jQuery

jQuery or JS?

- **Fast** and **concise** JavaScript library
- Simplifies
  - HTML document traversing
  - Event handling
  - Animations
  - AJAX
- Some instructions start with “$”

jQuery - Files

Needs to be imported in the **<head>** tag

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

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

It should go **before** the **<script>** tag, and before any other library that uses jQuery.

jQuery - DOM traversing

- ready and onload

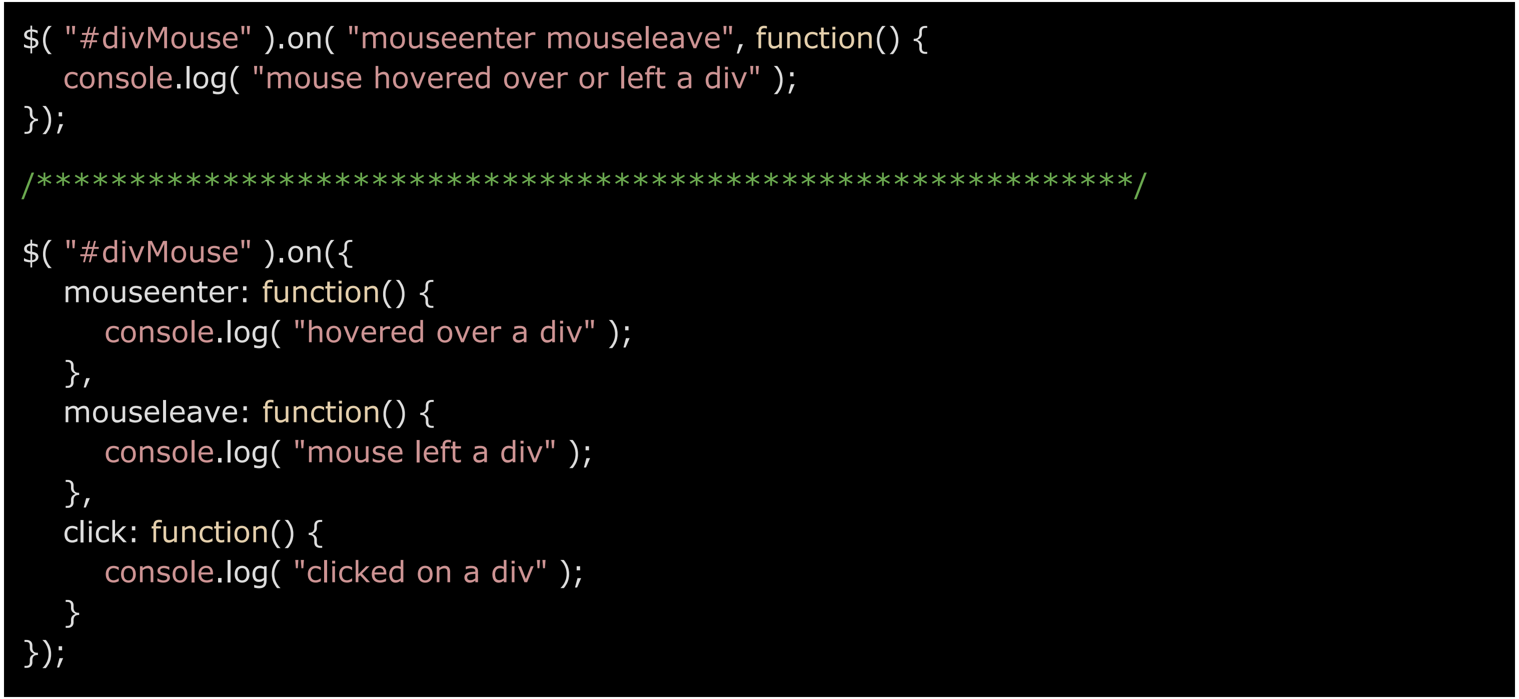
```
// ready: when DOM is loaded
$( function() {…})
$().ready( function() { … })
$( document ).ready( function() { … })
$( "document" ).ready( function() { … })
$( "img" ).ready( function() { … })

//onload: when everything is loaded (img, iframes, etc)
$( window ).on( "load", function() { … })
```

