

jQuery

how to use javascript without too many problems

Introduction

***jQuery is a library of javascript functions** (a so-called software framework), cross-browser for web applications, which aims to simplify client-side programming of HTML pages.*

wikipedia

Thanks to jQuery you can write a few lines of code to call a lot of javascript functions.

why jQuery?

- HTML/DOM and CSS manipulation
 - methods for HTML events
 - effects and animations
 - AJAX programming support
- is widespread, tested and maintained
 - open source
- compact (less than 100kb, if compressed)
- can be extended with a variety of plugins

why jQuery?

Differences

code written with javascript

```
var external_links = document.getElementById('external_links');
var links = external_links.getElementsByTagName('a');
for (var i=0;i < links.length;i++) {
    var link = links.item(i);
    link.onclick = function() {
        return confirm('You are going to visit: ' + this.href);
    };
}
```

same code written with jQuery

```
$('#external_links a').click(function() {
    return confirm('You are going to visit: ' + this.href); // Il + serve a concatenare le stringhe
});
```

screens taken from wikipedia

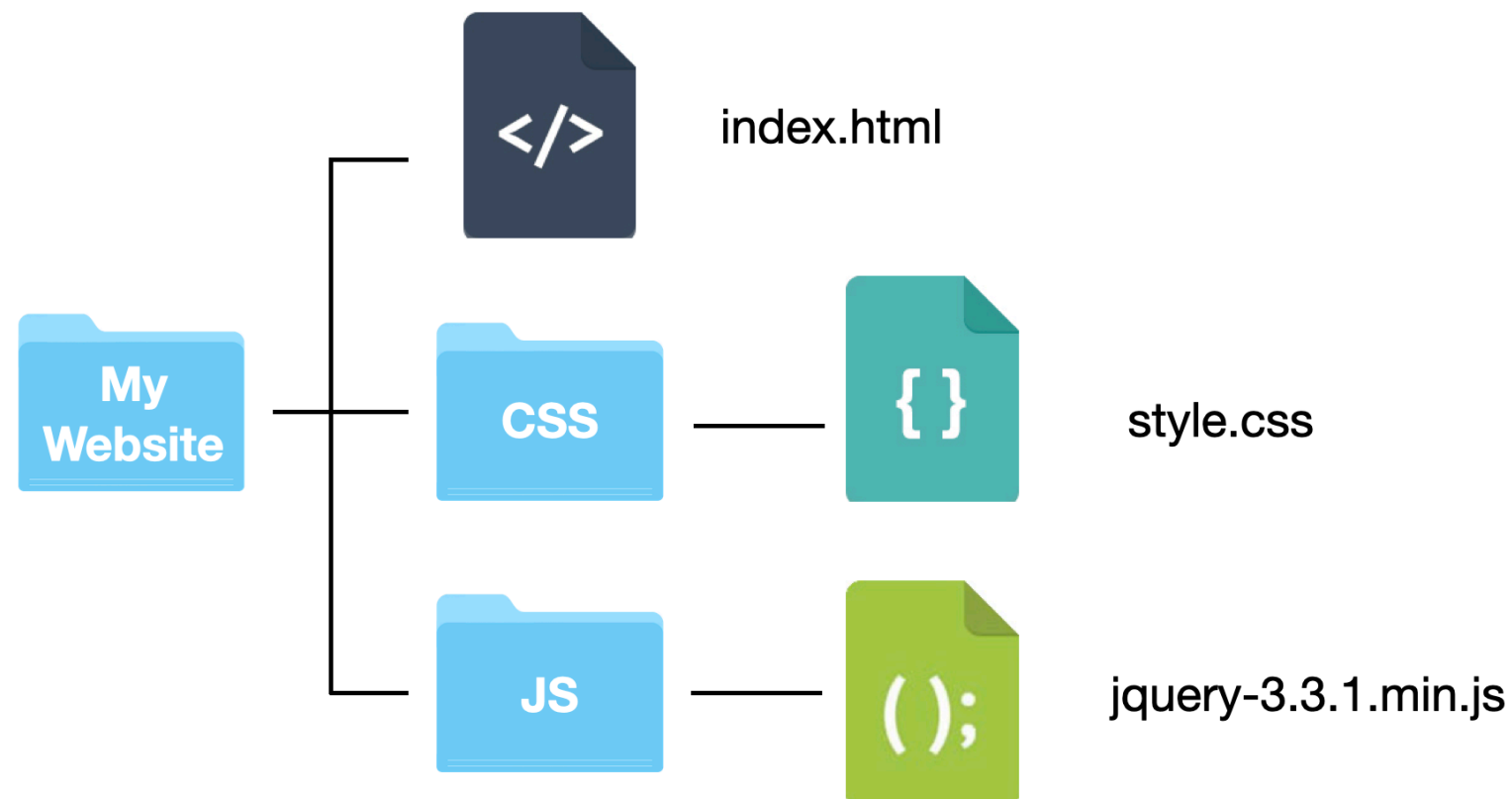
jQuery - first steps

Like a css file we need to put it inside a specific folder of our website. In this case we will create a “js” folder in which we will download and save the

jQuery library found at this link:

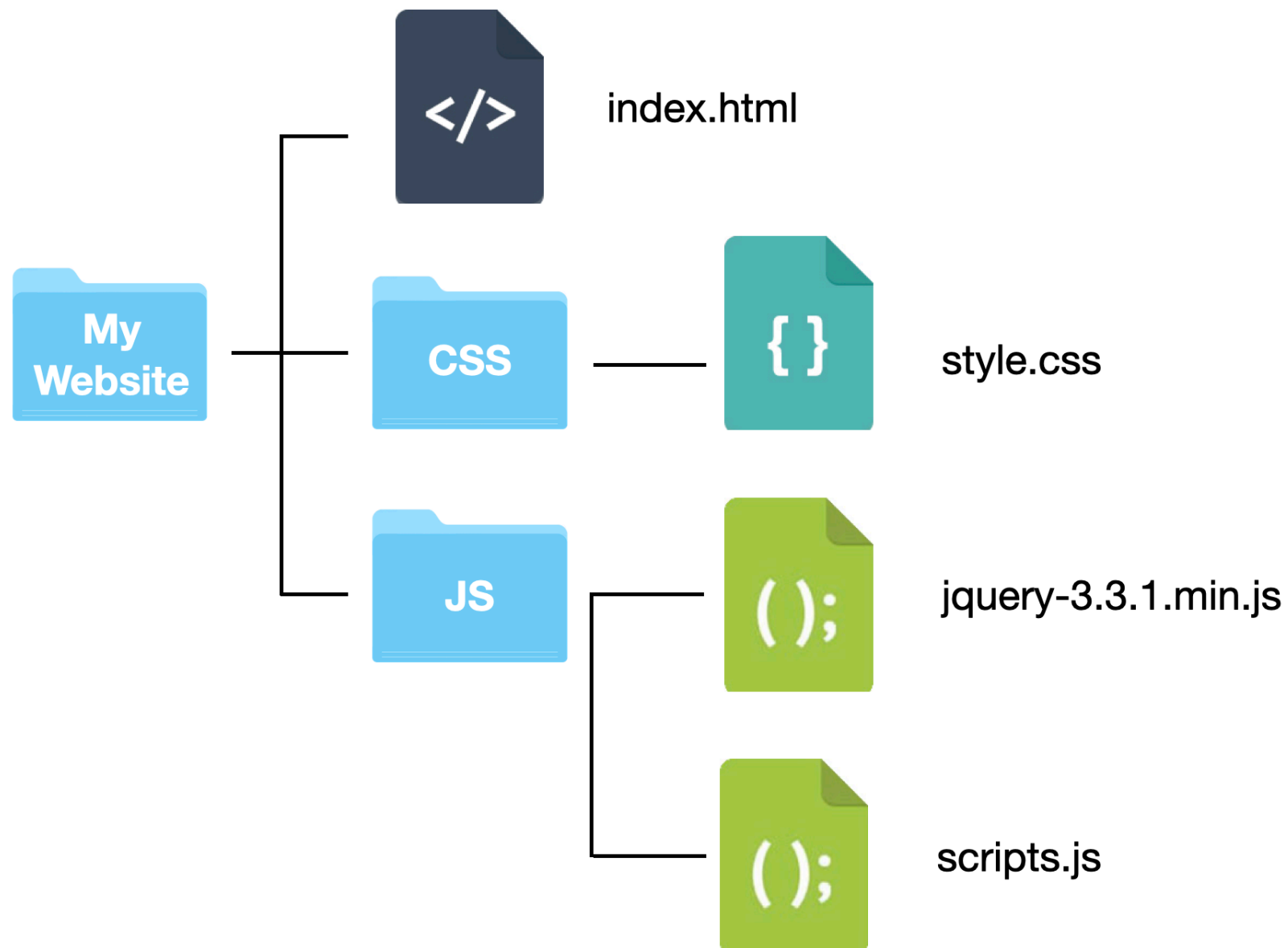
<https://jquery.com/download/>

You should download the compressed production one.



jQuery - first steps

Now you just need to create your own .js file to be added inside the “JS” folder.



