



# Objetos JavaScript:

Notación array, propiedades dinámicas, for/in y DOM

# Objetos

## ◆ Objeto

- Es una **agregación de variables**, denominadas **propiedades**
- Se suelen construir con el literal de objeto
  - ◆ { propiedad\_1:valor\_1, ....., propiedad\_n:valor\_n }

## ◆ Notación punto

- **objeto.propiedad**
  - ◆ Accede propiedades por su **nombre**

## ◆ Notación array

- **objeto["propiedad"]**
  - ◆ Accede pasando el nombre como un string
    - Que puede estar en una variable

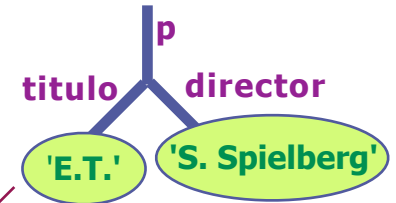
## ◆ Notación punto o array son equivalentes

## ◆ Propiedades inexistentes devuelven undefined

- Pero el operador punto (.) aplicado a **undefined**
  - ◆ provoca **error de ejecución**

## ◆ Los nombres de propiedades de un objeto

- deben ser **todos diferentes**
- tienen la misma **sintaxis que las variables**
  - ◆ a, \_method, \$1, una\_piña, ....



```
var p = {titulo:'E.T.', director:'S. Spielberg'};
```

```
// Referencias a propiedades
```

```
p.titulo           => 'E.T.'
```

```
p.director         => 'S. Spielberg'
```

```
p['titulo']         => 'E.T.'
```

```
p['director']       => 'S. Spielberg'
```

```
// Acceso a través de variables
```

```
var t = 'titulo'; // contiene string 'titulo'
```

```
p[t]               => 'E.T.'
```

```
// propiedades inexistentes
```

```
p.t               => undefined
```

```
p.estreno         => undefined
```

```
p['estreno']       => undefined
```

```
undefined[t]      => Error de ejecución
```

# Objetos anidados: árboles

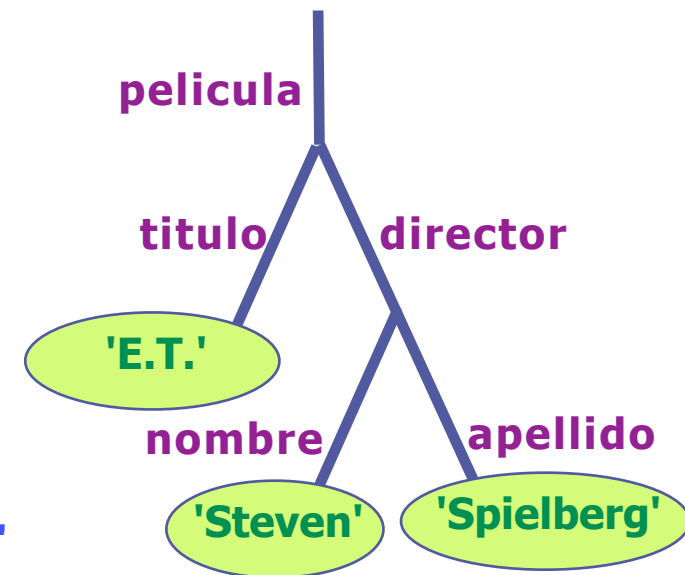
```
var pelicula = {  
  titulo: 'E.T.',  
  director: {  
    nombre: 'Steven',  
    apellido: 'Spielberg'  
  }  
};
```

- ◆ Los objetos pueden **anidarse** entre si
  - Los objetos anidados representan **árboles**

- ◆ La notación punto o array puede **encadenarse**
  - Representando un **camino en el árbol**

- ◆ Las siguientes expresiones se evalúan así:

- `pelicula.director.nombre`  $\Rightarrow$  'Steven'
- `pelicula['director']['nombre']`  $\Rightarrow$  'Steven'
- `pelicula['director'].apellido`  $\Rightarrow$  'Spielberg'
- `pelicula.estreno`  $\Rightarrow$  undefined
- `pelicula.estreno.año`  $\Rightarrow$  **Error\_de\_ejecución**



# Nombres extendidos de propiedades

## ◆ Utilizando **literales de objeto** y **notación array**

- Es posible tener objetos con **nombres** de propiedades **arbitrarios** o **extendidos**

