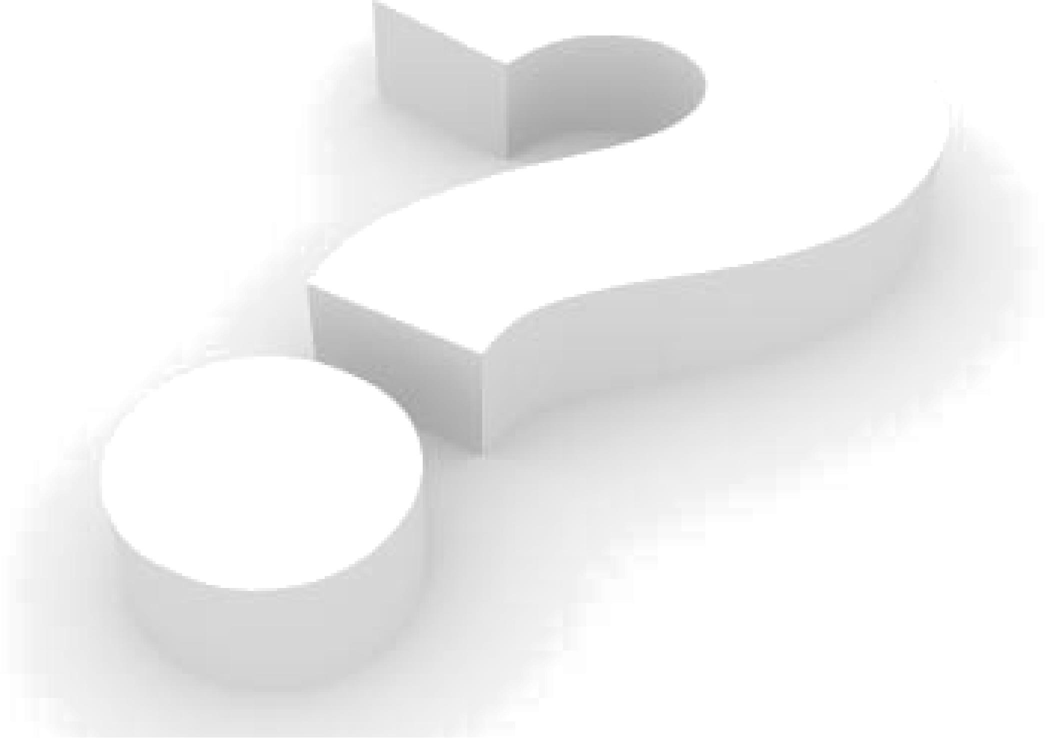
- Find a complete example of all these animations at the following URL: http://jsfiddle.net/bE4sr/1/

CSS Manipulation

Exercise

- We’re going to play with jQuery and add some CSS properties to our Resume page
  - First, install jQuery in your page
  - Then, use jQuery to add a background color to your link to Google
  - Third, your « More about me » title has to blink
  - Finally, use the animate function to add a grow effect to your main title!

Questions ?



jQuery

EVENTS

Events

Presentation

- Methods to keep clean separation of structure (HTML) and behaviours (JavaScript)

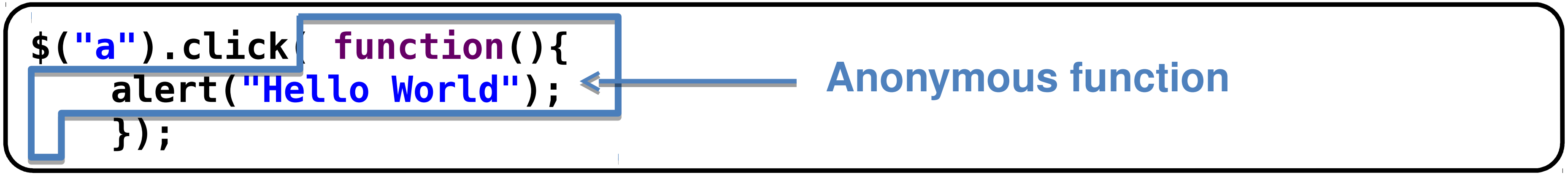
- Another way to write addEventListener

Events

Presentation

- Using:
  - $(selector).event(function() { code });

