

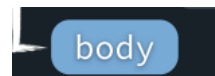
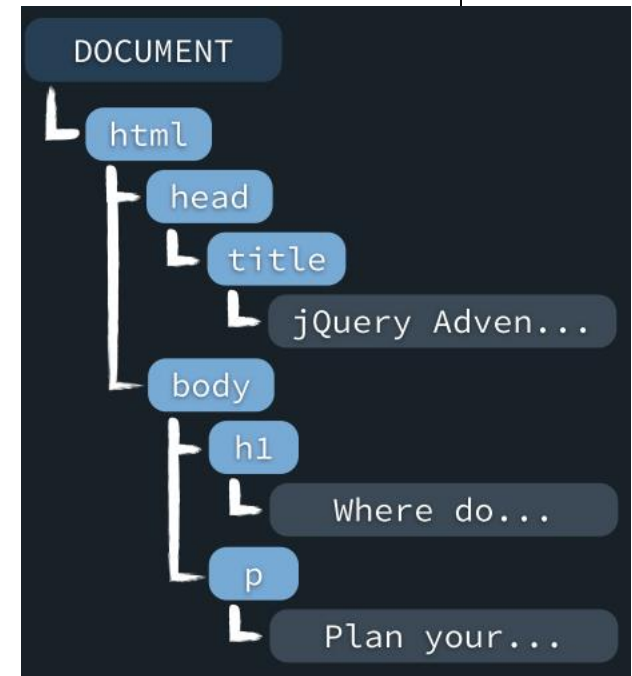
jQuery

Navegacion DOM

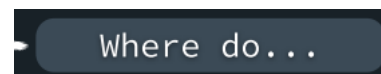




```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Adventures</title>
</head>
<body>
  <h1>Where do you want to go?</h1>
  <p>Plan your next adventure.</p>
</body>
</html>
```



DOM element



Texto



Buscar en el DOM

<body>

<h1>Where do you want to go?</h1>

<p>Plan your next adventure.</p>

- Si ejecuto \$("h1").text();
→ Where do you want to go?