## ◆ La **notación array** puede utilizar

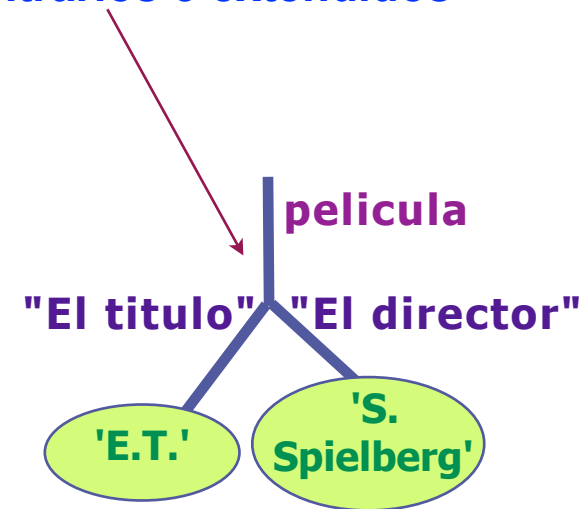
- **strings arbitrarios** como **nombres de propiedades**
  - ♦ por ejemplo, `pelicula["El director"]`, `objeto[""]` o `a["%43"]`

## ◆ El **literal de objeto** permite también

- **strings arbitrarios** como nombres de propiedades
  - ♦ Por ejemplo, `{"El titulo": 'E.T.', "El director": 'S. Spielberg'}`

## ◆ **OJO!** normalmente es conveniente utilizar nombres para **notación punto**

- Strings arbitrarios pueden ser útiles en objetos tipo diccionario o similares



# Propiedades dinámicas

## ◆ Las **propiedades** se pueden **crear** y **destruir**

- Para ello se utilizan 3 sentencias
  - ◆ Asignación de valores
  - ◆ Borrado de propiedades
  - ◆ Comprobar si existe una propiedad

## ◆ Asignar a (**y crear**) propiedades: **x.c = 4**

- **asigna 4** -> si la propiedad **c** **existe**
- **crea c y le asigna 4** -> si la propiedad **c** **no existe**

## ◆ Borrar propiedades:

- **elimina x.c** -> si la propiedad **x.c** **existe**
- **no hace nada** -> si la propiedad **x.c** **no existe**

## ◆ Borrar propiedades:

- devuelve **true** -> si la propiedad **x.c** **existe**
- devuelve **false** -> si la propiedad **x.c** **no existe**

La propiedad ya existe y **solo cambia el valor a 7**

La propiedad no existe y **se crea** con el valor 5

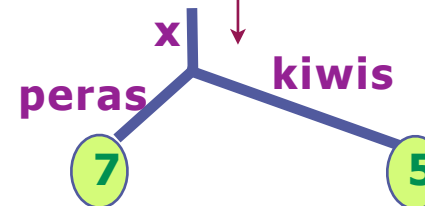
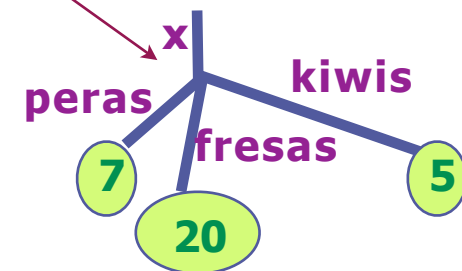
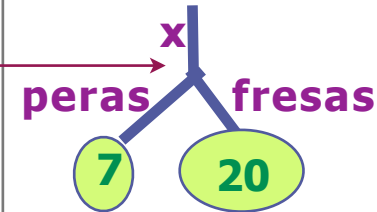
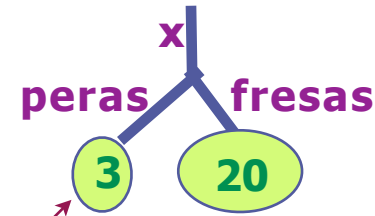
```
var x = { peras:3, fresas:20};
```

```
x.peras = 7;
```

```
x.kiwis = 5;
```

```
delete x.fresas;
```