jQuery - DOM traversing

- Shorter functions, more intuitive

```
// getElementByID is now just $( "#name" )
// .css( "property" , "value" )
$( "#container" ).css( "color" ,  "blue" );

// getElementsByTag is $( "tag" ), and getElementsByClass is $( ".class" )
// .forEach(function()): loop each of the elements in an array
$( "p" ).forEach( function() {
      $( "this" ).parent().addClass( "text" );
 });
```

jQuery - Event handling



```
$( "#divMouse" ).on( "mouseenter mouseleave", function() {
   console.log( "mouse hovered over or left a div" );
});

/************************************************************/

$( "#divMouse" ).on({
   mouseenter: function() {
      console.log( "hovered over a div" );
   },
   mouseleave: function() {
      console.log( "mouse left a div" );
   },
   click: function() {
      console.log( "clicked on a div" );
   }
});
```

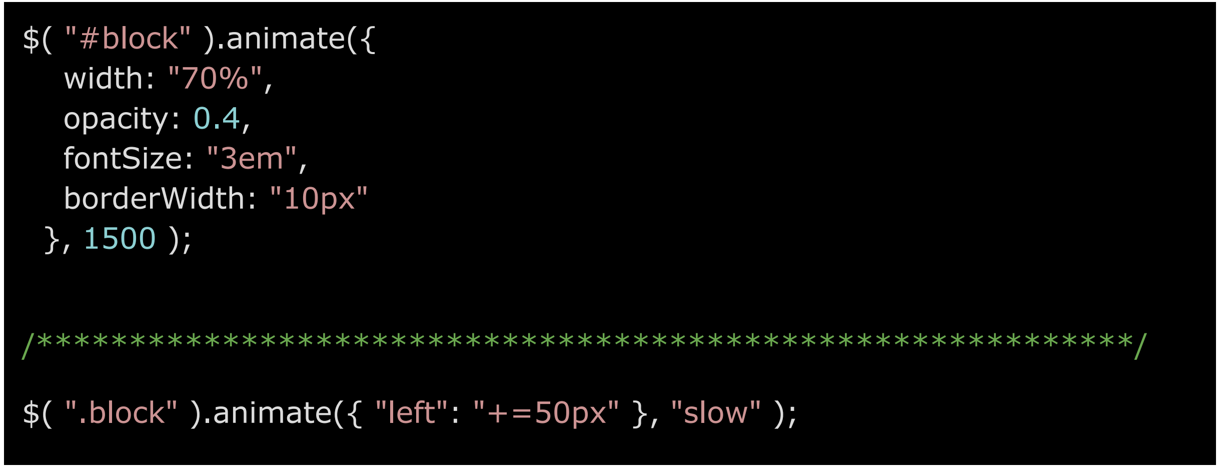
jQuery - Event handling

```
$( "button" ).forEach(function() {
        $("this").click(function() {
                console.log( "yay you clicked me" );
        });
});

$( "#name-input" ).change(function() {
   console.log( "focus, change, lost focus" );
});

$( "#name-input" ).keydown(function() {
   console.log( "each time a key is pressed inside this input" );
});

$( ".first-fieldset" )[1].hover(function() {
   console.log( "this are not the droids you were looking for" );
});
```

jQuery - Animations

JS: using **setInterval** function

jQuery: you set the result in the animate method



```
$( "#block" ).animate({
   width: "70%",
   opacity: 0.4,
   fontSize: "3em",
   borderWidth: "10px"
  }, 1500 );


/************************************************************/

$( ".block" ).animate({ "left": "+=50px" }, "slow" );
```