Cambiar elementos

- Si ejecuto `$("h1").text("Where to?");`



Seleccionar más de un elemento



`<h1>Where do you want to go?</h1>`

`<h2>Travel Destinations</h2>`

`<p>Plan your next adventure.</p>`

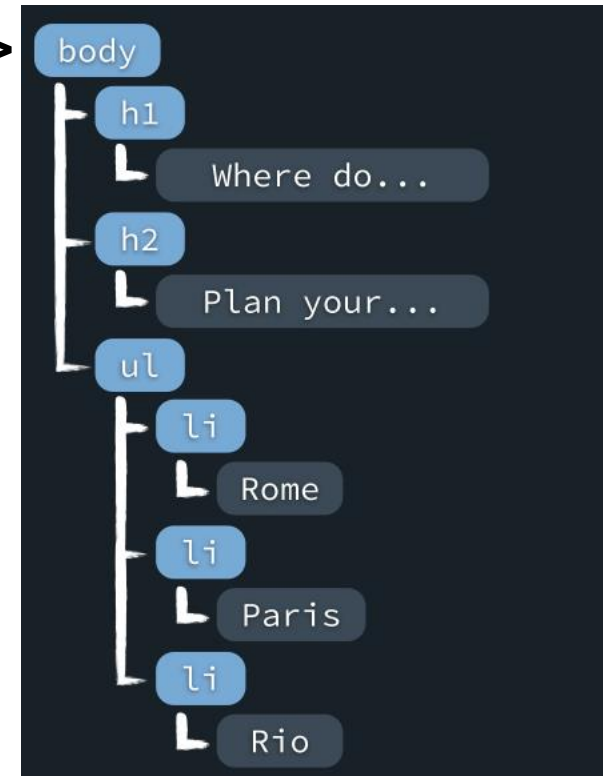
`<ul id="destinations">`

`Rome`

`Paris`

`<li class='promo'>Rio`

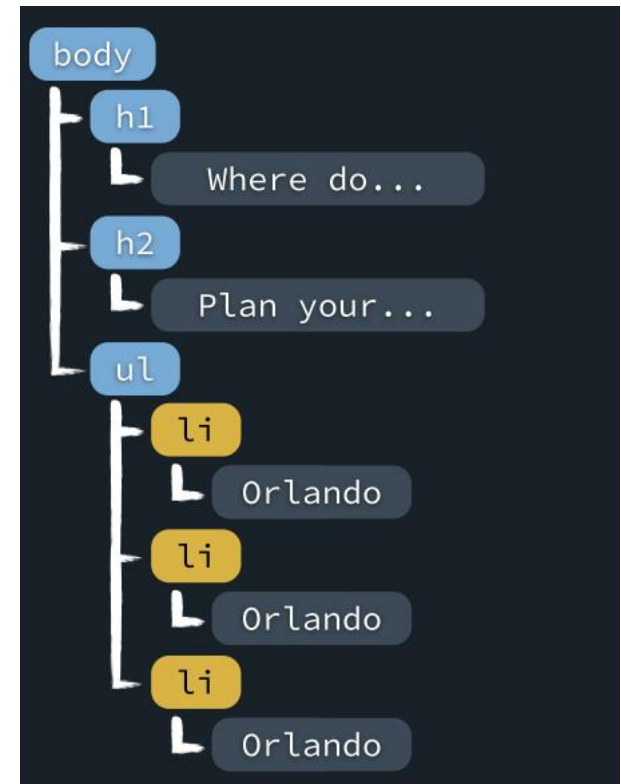
``



Seleccionar más de un elemento



- Si ejecuto
`$("li").text("Orlando");`





`<h1>Where do you want to go?</h1>`

`<h2>Travel Destinations</h2>`

`<p>Plan your next adventure.</p>`

`<ul id="destinations">`

`Rome`

`Paris`

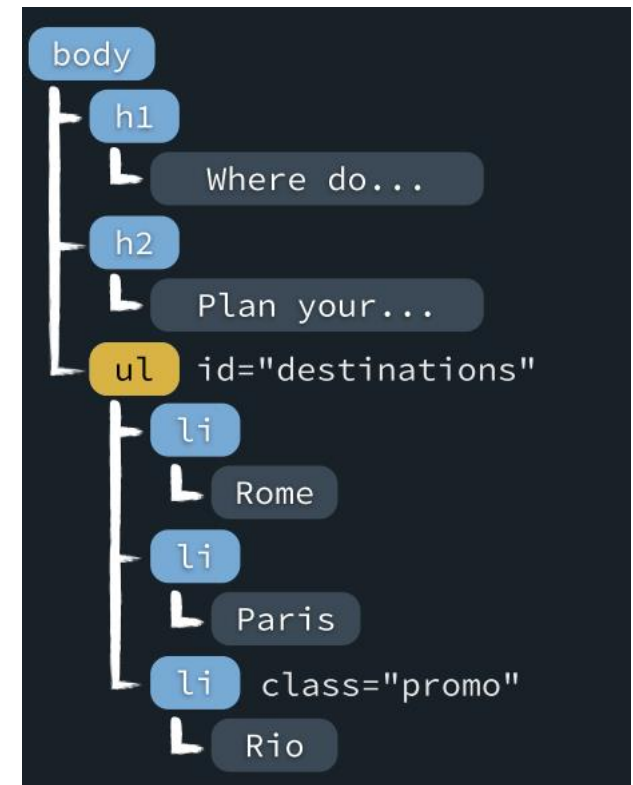
`<li class='promo'>Rio`

``



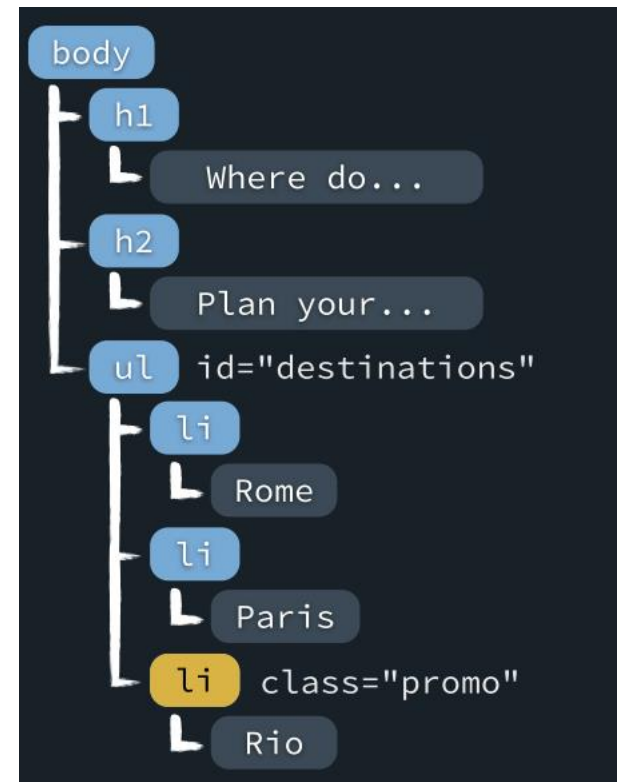


- Si ejecuto
`$("#destinations");`





- Si ejecuto
`$(".promo");`





Descendentes

`<h1>Where do you want to go?</h1>`

`<h2>Travel Destinations</h2>`

`<p>Plan your next adventure.</p>`

`<ul id="destinations">`

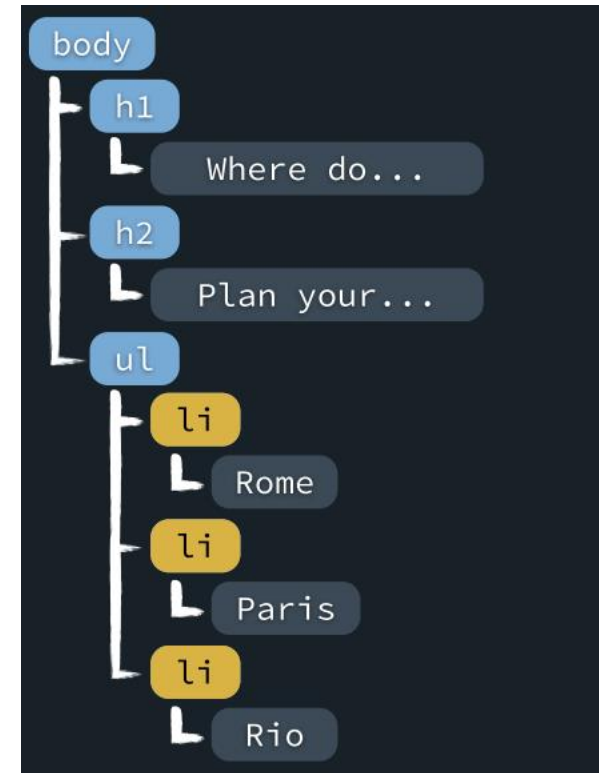
`Rome`

`Paris`

`<li class='promo'>Rio`

``

- Si selecciono `$("#destinations li");`



Seleccionar descendientes directos



- Si selecciono `$("#destinations li");`

→ incorrecto

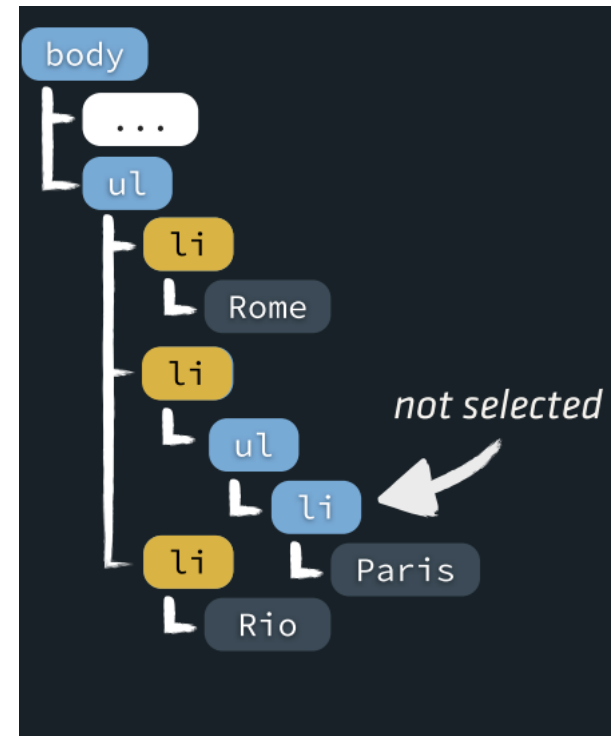


Seleccionar descendientes directos



```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

- `$("#destinations > li");`
- Hijo directo de “destinations”

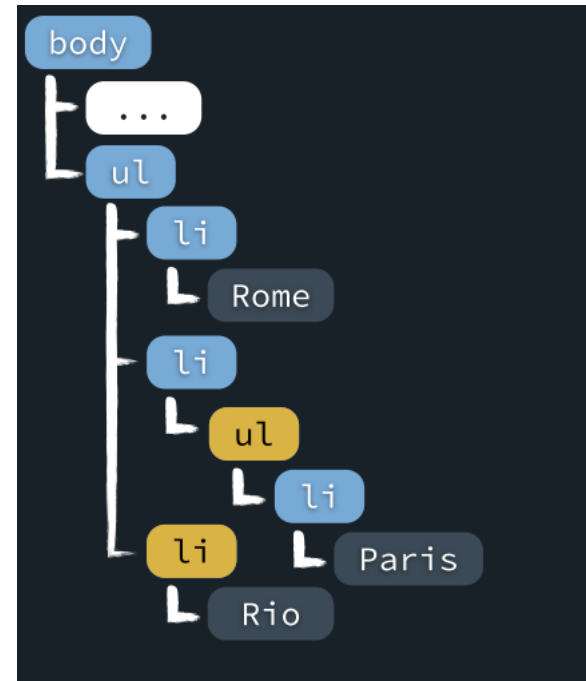


Selecciones múltiples híbridas