- Example:



```
$("a").click( function(){
    alert("Hello World");
    });
```

Anonymous function

Events

Presentation

- Here are some jQuery events. Just replace « handler » by the function you want to use:

Function	Description
.blur(handler)	When the element loses focus
.change(handler)	When the element’s value changes (checkbox, select, …)
.click(handler)	When the element is clicked
.focus(handler)	When the element gains focus
.submit(handler)	When the user attemps to submit a form

Events

Presentation

- Here are some jQuery events. Just replace « handler » by the function you want to use:

Function	Description
.keydown (handler)	When a key is pressed on the keyboard
.keyup (handler)	When a key is released on the keyboard
.mouseover (handler)	When the cursor is over the element
.mouseout (handler)	When the cursor leaves the element
…	…

Quizz

- What does this snippet do?

```
$("form").submit( function() {
    if ($("input:first").val() == ”admin") {
        $("span").text("Welcome Mr Admin").show();
        return true;
    }

    $("input:first").css("background-color", "red");
    $("span").text("Not valid!").show().fadeOut(1000);
    return false;
    });
```

What is « *this* » ?

- jQuery provides the **this** keyword
- Refers to the selected element inside nested functions

```
<div class="test">Rock Paper Scissors Lezard Spock</div>
<div class="test">One Two Three Four Five</div>

<script type="text/javascript">
    $(".test").hover( function() {
        $(this).css("color", "pink"); // Refers to the .test element actually being hovered
    }, function() {
        $(this).css("color", "black"); // Same
    });
    </script>
```

Callbacks

- Did you notice in the previous example?
  - The « hover » function took two anonymous functions in parameter

- The first one is launched when the event is raised

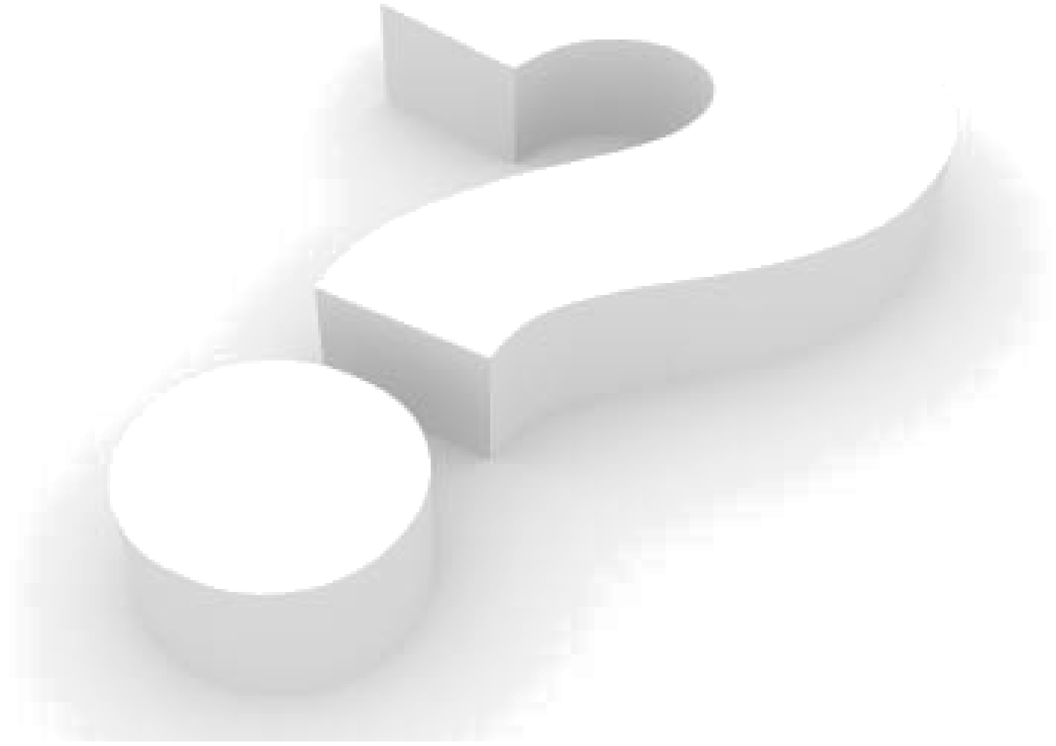
- The second one is a **callback** function
  - Remember? Executed when the event is finished