La propiedad se destruye



# Sentencia for/in

- ◆ La sentencia **for (var p in obj) {..bloque de instrucciones..}**
  - itera en todas las **propiedades del objeto obj**
- ◆ La variable **p** contiene en cada iteración
  - un **string** con el nombre de una **propiedad** diferente de **obj**
    - ◆ El contenido de la propiedad debe ser accedido con la notación array: **obj[p]**

```
<!DOCTYPE html>
<html>
<head>
  <title>For/In</title><meta charset="UTF-8"></head>
<body>

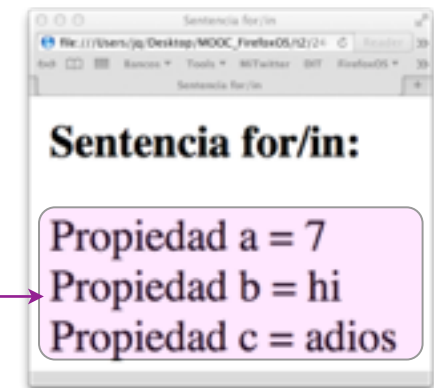
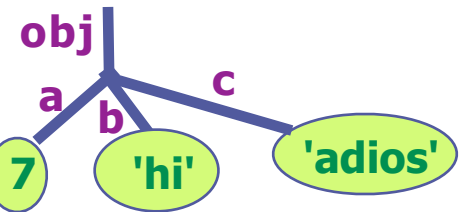
  <h3>Sentencia for/in:</h3>

  <script type="text/javascript">
    var obj = { a:7, b:'hi', c:'adiós'};

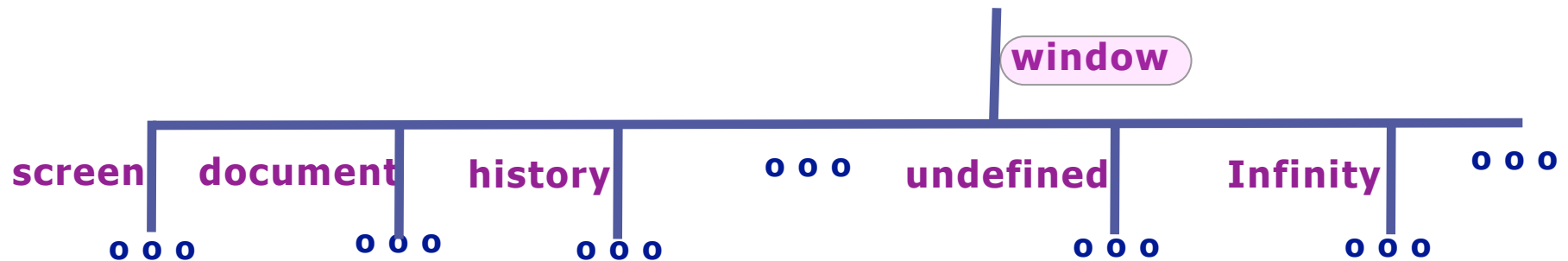
    for (var p in obj) { //cada iteración muestra una propiedad
      document.writeln("Propiedad " + p + " = " + obj[p] + "<br>");
    }
  </script>
</body>
</html>
```

Contiene un string con el nombre de la propiedad.

Contenido de la propiedad accesible con notación array.



# Objeto window o this



- ◆ Objeto **window**: entorno de ejecución de JavaScript
  - El **objeto global window** tiene propiedades con información sobre
    - ◆ Objetos predefinidos de JavaScript, el navegador, el documento HTML, .....
- ◆ **window** se referencia también como **this** (en el entorno global)
  - La **propiedad document** de **window** se referencia como
    - ◆ `window.document`, `this.document` o `document`
- ◆ Documentación: <https://developer.mozilla.org/en-US/docs/Web/API/Window>

# Object inspector

```
<!DOCTYPE html><html><head><title>DOM</title><meta charset="UTF-8">
<script type="text/javascript" src="https://code.jquery.com/jquery-2.1.4.min.js" ></script>

<script type="text/javascript">
$(function() {
  $("#b1").on("click",
    function() {
      var obj = window[$("#prop").val()];
      var tabla = $("#tabla");
      tabla.html("");

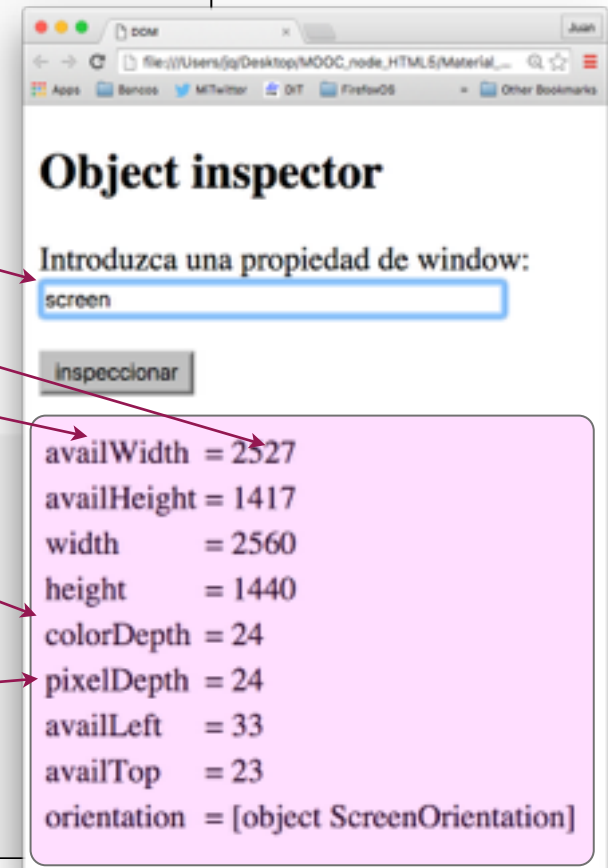
      for (var i in obj) { //cada iteración genera una fila de la tabla
        tabla.append("<tr><td>" + i + "</td><td>" + obj[i] + "</td></tr>");
      }
    }
  );
});
</script>
</head><body>

<h2> Object inspector </h2>

Introduzca una propiedad de window: <br>
<input type="text" size="40" id="prop"><p>
<button id="b1">inspeccionar</button>

  <!-- tabla con propiedades de screen -->
<table id="tabla"></table>

</body></html>
```