```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>
    <ul id="france">
      <li>Paris</li>
    </ul>
  </li>
  <li class='promo'>Rio</li>
</ul>
```

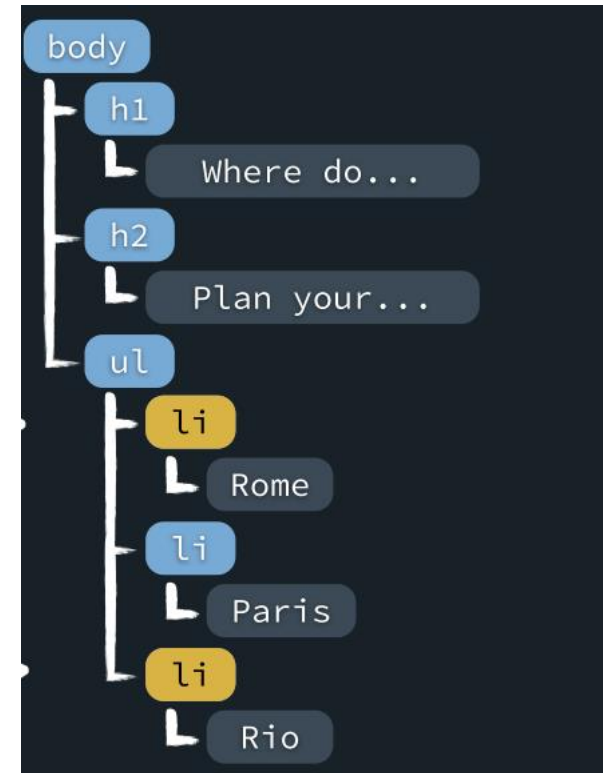
- `$(".promo, #france");`





Selecciones y filtros

- `$("#destinations li:first");`
- `$("#destinations li:last");`





Selecciones y filtros

- `$("#destinations li:odd");`
- `$("#destinations li:even");`



Anterior, precedente, primero, ultimo



```
<h1>Where do you want to go?</h1>
```

```
<h2>Travel Destinations</h2>
```

```
<p>Plan your next adventure.</p>
```

```
<ul id="destinations">
```

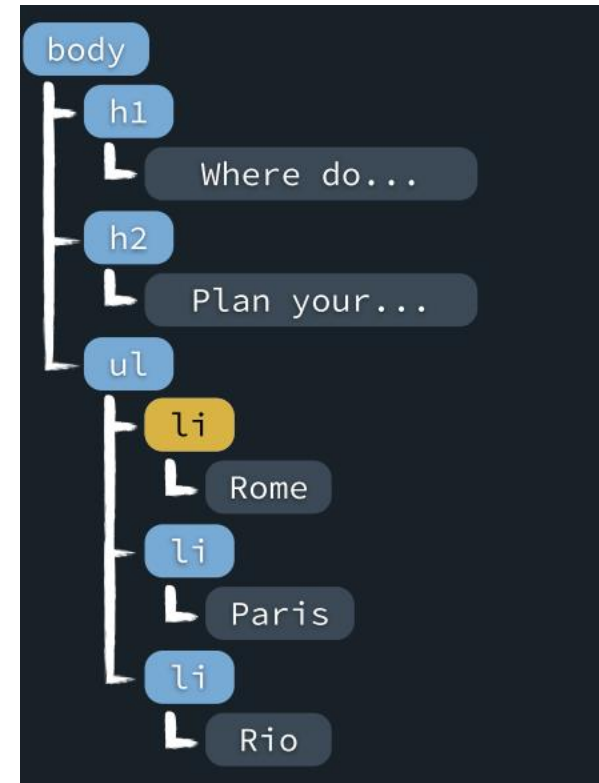
```
  <li>Rome</li>
```