Callbacks

- Most functions that accept callbacks are animation ones:
  - *fadeIn()*
  - *fadeOut()*
  - *slideUp()*
  - *slideDown()*
  - …

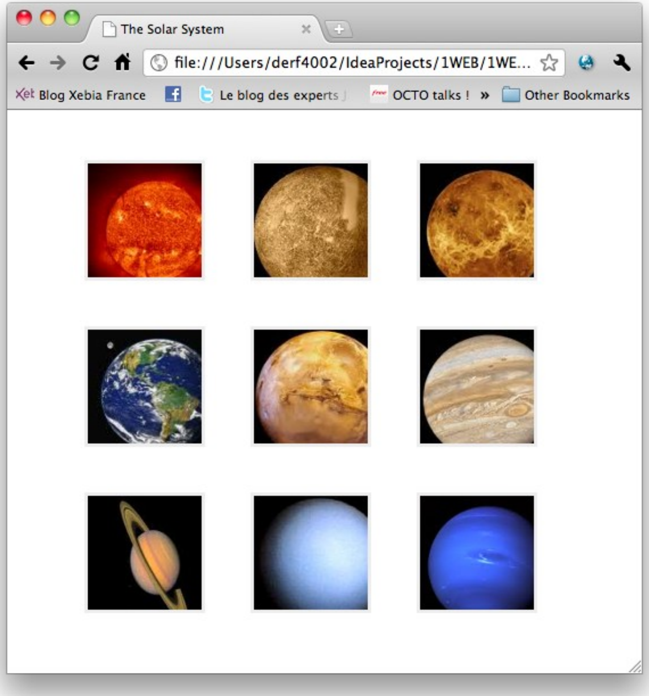
```
<div id="test">
    Rock Paper Scissors Lezard Spock
    </div>

    <script type="text/javascript">
    $("#test").slideUp(1234, function(){
        alert("You closed the div.");
    });
    </script>
```

Questions ?



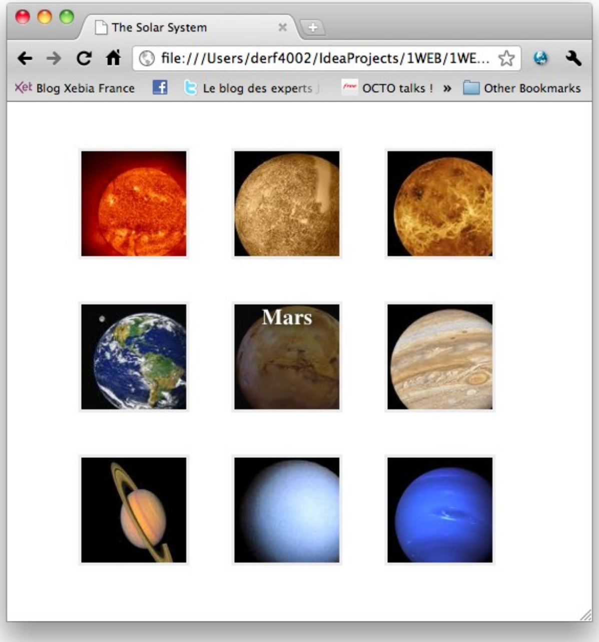
Exercise (1/2)

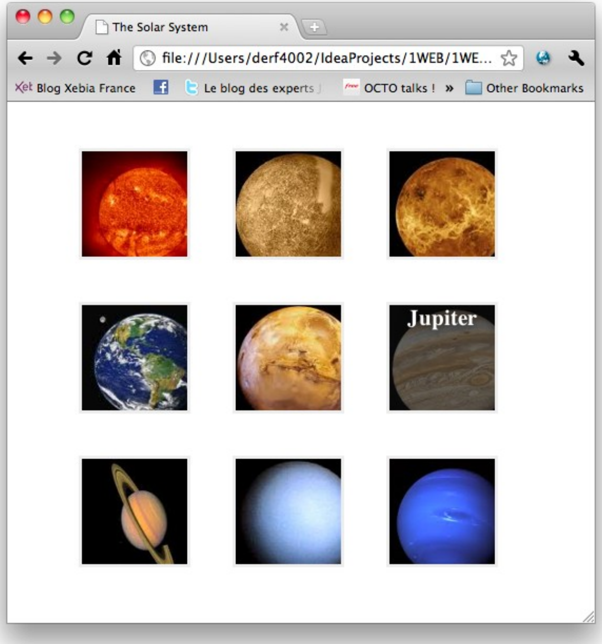
- Create a new webpage named **solar_system.html**
- The HTML must just be an unordered list of images
- Use CSS to make the page look like this :



