

JAVASCRIPT & JQUERY

THE CLIENT-SIDE SCRIPT

Hate it or love it, in any case you're stuck with it ;)

Client-Side: Your browser is running it.

WHAT'S WITH THE NAME?

JavaScript is to Java what Carpet is to Car.

OBJECT-ORIENTED

Sort of...

Understanding "Prototypes" in JavaScript

TYPES ARE CLOSE TO RUBY ONES

Ruby

```
"Hello Le Wagon"      # String
'Hello Le Wagon'      # String

42                     # Fixnum
42.0                   # Float

true != false         # Boolean
```

JavaScript

```
"Hello Le Wagon"      // String
'Hello Le Wagon'      // String
```

TYPES ARE CLOSE TO RUBY ONES

Ruby

```
[ 'Hello', 'Le', 'Wagon', 42 ]      # Array

{ name: 'bob', age: 42 }             # Hash
{ 'name' => 'bob', 'age' => 42 }      # Hash (a different one)
```

JavaScript

```
[ 'Hello', 'Le', 'Wagon', 42 ]      // Array

{ name: 'bob', age: 42 }             // Object
{ 'name': 'bob', 'age': 42 }         // Object (the exact same)
```

Take a break and laugh a bit

VARIABLES NEED AN EXPLICIT SCOPE

Ruby

```
hello = 'Hello Le Wagon'
```

JavaScript

```
var hello = 'Hello Le Wagon';  
^ a local variable           ^ get use to this little one ;)
```

SMALL DIFFERENCES

Ruby

```
if condition  
  do_something  
end
```

```
while condition  
  do_something  
end
```

JavaScript (notice the camel-cased doSomething ?)

```
if (condition) {  
  doSomething();  
}
```

```
while (condition) {  
  doSomething();  
}
```


FOR LOOPS

Ruby

```
10.times { |i| do_something(i) }  
  
# OR (to be fair)  
  
for i in 0...10  
  do_something(i)  
end
```

JavaScript

```
for (var i = 0; i < 10; i++) {  
  doSomething(i);  
}
```

FUNCTIONS

Ruby

```
def my_method(a, b)  
  a.do_something(b)  
end
```

JavaScript

```
function myMethod(a, b) {  
  return a.doSomething(b); // Explicit return is back! Yeah..  
}
```

KEYWORDS

Ruby

```
nil  
self
```

JavaScript

```
null  
this
```

VARIABLES SCOPE

```
var globalVar = 0; // | global scope
                    // |
function aFunction() { // | function scope
    var localVar = 2; // |
    anotherVar = 'very bad!'; // | <<< This is BAI
                    // |
    if(localVar == 2) { // |
        var globalVar = 3; // |
        localVar += globalVar; // |
    } // |
    return localVar + globalVar; // |
} // |
                    // |
alert(aFunction()); // |
```

JAVASCRIPT OBJECT

```
var bob = {  
  name: 'Bob',  
  age: 42  
};
```

```
bob[ 'name' ] // => "Bob"
```

```
bob.name      // => "Bob"      It's an Object, not a simple Hash
```

```
bob.name = "Bobby"
```

```
bob[ 'name' ] // => "Bobby"
```

FUNCTIONS CAN BE STORED IN VARIABLES

```
function myFunction(a, b) {  
  return a + b;  
}  
myFunction(2, 3); // => 5
```

```
var myFunction = function(a, b) {  
  return a + b;  
}  
myFunction(2, 3); // => 5
```

USING JAVASCRIPT WITH HTML

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <script src="js/some_script.js"></script>
  </body>
</html>
```

Loading a JS file is blocking the page rendering so putting it at the end is a good practice.

JQUERY

<http://jquery.com/>

WHAT IS IT?

A JavaScript library to ease your life.

CROSS-BROWSER STANDARDIZATION

THE \$ () FUNCTION

Will find everything in your HTML (DOM traversal)

```
$( 'div' )           // finds every <div> in the page
$( '.hello' )        // finds every tag with a "hello"
                     //   class="wagon hello" works fine
$( '#wagon' )        // finds the one element with id='wagon'
$( 'td.title > a' )  // any CSS selector will do
```

PLAY WITH ELEMENTS

```
$( '#wagon' ).hide();
```

```
$( '#wagon' ).show();
```

```
$( '.hello' ).css({ color: red; })
```

```
// It's chainable!
```

```
$( '.selected > a' ).addClass('selected').removeClass('left-alc
```

WE NEED TO GO DEEPER!

```
$('#wagon').find('.student').hide();    // Are you still there?  
  