```
  <li>Paris</li>
```

```
  <li class='promo'>Rio</li>
```

```
</ul>
```

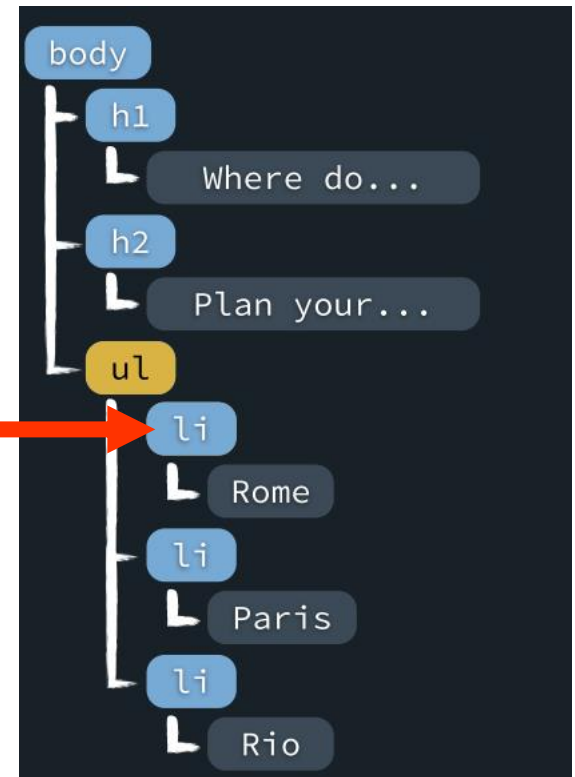
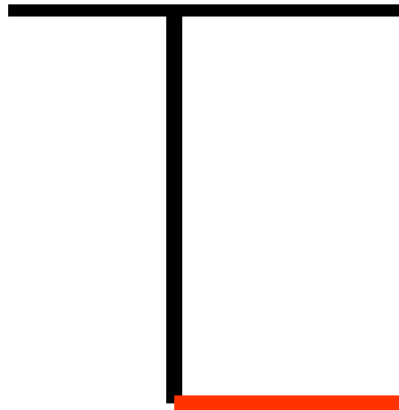
- `$("li:first");`
- `$("li").first();`





Seleccionar padre

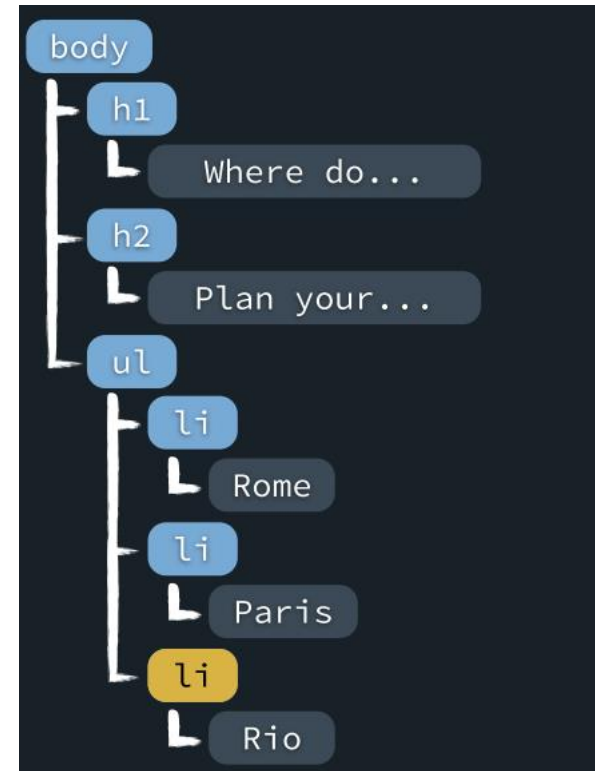
- `$("li").first().parent();`



Anterior, precedente, primero, ultimo



- `$("li:last");`
- `$("li").last();`





Elemento central?

`<h1>Where do you want to go?</h1>`

`<h2>Travel Destinations</h2>`

`<p>Plan your next adventure.</p>`

`<ul id="destinations">`

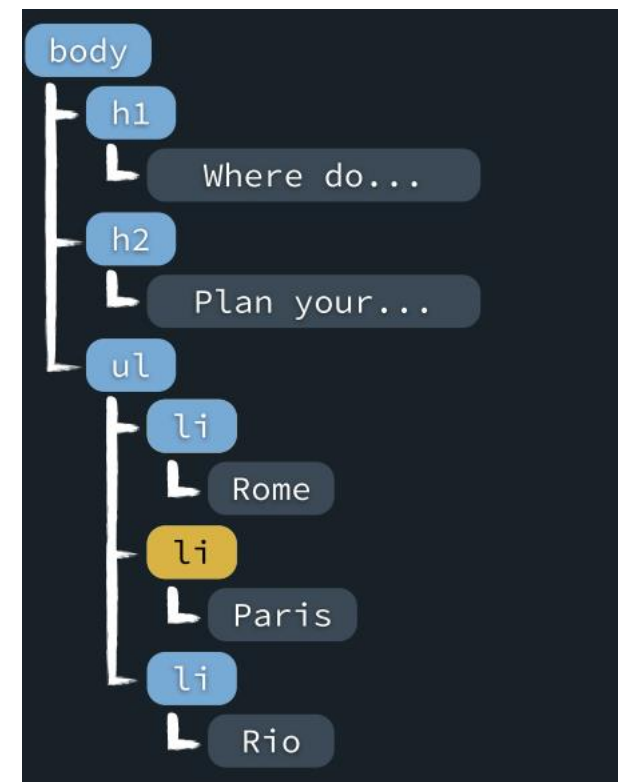
`Rome`

`Paris`

`<li class='promo'>Rio`

``

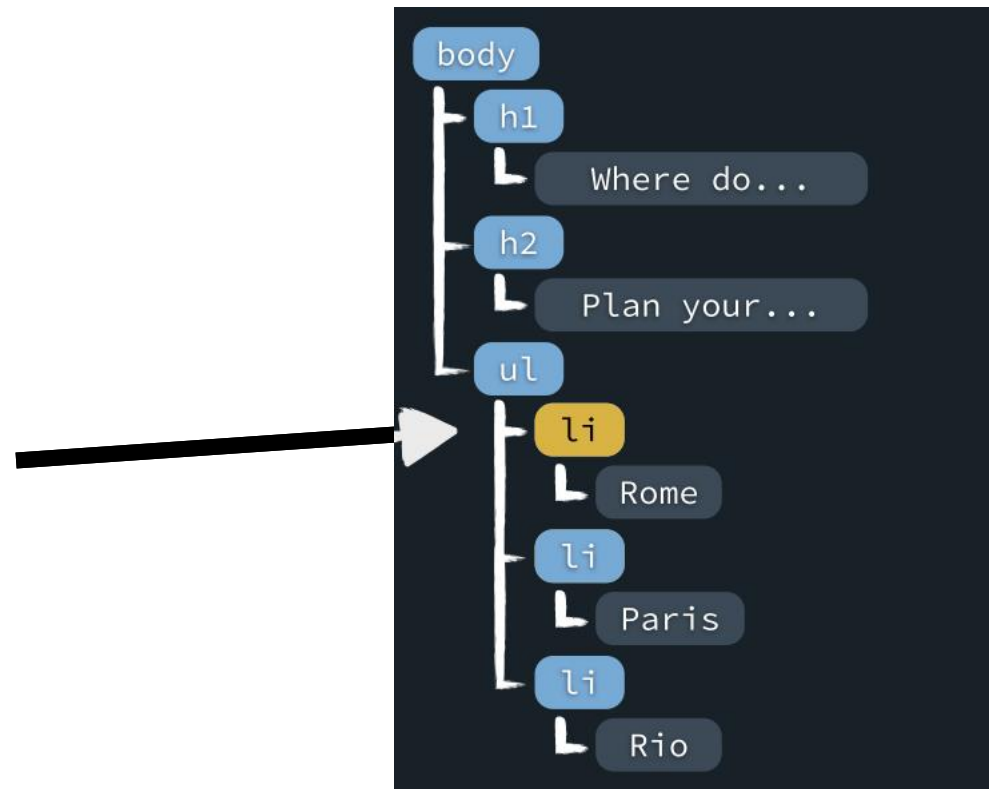
- `$("li").first().next();`



Anterior, precedente, primero, ultimo



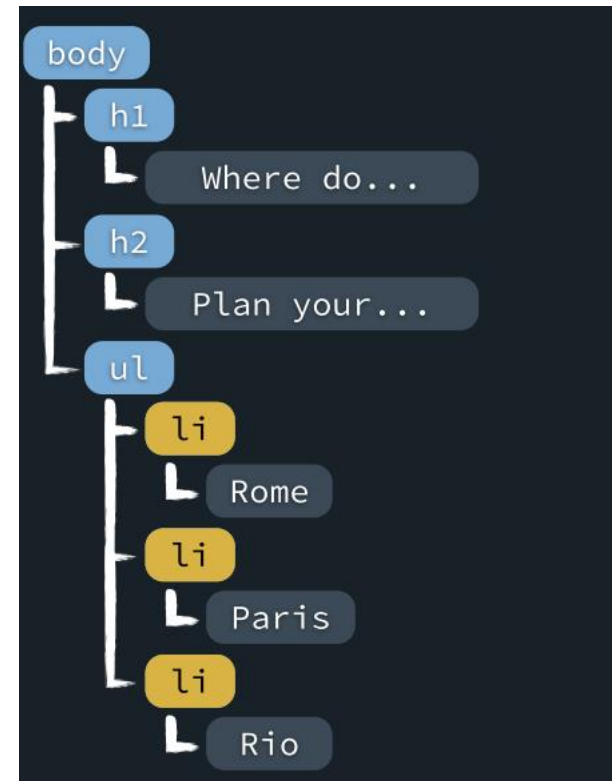
- `$("li").first().next().prev();`





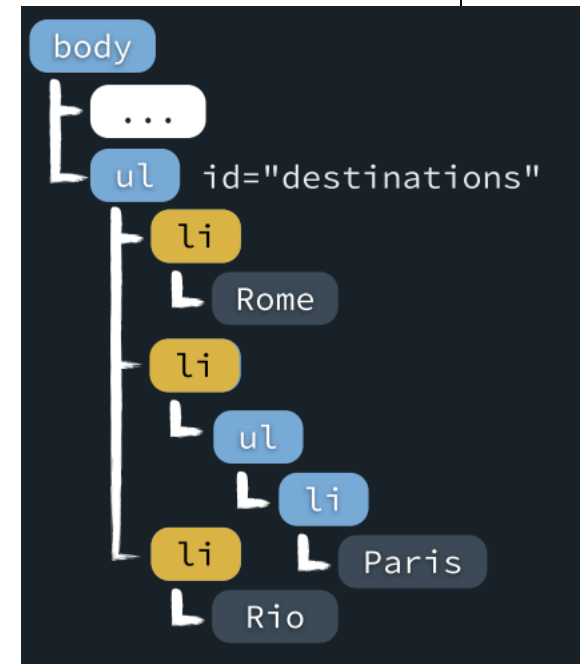
```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <li>Paris</li>
  <li class='promo'>Rio</li>
</ul>
```

- `$("#destinations li");`
O equivalente
- `$("#destinations").find("li");`



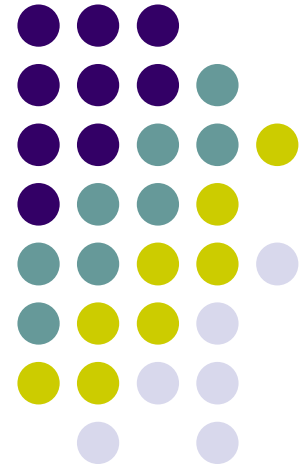