Exercise (2/2)

- When your mouse passes over an image:
  - It has to become darker and display the planet name





jQuery

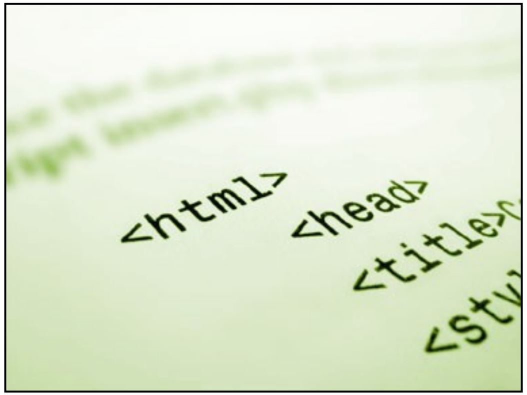
CONTENT MANIPULATION

Content manipulation

Presentation

- jQuery provides methods to add, retrieve, edit and remove HTML content

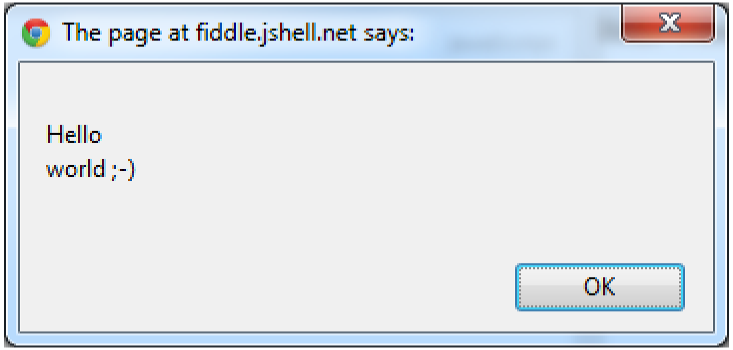
- We’re going to discover and explain them!



Textual content

- *.text()*
  - Get the combined text contents of each element in the set of matched elements, including their descendants

```
<p id="test">
    <span class="text">Hello</span>
    <span class="text">world ;-)</span>
    </p>

    <script type="text/javascript">
       var text = $("#test").text();
       alert(text);
    </script>
```



The page at fiddle.jshell.net says:

Hello
world ;-)

Textual content

- *.text(textString)*
  - Set the content of each element in the set of matched elements to the specified text

```
<p id="test"> <span class="text">Hello you ;-)</span> </p>

<script type="text/javascript">
   $("#test").text("<span>Hello by jQuery!</span>");
</script>
```

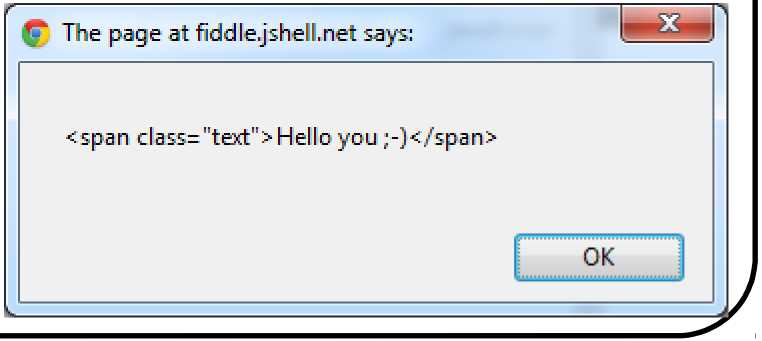
```
<p id="test">&lt;span&gt;Hello by jQuery!&lt;/span&gt;</p>
```

Content manipulation

HTML Content

- *.html()*
  - Get the HTML contents of the first element in the set of matched elements

```
<p id="test">
   <span class="text">Hello you ;-)</span>
</p>

<script type="text/javascript">
   var text = $("#test").html();
   alert(text);
</script>
```