HTML and jQuery

In the HTML file we need to import them.

To do so we use a generic `<script>` tag (we can use it also to write scripts directly in our html file like the `<style>` tag).

To implement the jQuery library we will write:

```
<script src="js/jquery-3.3.1.min.js" type="text/javascript"></script>
```

And for our js file in which we will write our code:

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

Structure

The jQuery method syntax is tailor-made for selecting HTML elements and performing some action on the element(s). The basic syntax is:

`$(selector).action()`

`$` sign to define/access jQuery

`(selector)` to “query (or find)” HTML elements

jQuery `.action()` to be performed on the elements. The action can execute itself or behave as an event.

Document ready event

It is really important to include all out functions inside the main event “document ready”. This will block every script to execute until the whole document has been loaded.

It is important because otherwise it could lead to some errors (can't execute a script on an element that didn't load yet).

The syntax is the following (red):

```
$(document).ready(function(){  
    $(selector).action();  
});
```

selectors

jQuery uses mostly the css selectors to find and apply actions to the html elements. So we can use element, class and id selectors just like in the css.

`$("p").hide()` - hides all `<p>` elements.

`$(".test").hide()` - hides all elements with `class="test"`.

`$("#test").hide()` - hides the element with `id="test"`.

`$(p.intro).hide()` - hides all `<p>` elements with `class="test"`.

`$("*").hide()` - hides all elements.

In addition it uses some own custom selectors

`$(this).hide()` - hides the current element.

<https://api.jquery.com/category/selectors/>

events

All the different visitors' actions that a web page can respond to are called events.

An event represents the precise moment when something happens.

Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

The most common events are:

Mouse events: click, dblclick, mousedown, mouseup, mouseenter, mouseleave, hover

Keyboard events: keypress, keydown, keyup

Form events: submit, focus, blur

Document/Window events: resize, scroll

Events mouse

You can set events depending on the mouse condition:

```
$("p").click(); //On clicking any paragraph//
```

```
$("p").dblclick(); //On double-clicking any paragraph//
```

```
$("p").mousedown(); //On holding down the mouse over any paragraph//
```

```
$("p").mouseup(); //On releasing the mouse over any paragraph//
```

```
$("p").mouseenter(); //On going over any paragraph with the mouse//
```

```
$("p").mouseleave(); //On leaving any paragraph with the mouse//
```

Events mouse

A special note to the .hover() event. You can use 2 events: .mouseenter and .mouseleave or just the .hover event where you specify two functions for it instead of one function for each event.

Instead of:

```
$("#box").  
mouseenter(function(){  
    alert("You entered box!");  
});  
$("#box").  
mouseleave(function(){  
    alert("Bye! You now leave  
box!");  
});
```

You can use:

```
$("#p1").hover(function(){  
    alert("You entered p1!");  
},  
function(){  
    alert("Bye! You now leave  
p1!");  
});
```

Either way is correct.

Events keyboard

You can set events depending on the keyboard condition:

```
$(“input”).keypress(); //On pressing any key inside an input element//
```

```
$(“input”).keydown(); //On pressing down any key inside an input element//
```

```
$(“input”).keyup(); //On releasing any key inside an input element//
```

Events form

You can set events depending on the form elements condition:

```
$(“form”).submit(); //On pressing the submit button of the form//
```

```
$(“input”).focus(); //On clicking inside the input element//
```

```
$(“input”).blur(); //On clicking outside the input element//
```

<https://api.jquery.com/category/forms/>

Events document/window

You can set events depending on the document/window condition:

```
$(window).resize(); //On resizing the browser's window//
```

```
$(window).scroll(); //On scrolling the browser's window//
```


Events

```
$(“p”).click(); //On clicking any paragraph//
```

But with each event we need then to specify what happens next.
Just like we saw in the document ready event.

```
$(“p”).click(function(){ //On clicking any paragraph//  
    $(this).hide(); //the paragraph clicked will hide//  
}); //close the function and click event//
```

Events

The complete syntax will be:

```
$(document).ready(function(){ //When the page is fully loaded//  
    $("p").click(function(){ //On clicking any paragraph//  
        $(this).hide(); //the paragraph clicked will hide//  
    }); //close the function and click event//  
}); //close the function and ready event//
```

Methods

Methods are what really happens to the elements, once we defined a selector and (maybe) an event.