```
<h1>Where do you want to go?</h1>
<h2>Travel Destinations</h2>
<p>Plan your next adventure.</p>
<ul id="destinations">
  <li>Rome</li>
  <ul id="france">
    <li>Paris</li>
  </ul>
  <li class='promo'>Rio</li>
</ul>
```



- `$("#destinations").children("li");`
Children() se distingue de find() porque selecciona solo los hijos directos

Añadir, eliminar del DOM



Formas de añadir un nodo al DOM

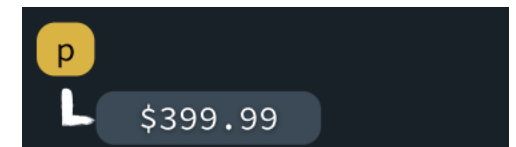


- Queremos que al pulsar el boton, desaparezca el boton y aparezca el precio.

```
$(document).ready(function() {  
  // Crear un nodo <p> con el precio  
  ....¿ a) b) o c) ?  
});
```

- a) `var price = "From $399.99";`
- b) `var price = "<p>From $399.99</p>";`
- c) `var price = $('<p>From $399.99</p>');`

c->crea un nuevo nodo pero no lo añade al DOM



Formas de añadir un nodo al DOM



```
var price = $('<p>From $399.99</p>');  
$(document).ready(function() {  
});
```

```
.append(<element>)  
.prepend(<element>)  
.after(<element>)  
.before(<element>)
```



Before

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').before(price);  
});
```

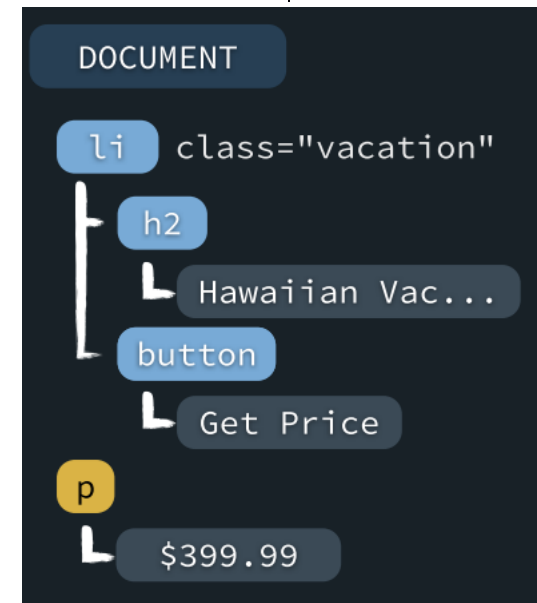


MAL!!! pone el nodo ANTES de “vacation”



After

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').after(price);  
});
```

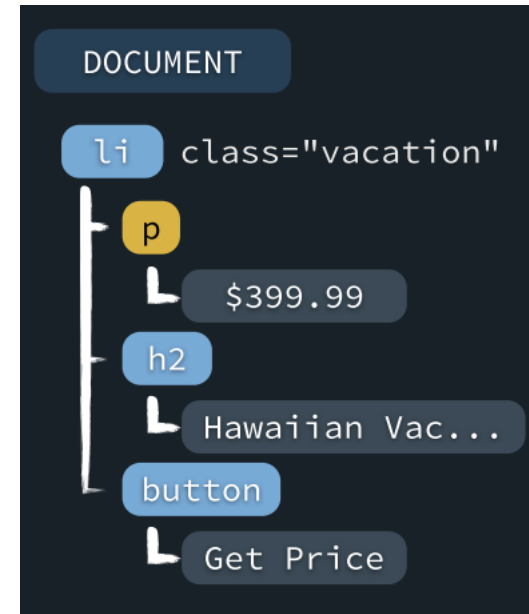


MAL!!! pone el nodo DESPUES de “vacation”



Prepend

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').prepend(price);  
});
```