The page at fiddle.jshell.net says:

<span class="text">Hello you ;-)</span>

Content manipulation

HTML Content

- *.html(textString)*
  - Set the HTML contents of each element in the set of matched elements

```
<p id="test"> <span class="text">Hello you ;-)</span> </p>

<script type="text/javascript">
    $("#test").html("<span>Hello by jQuery!</span>");
    </script>
```

```
<p id="test"><span>Hello by jQuery!</span></p>
```

Content manipulation

HTML Content

- *.replaceWith(textString)*
  - Replace each element in the set of matched elements with the provided new content

```
<p id="test"> <span class="text">Hello you ;-)</span> </p>

<script type="text/javascript">
   $("#test").replaceWith("<p>Hello by jQuery!</p>");
</script>
```

```
<p> Hello by jQuery! </p>
```

Insert inside an element

- *.prepend(textString)*
  - Insert some content at the beginning of each element in the set of matched elements
- *.append(textString)*
  - Insert some content at the end of each element in the set of matched elements

Insert inside an element

- Example:

```
<p>Hello by jQuery!</p>

<script type="text/javascript">
   $("p").prepend("<em>Before</em>");
   $("p").append("<em>After</em>");
</script>
```

```
<p><em>Before</em>Hello by jQuery!<em>After</em></p>
```

Content manipulation

Insert outside an element

- *.insertBefore(selector)*
  - Insert every element in the set of matched elements before the target

- *.insertAfter(selector)*
  - Insert every element in the set of matched elements after the target

Content manipulation

Insert outside an element

- Example:

```
<p>Hello by jQuery!</p>

<script type="text/javascript">
   $("<em>Before</em>").insertBefore("p");
   $("<em>After</em>").insertAfter("p");
</script>
```

```
<em>Before</em><p>Hello by jQuery!</p><em>After</em>
```

Content manipulation

Wrappers

- *.wrap(wrappingElement)*
  - Wrap an HTML structure around each element in the set of matched elements

Content manipulation

Wrappers

```
<p class="test">
   It’s gonna be legen...
</p>
<p> wait for it </p>
<p class="test"> ...dary ! </p>
<script type="text/javascript">
    $(".test").wrap("<div class='barney' />");
    </script>
```

```
<div class="barney">
   <p class="test"> It’s gonna be legen...</p>
</div>
<p> wait for it </p>
<div class="barney">
   <p class="test"> ...dary ! </p>
</div>
```

Content manipulation

Copy

- *.clone()*
  - Create a deep copy of the set of matched elements

```
<p class="test">Kage bunshin no jutsu!</p>

<script type="text/javascript”>
    $(".test").clone().appendTo("body");
    $(".test").clone().appendTo("body");
</script>
```

```
<p class="test">Kage bunshin no jutsu!</p>
<p class="test">Kage bunshin no jutsu!</p>
<p class="test">Kage bunshin no jutsu!</p>
<p class="test">Kage bunshin no jutsu!</p>
```