# JavaScript: jQuery UI

# Librerías jQuery UI



## ◆ jQuery UI enriquece jQuery

- Es muy eficaz soportando las **interacciones de usuario** mas habituales
  - ◆ Transforma elementos HTML en elementos activos de interacción con el usuario
    - Soporta las interacciones más habituales: arrastrar, depositar, seleccionar, ordenar , ..
  - ◆ La parte de interacción de usuario es la más utilizada
- Soporta también muchas **animaciones y efectos**
- También posee una galería de temas gráficos
  - ◆ Los temas no son muy populares

## ◆ jQuery UI es muy popular (<http://jqueryui.com>)

- Facilita mucho el diseño de la interfaz con el usuario
  - ◆ La versión actual (7-1-15) es **jQuery 1.11**
- Se distribuye con licencia abierta (MIT License)
  - ◆ La primera versión se publico en septiembre de 2007

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

## ◆ Tutorial:

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

# Interacciones de jQuery UI



- ◆ Los patrones de **interacción con el usuario** transforman elementos HTML en
  - **Arrastrables (Draggable)**: Permite arrastrar un elemento en un area
  - **Depositable (Droppable)**: Permite depositar un elemento en otro.
  - **Redimensionable (Resizable)**: Permite redimensionar un elemento.
  - **Seleccionable (Selectable)**: Permite seleccionar un elemento de una lista.
  - **Ordenable (Sortable)**: Permite ordenar una lista de elementos.
- ◆ Los componentes (widgets) construyen elementos de interacción
  - **Acordeón (Accordion)**: Expande un elemento de una lista.
  - **Autocompleción (Autocomplete)**: Completa palabras de una lista.
  - **Botón (Button)**: Da uniformidad a diversos tipos de botones.
  - **Introducción de fechas (Datepicker)**: Despliega un calendario.
  - **Deslizador (Slider)**: Crea un elemento deslizable.
  - Y muchos otros (Menu, Progressbar, Selectmenu, Spinner, Tabs, Tooltip, ..)
- ◆ Además hay efectos, transiciones, temas y otros elementos (menos utilizados)
- ◆ Tutorial: <http://www.desarrolloweb.com/manuales/manual-jqueryui.html>

# Puzzle

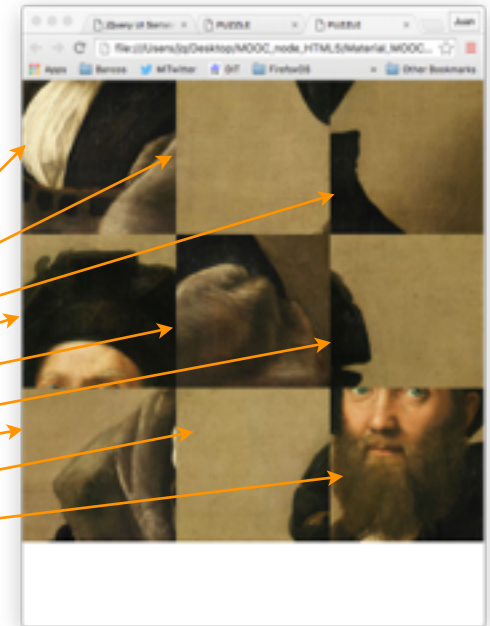
Elimina margen y padding de todo

Elimina estilo de lista

Posición flotante ocupando cada cuadrado 1/3 tercio de la anchura de la ventana, forma que se coloquen en 3 filas de 3 cuadrados.

Cada trozo de la imagen ocupa todo el espacio asignado al elemento de la lista

```
<ul id="sortable">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
```



- ◆ El ejemplo muestra un puzzle de 9 trozos de imagen
  - los trozos son cuadrados para que puedan intercambiarse
    - ♦ Están colocados en 3 líneas de 3 trozos
- ◆ La imagen se define en HTML como una lista de nueve elementos <li>
  - Cada elemento <li> contiene un trozo de la imagen
    - ♦ El puzzle se resuelve cambiando el orden de los trozos en la lista
- ◆ CSS coloca los elementos flotando hacia la izquierda
  - Cada elemento ocupa 1/3 del ancho de la ventana
    - ♦ Se colocarán en una matriz de 3 x 3, de izquierda a derecha

# Puzzle

Carga de jQuery  
y jQuery UI

```
<!doctype html>
<html> <head>
<meta charset="utf-8"> <title>PUZZLE</title>
<script src="jquery-2.1.4.js"> </script>
<script src="jquery-ui.min.js"> </script>
<style>
```

```
*          { margin:0; padding:0; }
#sortable  { list-style-type: none; }
#sortable li { float: left; width: 33.3%; }
#sortable li img {
  vertical-align:bottom;
  width: 100%;
  height: auto;
}
</style>
```