// The code above is faster than  
$('#wagon .student').hide();
```

GET SOME CONTENT

```
$( 'a' ).text()           // => ['read more', 'click here', ...]  
$( 'div' ).html()         // => ['<p>...', '...</div>']
```

```
$( 'a.hello' ).text( 'Hello Le Wagon!' );  
$( '#wagon' ).html( '<p>Here is some <strong>HTML</strong></p>' );
```

BE THE CREATOR

You can create new elements

```
var para = $('<p>').text("Hello I'm new here");  
  
var div = $('<div>');  
para.appendTo(div);  
  
$('body').append(div);
```

A new element can only be inserted once in the page.

GIVE ME MORE

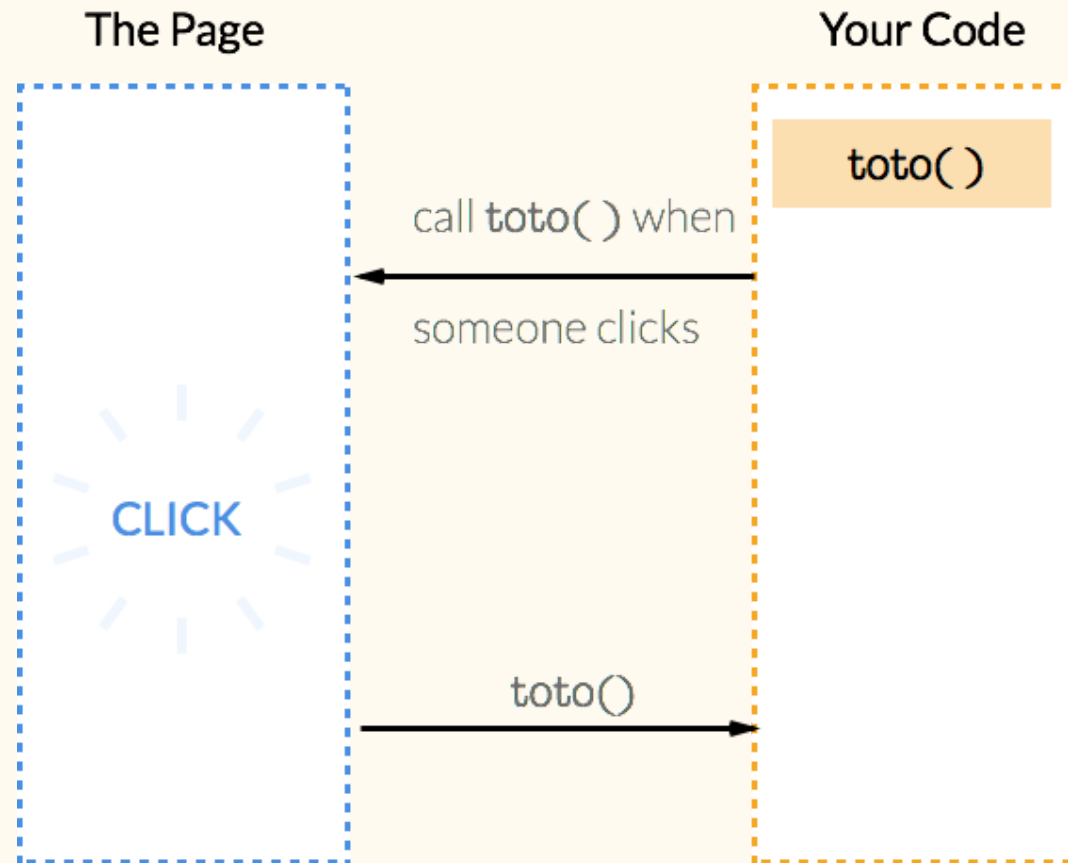
Find out on <http://api.jquery.com/>

EVENTS



Not that kind of event...

REACT TO EVENTS



SETUP AN EVENT HANDLER

```
// A common practice is to use anonymous functions as event handlers
$('img').on('click', function(the_event) {
    $(this).addClass('round-corner'); // this is the current element
});
```

Event Handlers are sometimes called Callbacks.

THE MOST IMPORTANT EVENT

When the page is ready for your code

```
$(document).ready(function() {  
    // executed when the document is ready!  
});
```

There is a shortcut

```
$(function() {  
    // executed when the document is ready!  
});
```

USEFUL FUNCTIONS (1/3)

DOM manipulation

- `.html()` / `.text()`
- `.attr()` / `.prop()`
- `.append()`
- `.prepend()`
- `.width()` / `.height()`

USEFUL FUNCTIONS (2/3)

CSS-related

- `.css()`
- `.addClass()`
- `.removeClass()`
- `.hasClass()`
- `.toggleClass()`

USEFUL FUNCTIONS (3/3)

Effects & animation

- `.animate()`
- `.show()` / `.hide()`
- `.fadeIn()`
- `.fadeOut()`
- `.slideDown()`
- `.slideUp()`

HAPPY HACKING!