Content manipulation

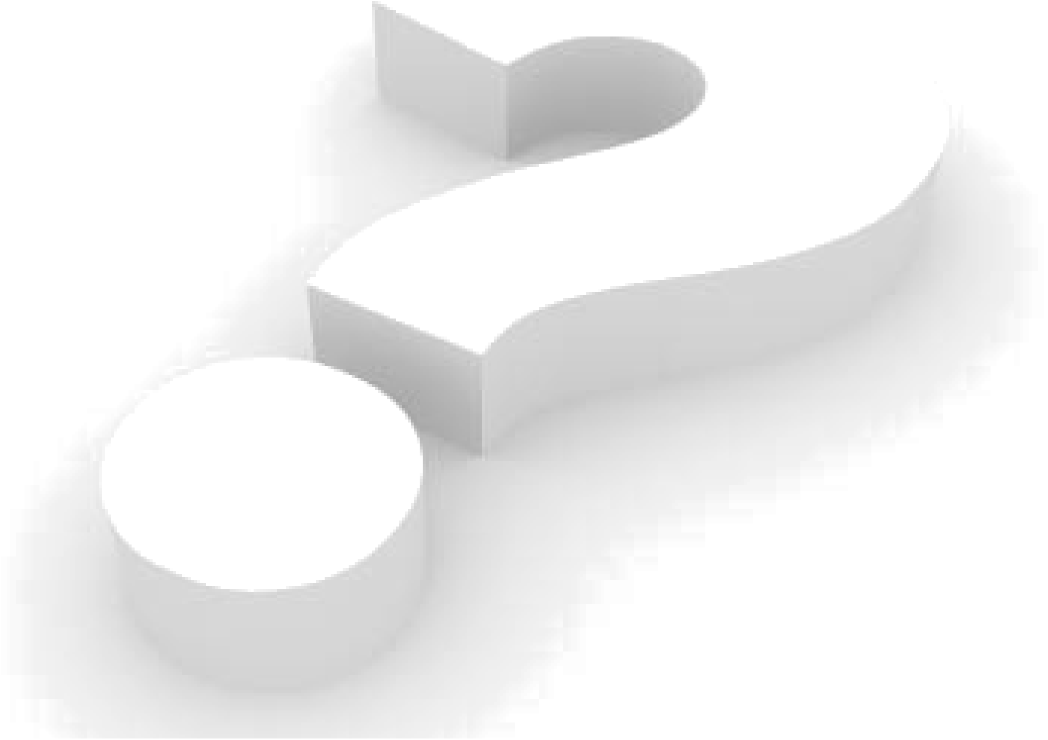
Remove

- *.remove()*
  - Remove the set of matched elements from the DOM

```
<p class="test"> It’s gonna be legen...</p>
<p> wait for it </p>
<p class="test"> ...dary ! </p>

<script type="text/javascript">
   $("p:not(.test)").remove();
</script>
```

```
<p class="test"> It’s gonna be legen...</p>
<p class="test"> ...dary ! </p>
```

Questions ?



Exercise (1/2)

- Modify your **solar_system.html** page
  - When you click on an image, it has to be displayed bigger in the center of the screen and all in background have to become darker

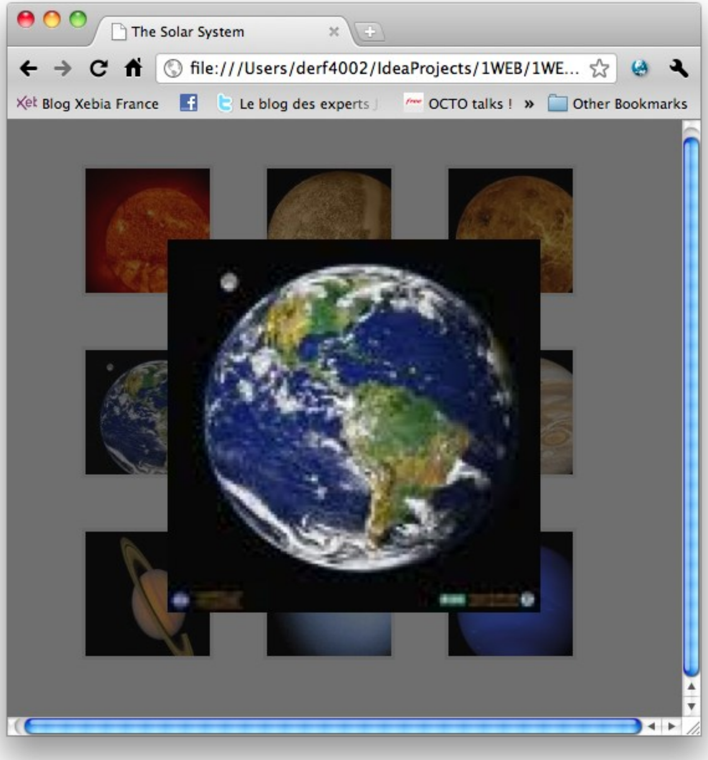
- Display the bigger image with a toggle effect

- To close and return it as the same screen as at beginning, click in the dark area

Content manipulation

Exercise (2/2)

- Your page should look like this:



Content manipulation

Demonstration

jQuery

Final Note

- jQuery makes your code easier, but:
  - It’s heavier than native JavaScript
  - Native JS updates made some jQuery functions useless

- You might not need jQuery!

  http://youmightnotneedjquery.com/