

# Introducción a jQuery

Primeros pasos con la librería Javascript

# Miguel Angel Alvarez

@midesweb

malvarez@desarrolloweb.com







# #dwjQuery



"Comenzamos jQuery en @EscuelaIT con @midesweb"

# ¿Qué es jQuery?

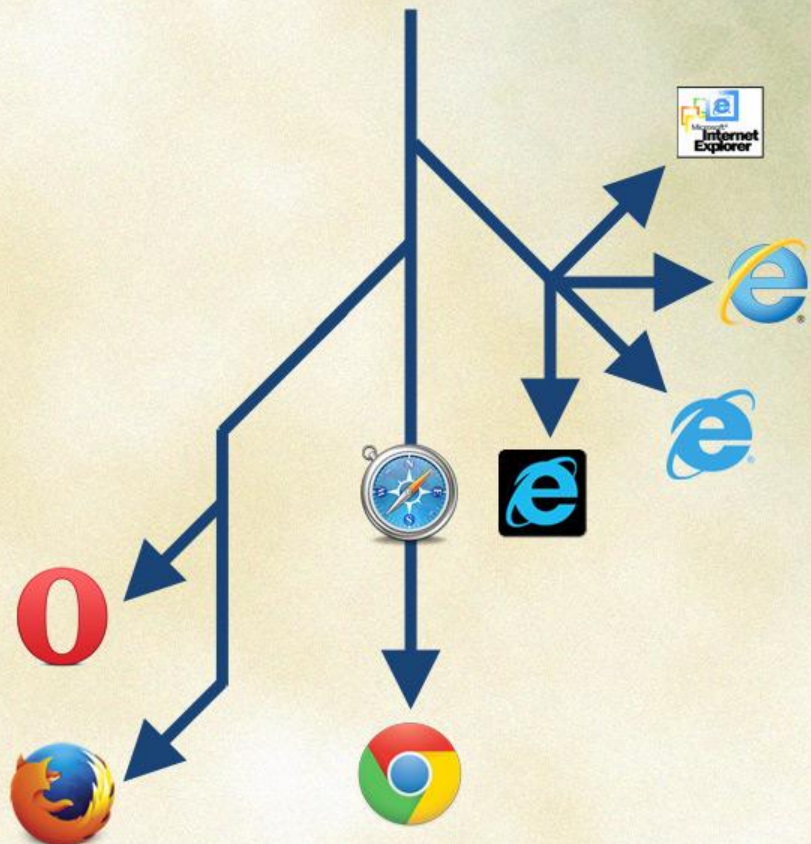






- 1) Haz más cosas escribiendo menos código
- 2) Olvídate los problemas de compatibilidad de los navegadores

JS



JS





# Primeros pasos

Descarga de la librería



<http://jquery.com/>

Versión

CDN

# Versión



- jQuery 1.X
- jQuery 2.X (no tiene soporte para IE8 y anteriores)



# CDN

## Content Delivery Network

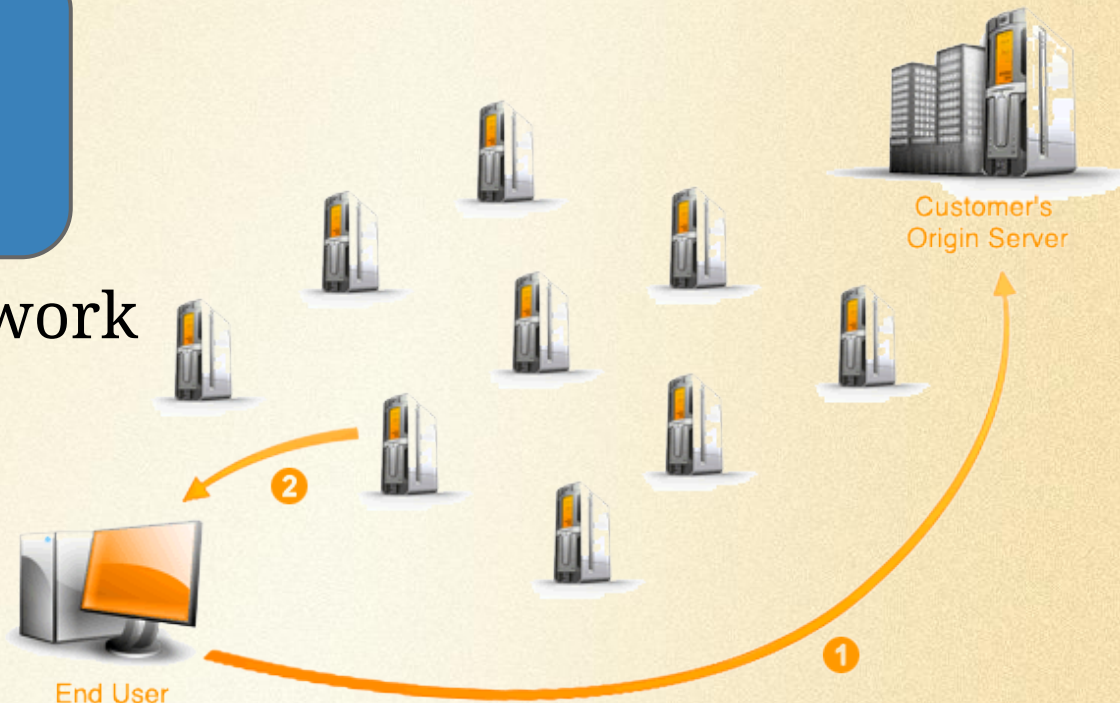
(Red de distribución de contenido)

### Pros:

- Servidor más rápido
- Más cercano
- Posiblemente cacheado

### Contras:

- Necesitas Internet
- CDN puede estar caído



# CDN

¿Diferencias de CDN o hosting local de jQuery?