Estilo anterior compactado

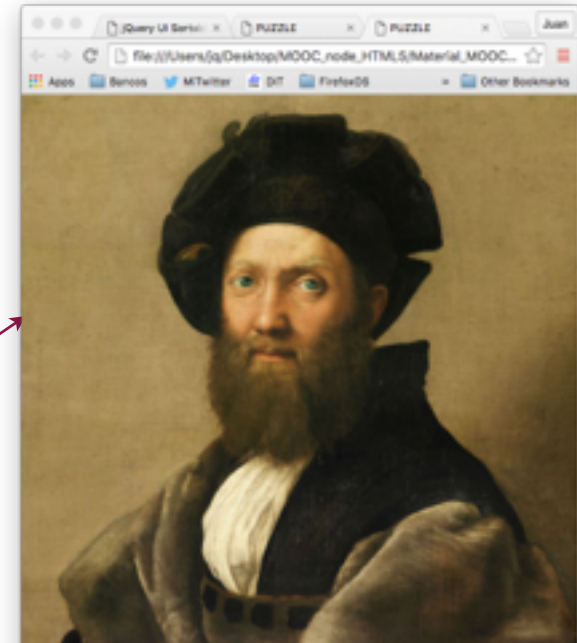
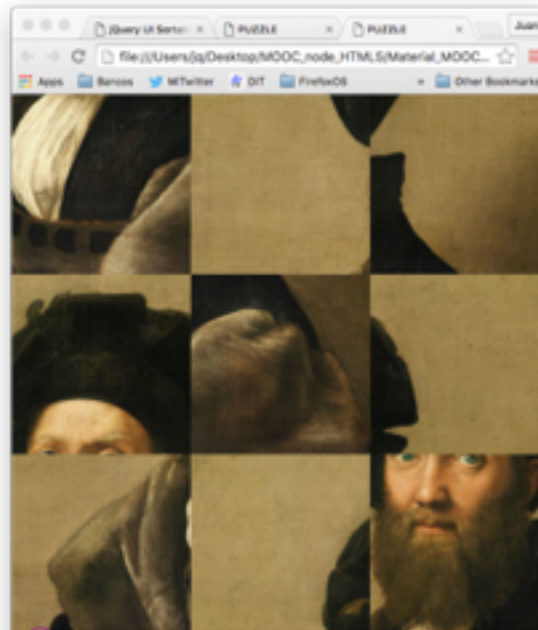
```
<script>
$(function() {
  $( "#sortable" ).sortable();
  $( "#sortable" ).disableSelection();
});
</script>
```

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

```
<ul id="sortable">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>
</body></html>
```

- ◆ Ordenación de listas es un patrón de interacción habitual
  - Los elementos se arrastran con el ratón para cambiar de orden
    - ♦ Esto permitirá mover los cuadrados de sitio
  - jQuery UI lo soporta como uno de los posibles patrones
    - ♦ Se denomina **sortable** y se aplica a listas sin selección
- ◆ La lista se transforma en ordenable (sortable) aplicando
  - el método **sortable()** (jQueryUI) a su **objeto jQuery** (DOM)
    - ♦ La selección debe quitarse con el método **disableSelection()**

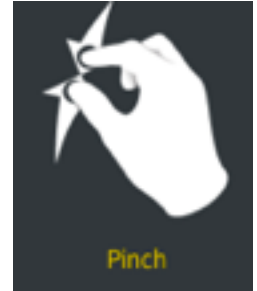
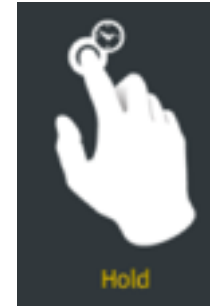
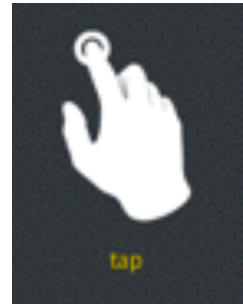
jQuery UI transforma la lista en ordenable (sortable)





# JavaScript: Eventos táctiles

# Eventos táctiles



## ◆ Telefonos y tabletas usan pantallas táctiles

- Estos se manejan con toque (touch) y presionar (taphold)
  - ◆ Las aplicaciones más avanzadas utilizan gestos (gestures): desplazamiento, pellizco, ..

## ◆ W3C ha normalizado eventos táctiles básicos

- Están soportados en los navegadores, pero son muy primitivos, son
  - ◆ touchstart, touchmove, touchend
    - [https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Events/Touch\\_events](https://developer.mozilla.org/en-US/docs/Web/Guide/API/DOM/Events/Touch_events)
- Las aplicaciones suelen utilizar librerías que soportan eventos complejos
  - ◆ Las librerías JavaScript están evolucionando todavía



# Librería jQuery UI Touch Punch

## ◆ jQuery UI Touch Punch (última versión 7-1-15: **1.11.3**)

- Añade **soporte táctil** a **jQuery UI** para adaptación a móviles y tabletas
  - ◆ Coding libre con licencias MIT y GPLv2
    - Ver: <https://github.com/furf/jquery-ui-touch-punch>

## ◆ Transforma eventos de ratón a los equivalentes en pantalla táctil

- Permite arrastrar, ordenar, seleccionar o redimensionar como en el desktop
  - ◆ Utilizando **tocar** (touch) en sustitución de clic y **presionar** (taphold) para arrastrar
- No soporta gestures: swipe, pinch, ....