Since jQuery is a javascript library we can use all the Javascript methods.

The most used are anyway these ones:

Animations: hide, show, toggle, slide, fade, animate

CSS Manipulation: addClass, removeClass, toggleClass, css

Misc: alert

Methods animations

You can hide and show HTML elements with the `hide()` and `show()` methods:

```
$("#hide").click(function(){ //When you click the element with hide id//  
    $("p").hide(); //all the paragraphs will be hidden//  
});
```

```
$("#show").click(function(){ //When you click the element with show id//  
    $("p").show(); //all the paragraphs will be shown//  
});
```

Methods animations

You can toggle between the **hide()** and **show()** methods with the **toggle()** method

```
$(“button”).click(function(){ //When you click the button element//  
    $(“p”).toggle(); //all the paragraphs will show if hidden and hide if shown//  
});
```

Methods animations

You can create a sliding effect on elements with the **slideDown()**, **slideUp()** and **slideToggle()** methods:

```
$("#flip").click(function(){ //When you click the element with flip id//  
    $("#panel").slideDown(); //the element with panel id will be opened from the top to bottom//  
});
```

```
$("#flip").click(function(){ //When you click the element with flip id//  
    $("#panel").slideUp(); //the element with panel id will close from the bottom to top//  
});
```

```
$("#flip").click(function(){ //When you click the element with flip id//  
    $("#panel").slideToggle(); //the element with panel id will be open or close//  
});
```

You can also specify a duration of the effect: “slow”, “fast” or in milliseconds.

Methods animations

You can create a fading effect on elements with the **fadeIn()**, **fadeOut()** and **fadeToggle()** methods:

```
$(“button”).click(function(){ //When you click the button element//  
    $(“#fade”).fadeIn(); //the element with fade id will appear gradually//  
});
```

```
$(“button”).click(function(){ //When you click the button element//  
    $(“#fade”).fadeOut(); //the element with fade id will disappear gradually//  
});
```

```
$(“button”).click(function(){ //When you click the button element//  
    $(“#fade”).fadeToggle(); //the element with fade id will appear or disappear gradually//  
});
```

You can also specify a duration of the effect: “slow”, “fast” or in milliseconds.

Methods manipulation

You can add classes to one or more elements with the **addClass()** method:

```
$(“button”).click(function(){ //When you click the button element//  
    $(“h1, h2, p”).addClass(“blue”); //the h1, h2 and paragraph elements will have the class blue added//  
    $(“div”).addClass(“red”); //the div element will have the class red added//  
});
```

You can use more than one selector separated by a comma.

Methods manipulation

You can remove classes to one or more elements with the **removeClass()** method:

```
$(“button”).click(function(){ //When you click the button element//  
    $(“h1, h2, p”).removeClass(“blue”); //the h1, h2 and paragraph elements will have the class blue  
    removed//  
});
```

You can use more than one selector separated by a comma.

Methods manipulation

You can add a css property to one or more elements with the **css()** method:

```
$(“button”).click(function(){ //When you click the button element//  
    $(“p”).css(“background-color”, “yellow”); //all the paragraphs will have a yellow  
background color//  
});
```

The syntax is:

```
$(selector).css(“property”, “value”); //with a comma separating the property name and  
the value//
```

If you want to add more than one property the syntax is:

```
$(selector).cs({“property”: “value”, “property”: “value”}); //with a semicolon sep-  
arating the property and the value and a comma separating the different properties + values//
```

<https://api.jquery.com/category/manipulation/>

Methods misc

You can create a system pop-up message with the **alert()** method:

```
$("#box").mouseenter(function(){ //When you go with the mouse over the element with  
box id//  
    alert("You entered box!"); //An alert with a custom message will be shown//  
});
```