jQuery and CSS animations

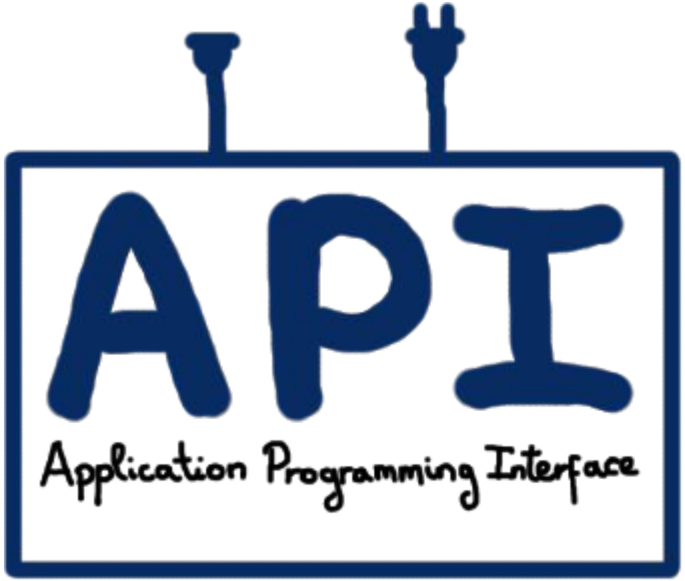
Animation can be a little jumpy the 1st time around...

Play it Again!

Want to learn how this thing works?

Read the Article

by: ANTHONY CALZADILLA
follow me on twitter!


API
Application Programming Interface

AJAX
Asyncronous JavaScript And XML

AJAX?

- **A**synchronous **J**avaScript **A**nd **X**ML
- To **send and retrieve data** from a server **asynchronously** (in the background), without interfering with the display and behaviour of the webpage

API

- **A**pplication **P**rogramming **I**nterface
- Functions and procedures that allow, using applications, the **communication between services** like web servers, OS, etc.
- Gives access to **features and data**

API

https://www.pokeapi.co/api/v2/pokemon/549/

https://swapi.co/api/planets/1/

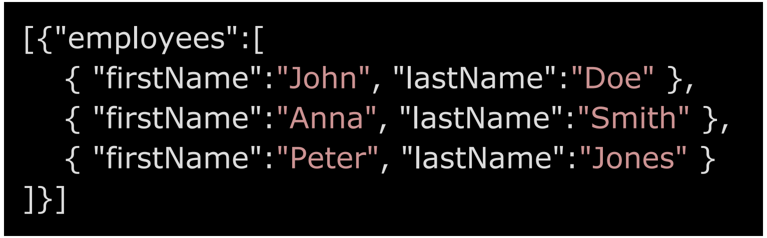
https://swapi.co/api/people/1/?format=json

http://steamspy.com/api.php?request=top100in2weeks

http://steamspy.com/api.php?request=genre&genre=Early+Access

JSON

- **J**ava**S**cript **O**bject **N**otation
- Syntax for storing and send/receive data
- Text, written in JavaScript Object notation
  - Data is name/value pairs, separated by commas
  - Curly braces ({}) hold objects
  - Square braces ([]) hold arrays



```
[{"employees":[
   { "firstName":"John", "lastName":"Doe" },
   { "firstName":"Anna", "lastName":"Smith" },
   { "firstName":"Peter", "lastName":"Jones" }
]}]
```

HTTP

- **H**yper**t**ext **T**ransfer **P**rotocol: to enable communication between client and server. It uses name/value pairs.
- Methods
  - GET : parameters in the URL

```
https://www.mywebsite.com/main_form.php?name1=value1&name2=value2
```

  - POST: parameters are hidden (forms, etc.)

```
/** Package sent when submitting a form**/
POST /main_form.php HTTP/1.1
Host: mywebsite.com
name1=value1&name2=value2
```

jQuery - AJAX

The **.ajax()** method can be GET or POST

```
$.ajax({
 type: "POST",  //default value is GET
 url: url,
 data: data,
 success: success,
 dataType: dataType
});
```

jQuery - GET

```
/*
*  $.get(url,{parameter:value}, function(what_the_get_returns){...}, json=convert to json)
*/

$.get( "/form", { name: "John", surname: "Jones", "choices[]": ["Football", "Basketball"] },
function( data ) {
   /* data=
   * {fullname:”John Jones”, sports: “Football and basketball”}
   */
  $( "body" )
    .append( "Name: " + data.fullname )
    .append( "Sports: " + data.sports );
}, "json" );
```