## ◆ Desarrollada por Dave Furfero

- Información y código: <https://github.com/furf>



# Puzzle para móvil

Añade interacción táctil.

```
<!doctype html>
<html> <head>
<meta charset="utf-8"> <title>PUZZLE</title>
<script src="jquery-2.1.4.js"> </script>
<script src="jquery-ui.min.js"> </script>
<script src="jquery.ui.touch-punch.min.js"> </script>
<style>
```

```
* {
  user-select: none;
  -webkit-touch-callout: none;
  -webkit-tap-highlight-color: rgba(0,0,0,0);
  -webkit-tap-highlight-color: transparent;
```

```
margin:0;
padding:0;
```

Comandos de compatibilidad con pantallas táctiles

```
#sortable {
  list-style-type: none;
  margin-left:auto;
  margin-right:auto;
}
```

```
#sortable li {
  float: left;
  width: 33.3%;
}
```

```
#sortable li img {
  vertical-align:bottom;
  width: 100%;
  height: auto;
```

100vh significa 100% de la altura de la ventana visible. Muy util para responsive.

```
/* Apaisado: anchura = máxima altura */
@media (orientation:landscape){
  #sortable { width:100vh; } /* 100vh: 100% alto ventana */
}
```

```
/* Vertical: se adapta al máximo ancho */
@media (orientation:portrait){
  #sortable { width:100%; }
}
```

```
</style>
.....
```

```
<script>
$(function() {
  $( "#sortable" ).sortable();
  $( "#sortable" ).disableSelection();
});
</script>

</head> <body>

<ul id="sortable">
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
  <li></li>
</ul>

</body></html>
```

## ◆ El uso de jQuery UI Touch Punch

- nos permite adaptar el puzzle muy fácilmente
  - ◆ Solo se hacen cambios CSS para adaptarse a
    - Portrait/Landscape y multi-navegador



# Librerías para eventos táctiles

## ◆ **jQuery mobile:** librería desarrollada por la Fundación jQuery

- Para desarrollar aplicaciones táctiles, incluyendo móviles
  - ◆ La versión actual (7-1-15) es **jQuery mobile 1.4**
    - Documentación y código: <http://jquerymobile.com/>

## ◆ Soporta gestos y eventos táctiles (gestures)

- Pero no es compatible con jQuery UI

## ◆ El soporte a gestos y eventos táctiles en JavaScript

- No esta todavía maduro, hay otras librerías interesantes, como
  - ◆ interact.js, kendo.ui, lungu.js, zepto.js,...

# Gestos táctiles

```
<!DOCTYPE html> // solo funciona en desktop con Firefox o Safari
<html><head><title>Evento jQuery</title><meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="jquery.mobile-1.4.5.min.css">
<script type="text/javascript" src="jquery-2.1.4.min.js" > </script>
<script type="text/javascript" src="jquery.mobile-1.4.5.min.js" > </script>
```

jQuery mobile

```
<style>
*{
  -webkit-touch-callout: none; /* anula el efecto visual del touch prolongado sobre el objeto */
  user-select: none; /* user-select, hace el objeto no seleccionable */
  -webkit-tap-highlight-color: rgba(0,0,0,0); /* anula el feedback de color al hacer tap en un link */
  -webkit-tap-highlight-color: transparent; /* similar anterior en algunos android */
}
```

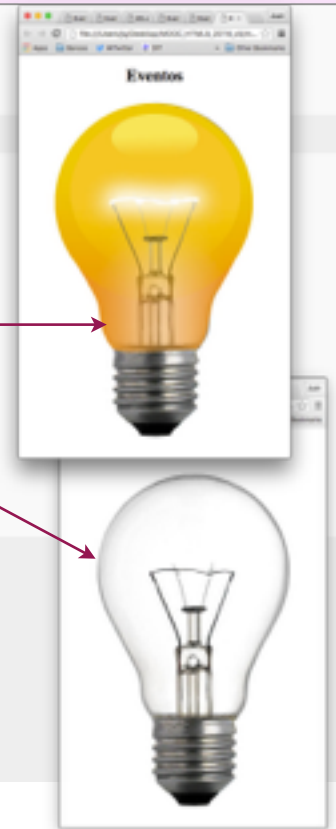
```
body {text-align:center;}
</style>
```