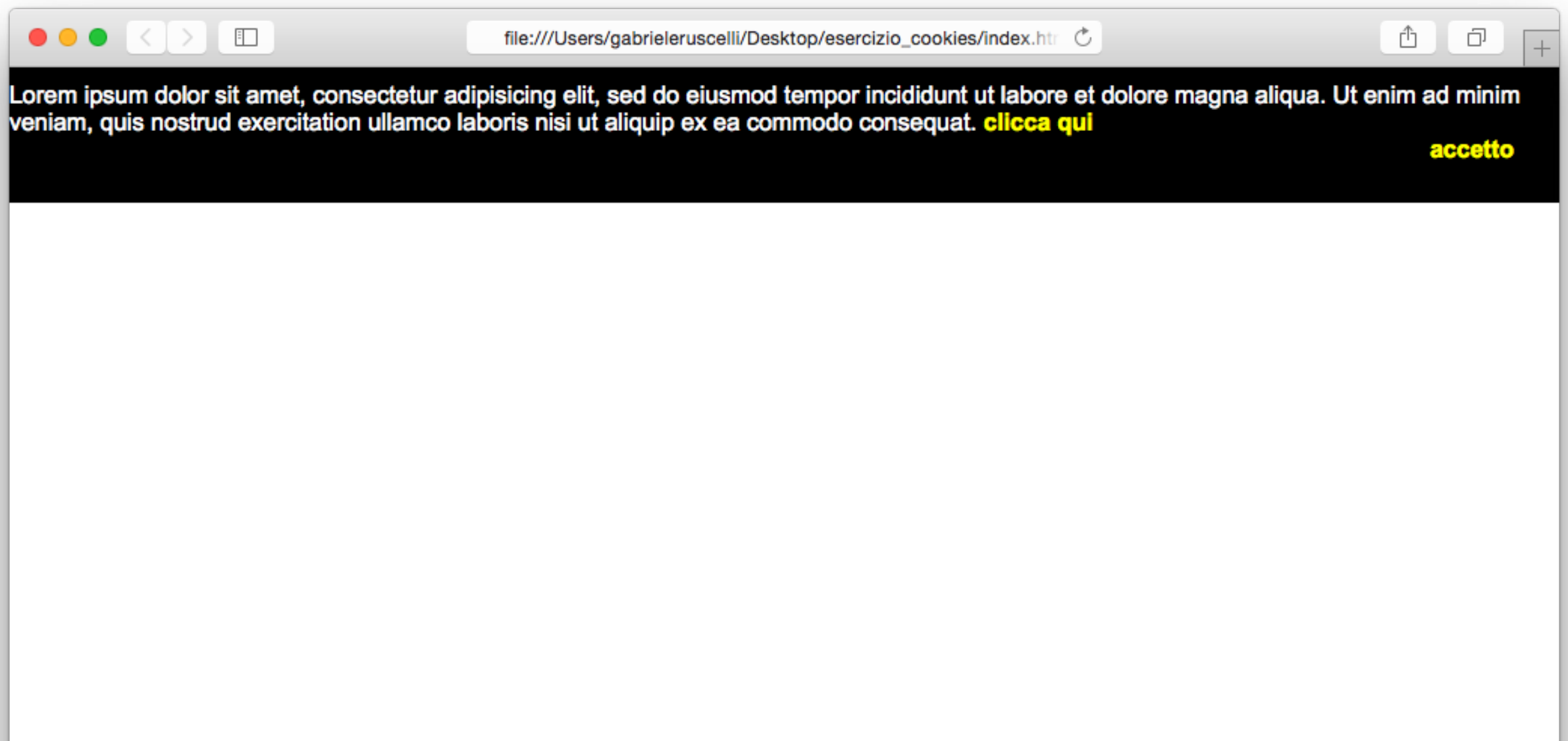
```
$("#box").mouseleave(function(){ //When you leave with the mouse the element with box  
id//  
    alert("Bye! You now leave box!"); //An alert with a custom message will be shown//  
});
```

Exercise 1

Create a bar that informs users that the site uses cookies.

1st step: use HTML and CSS for construction and style;

2nd step: use jQuery to implement the function that allows the box to disappear.



Exercise 2

Create a filtering system that allows you to delete some colors once you click on the name.

N.B. pay attention to the block and button id names