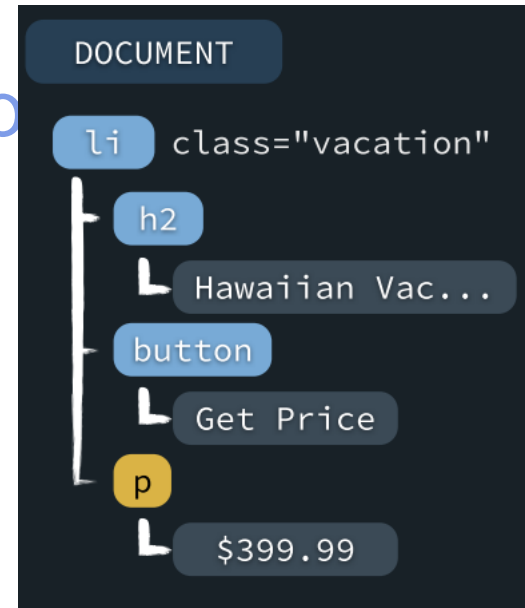


MAL!!! pone el nodo al PRINCIPIO de la lista de “vacation”



Append

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>'  
    $('<p>From $399.99</p>'.vacation').append(price);  
});
```

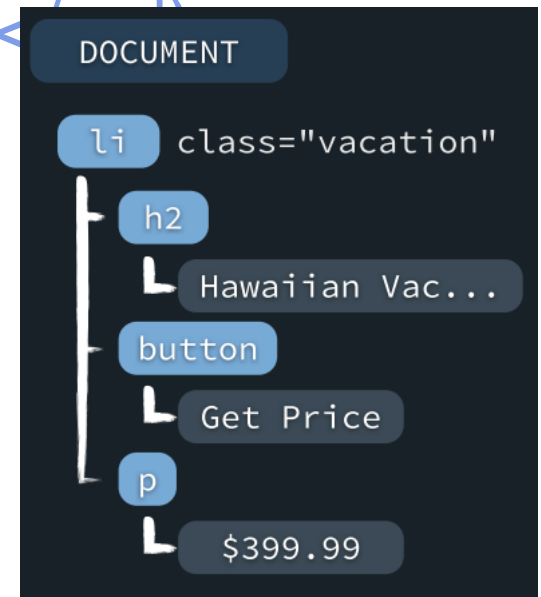


CORRECTO!!! pone el nodo al FINAL de la lista de “vacation”



Remove

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').append(price);  
    $('button').remove();  
});
```



Añade el precio y elimina el `<button>` desde el DOM (remplazar)



.appendTo()

```
$(document).ready(function() {  
    var price = $('<p>From $399.99</p>');  
    $('.vacation').append(price);  
    $('button').remove();  
});
```

`$('.vacation').append(price);`

Es equivalente a

`price.appendTo($('.vacation'));`

Alternativas:

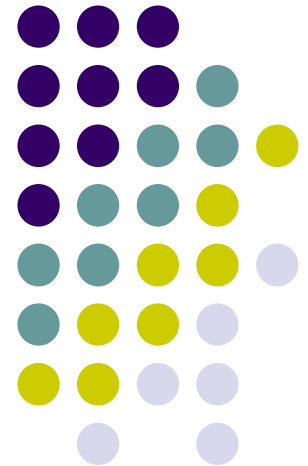
`.appendTo(<element>)`

`.prependTo(<element>)`

`.insertAfter(<element>)`

`.insertBefore(<element>)`

Manipular funciones





Funciones y eventos

Recordamos que:

```
$(document).ready(<event handler function>);
```

- El método “ready” recibe un argumento de función como parámetro

```
function() {  
    // executing the function runs the code  
}
```

- La función ejecuta un código

```
$(document).ready(function() {  
    // this function runs when the DOM is ready  
});
```

- Pasamos la funcion como evento asociado al DOM



Código no obstructivo

```
$(document).ready(function() {
```

```
...
```

```
});
```

- Se ejecuta cuando el DOM se ha cargado

```
$('button').on('click', function() {
```

```
});
```

- Espera un click y ejecuta la función

```
$(document).ready(function() {
```

```
    $('button').on('click', function() {
```

```
    });
```

```
});
```

- Cuando el DOM se ha cargado espera un click del mouse

Codigo no-obstrutivo para reemplazar elementos



```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $('button').remove();  
    });  
});
```



Uso de transversing

- Supongamos que tenemos éste sitio web

A screenshot of a website titled "Vacation Packages" with the subtitle "jQuery Travels". The website features three dark gray cards, each representing a vacation deal. Each card has a title, a section for comments, and a green "GET PRICE" button. The first card is for "Hawaiian Vacation" with two comments: "Amazing Deal!" and "Can't wait to take this trip!". The second card is for "Orlando" with one comment: "There aren't any comments on this deal yet." The third card is for "Visit Japan" with one comment: "Never been, but can't wait!".

Vacation Packages
jQuery Travels

Hawaiian Vacation

Comments on this deal:
"Amazing Deal!"
"Can't wait to take this trip!"

GET PRICE

Orlando

Comments on this deal:
There aren't any comments on this deal yet.

GET PRICE

Visit Japan

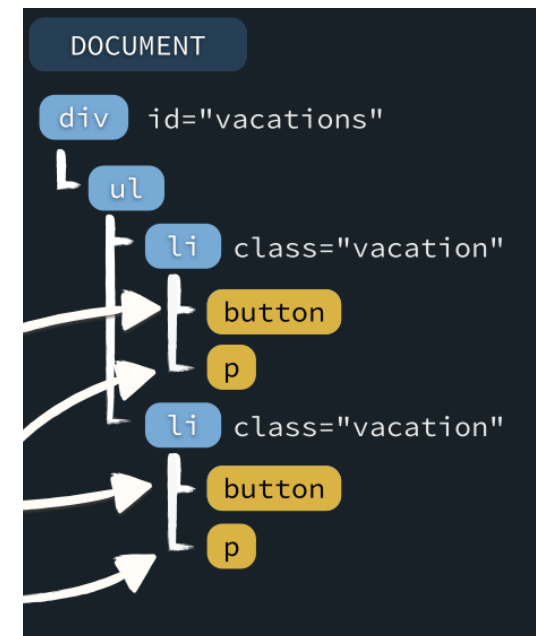
Comments on this deal:
"Never been, but can't wait!"