Comandos de compatibilidad con pantallas táctiles

```
<script type="text/javascript">
$(function(){
  $.event.special.tap.emitTapOnTaphold = false;
  $("#i1").on("tap", function() {$('#img').attr('src', 'lamp_on.jpg')});
  $("#i1").on("taphold", function(){$('#img').attr('src', 'lamp_off.jpg')});
  $("#i1").on("swipe", function(){$('#img').attr('src', 'lamp_off.jpg')});
});
```

**Gestos táctiles (gestures). En desktop se usan tap y taphold. OJO! Abrir con Firefox en local. En chrome se ve bien solo en neocities.**

```
</script>
</head>
<body>
  <h1>Eventos jQuery</h1>
  
</body>
</html>
```

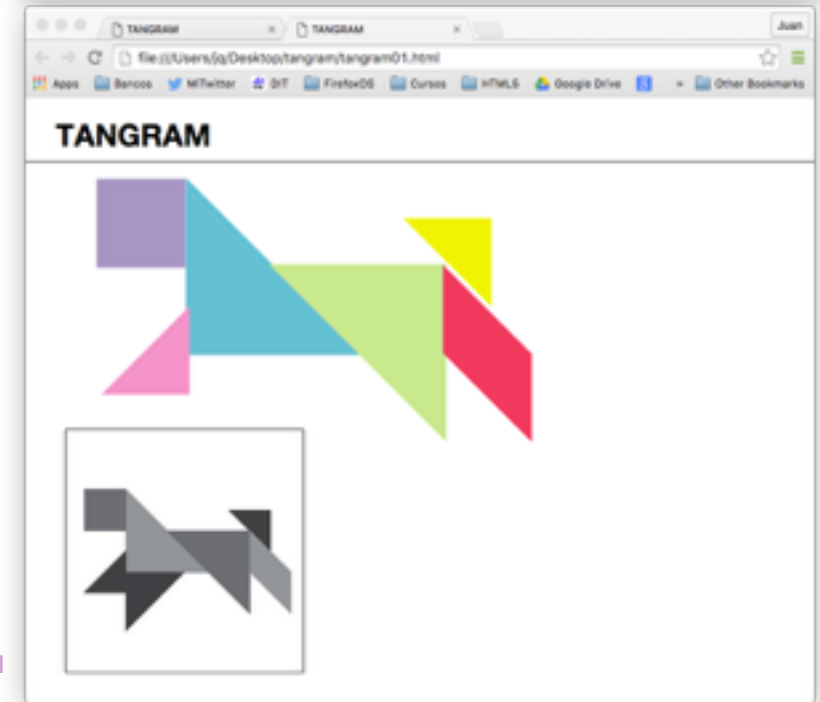
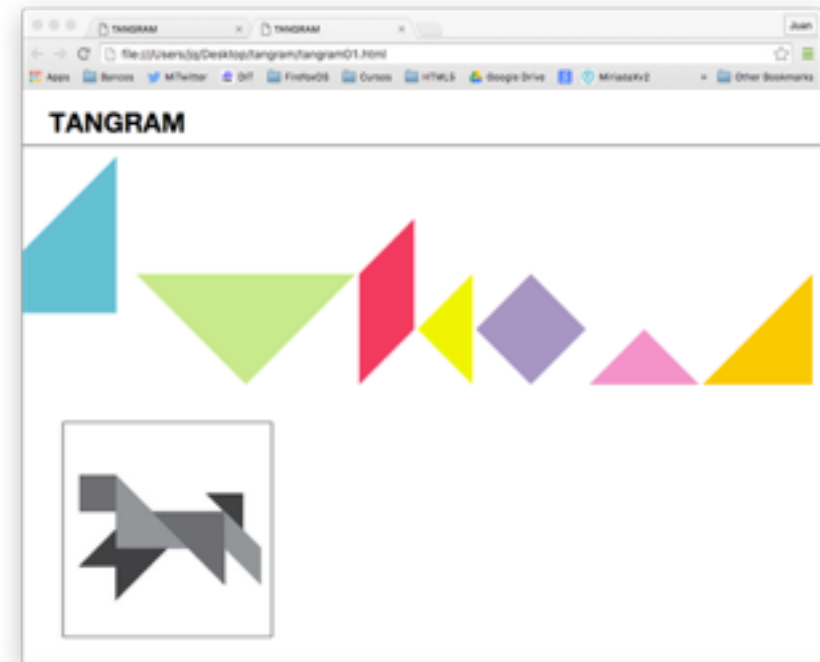




## Ejemplo: Tangram

# Tangram

- ◆ Tangram muestra una lista de figuras geométricas
  - que se pueden arrastrar y rotar
- ◆ Los trozos son figuras geométricas
  - Creadas con SVG (Scalable Vector Graphics)
    - ◆ En ficheros independientes
  - SVG es una extensión de html para gráficos
- ◆ Los trozos de la imagen pueden
  - Arrastrarse en el navegador
    - ◆ Se implementa con la interacción draggable
      - <https://jqueryui.com/draggable/>
  - Girarse haciendo click con el ratón
    - ◆ Se giran con CSS en el manejador del evento click





rotar imágenes utiliza la propiedad CSS transform. Muy eficaz para transformaciones gráficas.