jQuery - POST

```
$.post( url, obj_to_send, function( data ) {
  // data = data received
  // as we used the "json" argument at the end, data is parsed JSON
  alert(data.name);
  /*
    if we have received {name:"john"}, data.name = john

    if the object is like [{name:"john"}], to retrieve "john" we would
        need data[0].name

    if the object is like {name:["john","chris"]}, to get "chris" we
        would need data.name[2]
  */

}, "json" );
```

jQuery - JSON

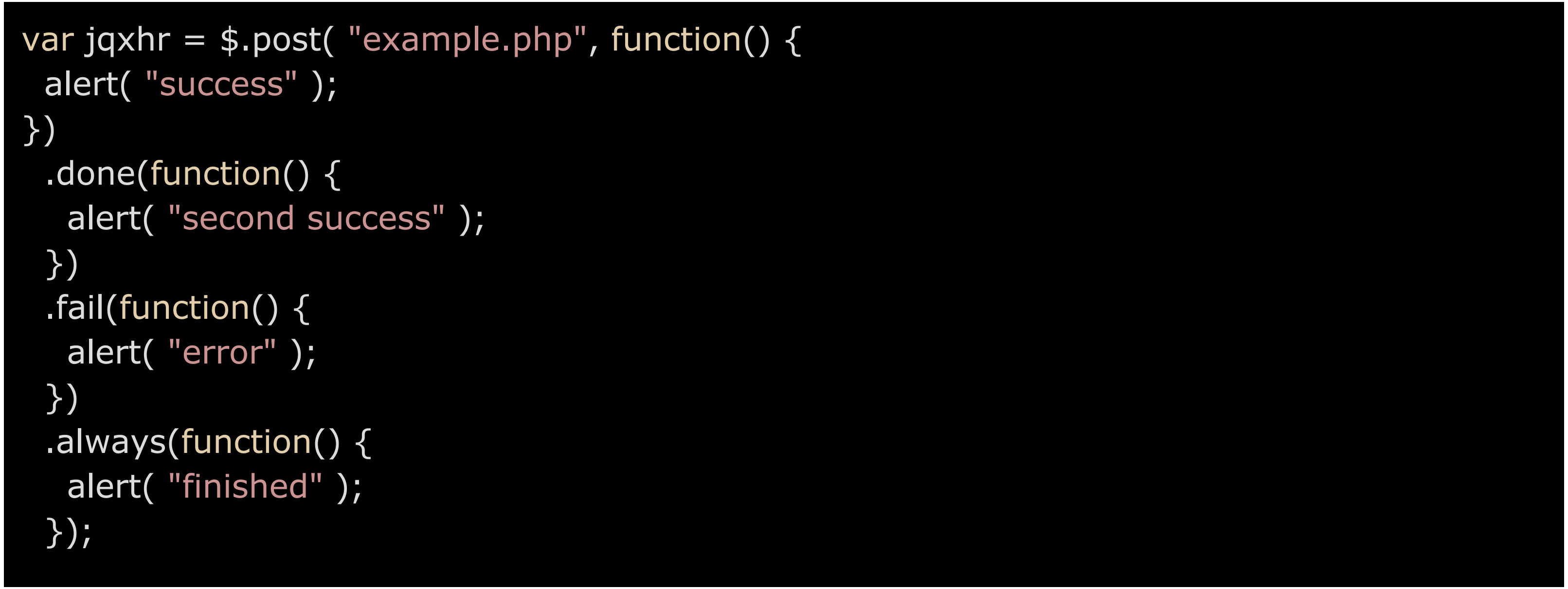
- Convert string to object

```
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';
var myObj = JSON.parse(myJSON);
```

- Convert object to string

```
var myObj = { "name":"John", "age":31, "city":"New York" };
var myJSON = JSON.stringify(myObj);
```

jQuery - AJAX events



```
var jqxhr = $.post( "example.php", function() {
  alert( "success" );
})
  .done(function() {
    alert( "second success" );
  })
  .fail(function() {
    alert( "error" );
  })
  .always(function() {
    alert( "finished" );
  });
```