<http://www.desarrolloweb.com/articulos/jquery-cdn-hosting-local.html>

¿Cuál me recomiendas? **Los dos!!**

<http://www.desarrolloweb.com/articulos/carga-combinada-jquery.html>



# Dónde colocar el JS



</body>

<http://html5boilerplate.com/>

# \$ es la función jQuery



La función jQuery es el "motor" para poner en marcha las funcionalidades de la librería.

Sirve básicamente:

- Seleccionar elementos del navegador y **extenderlos** con las funcionalidades de jQuery
- Invocar funcionalidades concretas



# "Document ready"



Ejecutar acciones cuando el navegador está listo.

```
$ (document) .ready ()
```

# "Document ready"



Ejecutar acciones cuando el navegador está listo.

```
$ (document) .ready ( ) ;
```

No puedo ejecutar instrucciones que alteren el DOM antes del navegador haberlo construido.



# "Document ready"



Ejecutar acciones cuando el navegador está listo.

```
$ (document) .ready ( ) ;
```

\$() sirve para "Seleccionar elementos del navegador y **extenderlos** con las funcionalidades de jQuery"

# "Document ready"



Ejecutar acciones cuando el navegador está listo.

```
$(document).ready(function() {  
    //código a ejecutar  
    //cuando el navegador esté listo  
});
```



# "Document ready"



## Optimiza el evento "onload" nativo de Javascript

(El evento "onload" espera a que TODO esté cargado, imágenes, banners, iframes, embebidos, etc. El evento "ready" de jQuery requiere que todo eso se haya cargado por completo)

# "Document ready"



```
$(document).ready(function() {  
    //código a ejecutar  
    //cuando el navegador esté listo  
});
```

Usando una función anónima

```
$(document).ready(iniciar);  
  
function iniciar() {  
    //código a ejecutar  
    //cuando el navegador esté listo  
}
```



# "Document ready"



```
$(document).ready(function(){  
    //código a ejecutar  
    //cuando el navegador esté listo  
});
```

Shortcut de `on("ready", ...)`

```
$(document).on("ready", function(){  
    //Estoy listo  
});
```

Evento ready definido con el método `on()`, sin shortcut

```
$(function(){  
    //Estoy listo  
});
```

Alternativa al document ready con función "callback"

# Extender objetos



La mayoría de las veces usaremos `$()` para crear objetos jQuery con elementos del DOM.

## 1) Usando un objeto nativo

```
$(document).ready(...)  
$(document.formulario.elemento)
```



# Extender objetos



La mayoría de las veces usaremos `$()` para crear objetos jQuery con elementos del DOM.

## 2) Usando un selector

```
$("#idelemento")  
$(".classelementos")
```

# Extender objetos



La mayoría de las veces usaremos `$()` para crear objetos jQuery con elementos del DOM.

## 3) Usando un HTML

```
$("<div>")  
$("<article><h1>Titulo</h1><p>Cualquier cosa</p></article>")
```



# Recorridos each



En un objeto jQuery puedo tener uno o varios elementos seleccionados. Si necesito recorrellos uno a uno, puedo usar [.each\(\)](#)



**Ojo:** No siempre es necesario recorrer con `.each()` porque los métodos jQuery ya hacen el recorrido implícitamente.

# Recorridos each



En un objeto jQuery puedo tener uno o varios elementos seleccionados. Si necesito recorrerlos uno a uno, puedo usar [.each\(\)](#)

```
$("#p").each(function(id, elem){  
    //id es el número de iteración, comenzando por 0  
    //elem es el elemento actual, sin extender con jQuery  
});
```



# Palabra this



Tiene una referencia al elemento actual.

```
$("p").each(function(id, elem){  
    //this tiene una referencia actual sobre el que estoy en esta iteración  
    //sin extender con jQuery  
});
```

Encontraremos que a veces se refiere a "this" con el término "contexto".

# Devolver false



Podemos devolver false en cualquier iteración para salirnos del bucle .each().

```
$("#p").each(function(id, elem) {  
    if(quiero_salir){  
        return false;  
    }  
});
```



# Gracias!!



Manual de jQuery en DesarrolloWeb.com

<http://www.desarrolloweb.com/manuales/manual-jquery.html>

Curso de jQuery en EscuelaIT

<http://escuela.it/cursos/jquery/>