GET PRICE



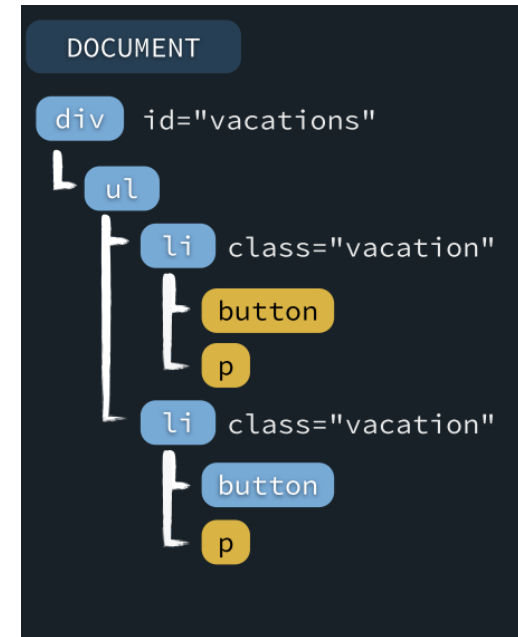
```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $('button').remove();  
    });  
});
```

- MAL!! cada boton sera cambiado
- Alternativa usar `$(this).remove();`





```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $('.vacation').append(price);  
        $(this).remove();  
    });  
});
```



- Solo elimina el botón que ha sido clicado



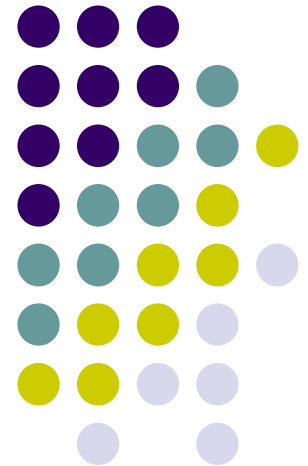
```
$(document).ready(function() {  
    $('button').on('click', function() {  
        var price = $('<p>From $399.99</p>');  
        $(this).remove();  
    });  
});
```

Alternativa:

```
$(this).after(price);
```



Acciones on click





Deslizado

```
<li class="confirmation">
  <button>FLIGHT DETAILS</button>
  ...
  <ul class="ticket">...</ul>
</li>
```

jQuery Object
Methods

- .slideDown()
- .slideUp()
- .slideToggle()

```
$('.confirmation').on('click', 'button', function() {
  $(this).closest('.confirmation').find('.ticket').slideDown();
});
```

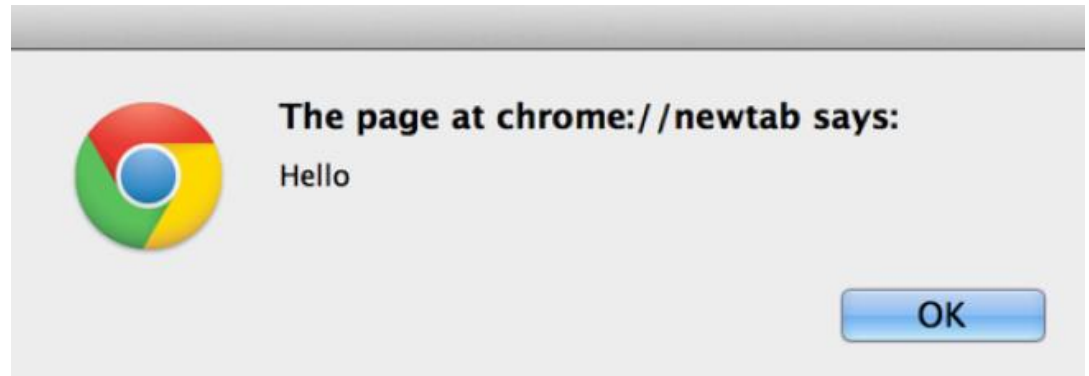
Busca entre los padres

Busca entre los hijos

Alertas

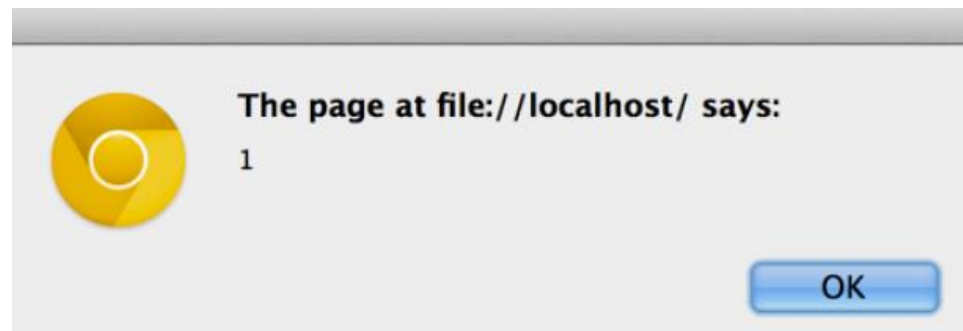


- `alert('Hello');`



- `$('li').length;`

Me dice cuantos elementos (“nodos”) hay en la pagina





Interacciones con teclado

```
<div class="vacation" data-price='399.99'>  
  <h3>Hawaiian Vacation</h3>  
  <p>$399.99 per ticket</p>  
  <p>  
    Tickets:  
    <input type='number' class='quantity' value='1' />  
  </p>  
</div>  
<p>Total Price: $<span id='total'>399.99</span></p>
```



- Queremos que cuando cambiamos los tiquets, actualice el precio



```
$(document).ready(function() {  
    $('.vacation').on('?', '.quantity', function() {  
        // a) Recuperar el precio unitario  
        // b) Recuperar la cantidad que se ha entrado  
        // c) Definir el precio = cantidad * precio  
    });  
});
```

Opciones: **keypress, keydown, keyup**

<http://api.jquery.com/category/events/keyboard-events/>



a) `var price = +$(this).closest('.vacation').data('price');`

El “+” permite convertir el string en un entero.

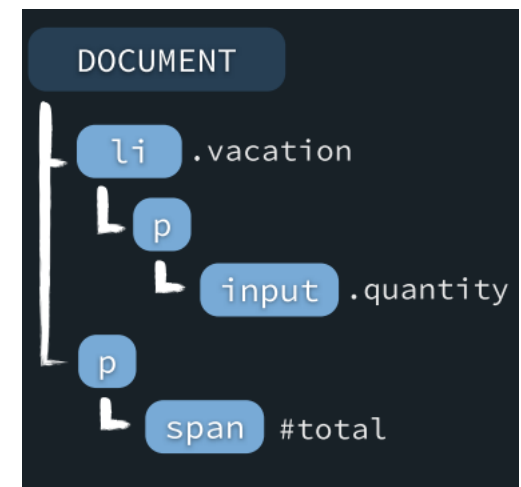
b) `var quantity = +$(this).val();`

- Convierte el valor del string en entero

c) `$('#total').text(price * quantity);`

- Paso como parametro el valor calculado

```
$(document).ready(function() {  
  $('.vacation').on('keyup', '.quantity', function() {  
    var price = +$(this).closest('.vacation').data('price');  
    var quantity = +$(this).val();  
    $('#total').text(price * quantity);  
  });  
});
```



Resaltar un recuadro usando CSS



application.js

```
$(document).ready(function() {  
    $('#vacations').on('click', '.vacation', function() {  
        $(this).css({'background-color': '#563',  
                    'border-color': '1px solid #967'});  
        $(this).find('.price').show();  
    });  
});
```



Se transforma en:

```
$(document).ready(function() {  
    $('#vacations').on('click', '.vacation', function() {  
        $(this).addClass('highlighted');  
    });  
});
```

application.css

```
.highlighted {  
    background-color: #563  
    border-color: 1px solid #967;  
}  
.highlighted .price {  
    display: block;  
}
```



- Problema: de la forma anterior añadimos un elemento, pero como hacemos para quitarlo con el evento?

```
$(document).ready(function() {  
  $('#vacations').on('click', '.vacation', function() {  
    $(this).addClass('highlighted');  
  });  
});
```

Lo remplazamos por

```
$(this).toggleClass('highlighted');
```



Animaciones / movimientos

```
$(document).ready(function() {  
    $('#vacations').on('click', '.vacation', function() {  
        $(this).toggleClass('highlighted');  
    });  
});
```

MAL: `$(this).css({'top': '-10px'});`

El box hace un “salto” de 10 pixeles

BIEN: `$(this).animate({'top': '-10px'});`

El box desliza un “salto” de 10 pixeles



- Opciones de “animate”

```
$(this).animate({'top': '-10px'});
```

```
$(this).animate({'top': '-10px'}, 400);
```

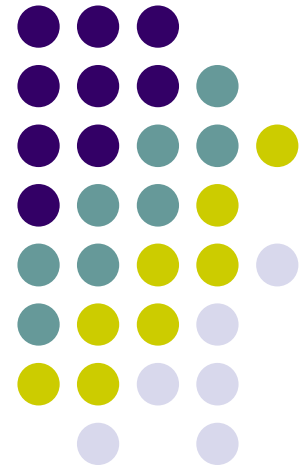
```
$(this).animate({'top': '-10px'}, 'fast');
```

```
$(this).animate({'top': '-10px'}, 200);
```

```
$(this).animate({'top': '-10px'}, 'slow');
```

```
$(this).animate({'top': '-10px'}, 600);
```

Ajax con JQuery



Ajax



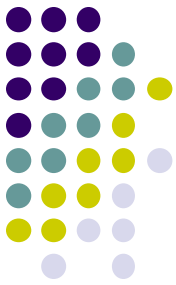
- AJAX permite cargar datos en background y visualizarlos en la pagina web, sin recargar la página entera.
- jQuery proporciona varios métodos para gestionar AJAX.
 - Se puede pedir texto, HTML, objetos XML, o JSON desde un servidor usando Get and Post –
 - Se pueden cargar datos externos directamente en los elementos HTML elegidos de la pagina web.

Load

Demo_test.txt

<h2>jQuery and AJAX is FUN!!!</h2>

<p id="p1">This is some text p>



- El metodo load() carga los datos desde un servidor y pone el resultado en el elemento DOM elegido.
- `$(selector).load(URL,data,callback);`

```
<!DOCTYPE html> <html> <head>
```

```
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
</script>
```

```
<script>
```

```
$(document).ready(function(){
    $("button").click(function(){
        $("#div1").load("demo_test.txt");
    });
});
```

```
});</script>
```

```
</head> <body>
```

```
<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>
```

```
<button>Get External Content</button>
```

```
</body></html>
```

Let jQuery AJAX Change This Text

Get External Content



jQuery and AJAX is FUN!

This is some text in a paragraph.

Get External Content



Parámetros

- Existen parámetros de callback que se ejecutan cuando load() se termina:
- responseTxt – contiene el resultado (o el contenido) cuando el método ha sido ejecutado con suceso.
- statusTxt – devuelve el estado de la llamada (error)
- xhr – contiene el objeto XMLHttpRequest



Ejemplo con parámetros

- ```
$("#button").click(function(){
 $("#div1").load("demo_test.txt",function(responseT
xt,statusTxt,xhr){
 if(statusTxt=="success")
 alert("External content loaded successfully!");
 if(statusTxt=="error")
 alert("Error: "+xhr.status+": "+xhr.statusText);
 });
});
```



# GET vs. POST

- **HTTP Request: GET vs. POST**

Para pedir-enviar datos entre servidor y cliente se usan: GET y POST. Aun que puedan ser usados con la misma función:

- **GET** – Pide datos desde una fuente.
- **POST** – Envía datos procesados a una fuente específica.

→ **Nota:** el método GET puede devolver datos en cache.



# GET

- `$.get(URL, callback);`

```
<!DOCTYPE html><html><head>
```

```
<script
```

```
 src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
```

```
</script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
 $("button").click(function(){
```

```
 $.get("demo_test.asp", function(data, status){
```

```
 alert("Data: " + data + "\nStatus: " + status);
```

```
 }); }); });</script>
```

```
</head> <body>
```

```
<button>Send an HTTP GET request to a page and get the result
 back</button>
```

```
</body> </html>
```

Demo\_test.asp

```
<%
```

```
 response.write("This is some text.")
```

```
%>
```





# POST

- `$.post(URL,data,callback);`

.....

```
<script>
$(document).ready(function(){
 $("button").click(function(){
 $.post("demo_test_post.asp",
 {
 name:"Donald Duck",
 city:"Duckburg"
 },
 function(data,status){
 alert("Data: " + data + "\nStatus: " + status);
 }); }); });
</script>
```

```
</head><body>
```

```
<button>Send an HTTP POST request to a page and get the result back</button>
```

```
</body></html>
```

```
demo_test_post.asp
<%
dim fname,city
fname=Request.Form("name")
city=Request.Form("city")
Response.Write("Dear " & fname & ". ")
Response.Write("Hope you live " & city & ".")
%>
```

# Compatibilidad entre framework JS



- jQuery usa el símbolo \$ para recuperar informaciones desde el DOM.
- **Que pasa si otros frameworks JavaScript usan el mismo símbolo?**
- En jQuery existe un método llamado noConflict().



```
var jq = $.noConflict();
jq(document).ready(function(){
 jq("button").click(function(){
 jq("p").text("jQuery is still working!");
 });
});
```

- O uso `jQuery(document).ready(function){`
- El método `noConflict()` devuelve una referencia a jQuery, que se puede salvar en una variable y usarla en el código.



- Si existe un bloque de código jQuery que usa el shortcut \$ y no se quiere cambiar todo, es posible pasar el símbolo \$ como parámetro al método ready. Eso permite acceder al jQuery usando \$, a dentro de esa función – mientras que a fuera se tiene que usar "jQuery":

```
$.noConflict();
jQuery(document).ready(function($){
 $("button").click(function(){
 $("p").text("jQuery is still working!");
 });
});
```