La rotación de las imágenes debe ser independiente. Se calcula en una propiedad diferente creada dinámicamente en el objeto DOM de cada imagen.

```
<script>
function rotar () {
  this.myAngle = (this.myAngle + 45) || 45;
  $(this).css({'transform':'rotate('+this.myAngle+'deg)'});
}

```

```
$(function() {
  $(".draggable").draggable({ stack: "img" });
  $("img").on("click tap", rotar);
});
</script>
</head><body>
```

Rotar imágenes

Arrastrar imágenes

```
<h1>TANGRAM</h1>
```

Imágenes en SVG (reescalables)

```







```

```
<div id="caja" ></div>
</body></html>
```

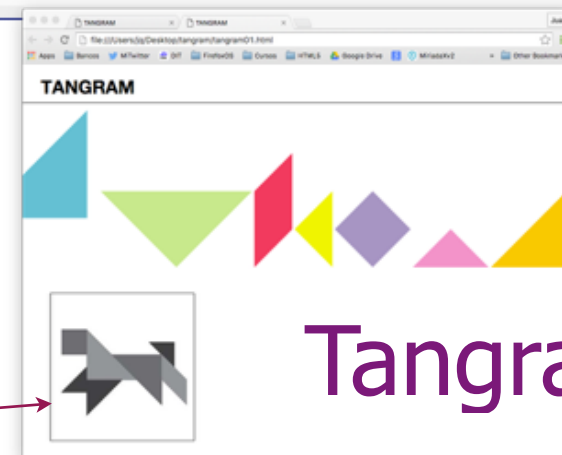
```
<!DOCTYPE html><html><head><title> TANGRAM </title>
<script src="jquery-2.1.4.js"></script>
<script src="jquery-ui.min.js"></script>
<script src="jquery.ui.touch-punch.min.js"></script>
```

Libraries JS

```
<style type="text/css">
body, html { width: 100%; height: 100%;
padding: 0; margin: 0;
overflow: hidden;
}
```

Estilos CSS

```
#caja {
  position: absolute;
  left:5%; bottom:5%;
  border: 1px solid #000000;
}
h1 {
  font-family: Gotham, "Helvetica Neue",
  Helvetica, Arial, sans-serif;
  padding-left: 1em; padding-bottom: 0.2em;
  border-bottom: 1px solid #000000;
}
</style>
```



# Tangram I

# Tangram II

```
<script>
var _45_degrees_more = function () {
  var angle = 0;
  return function(){ return (angle += 45);};
};

$(function() {
  $(".draggable").draggable({ stack: "img" });

  $(".img").each(function (index, elem) {
    var rotate = _45_degrees_more();
    $(elem).on("click tap", function () {
      $(this).css({'transform':'rotate('+rotate()+'deg)'});
    });
  });
});
</script>
</head><body>

<h1>TANGRAM</h1>









<div id="caja" ></div>
</body></html>
```

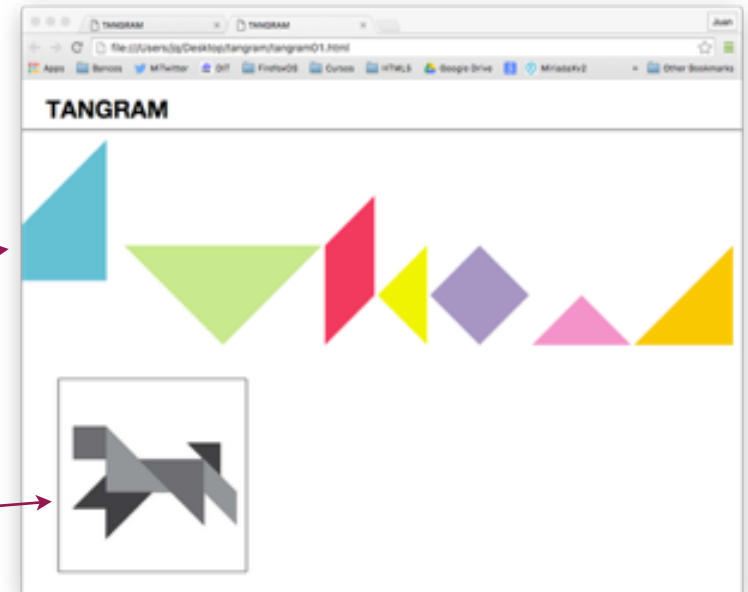
Como la rotación de las imágenes debe ser independiente, se calcula en este cierre que creará una **variable angle diferente** en cada manejador.

Arrastrar imágenes

Instanciar cierre con función \_45\_degrees\_more() que calcula el ángulo.

Rotar imagenes en manejador con propiedad transform de CSS.

Imágenes en SVG (reescalables)



# BOOTSTRAP

Responsive



# Bootstrap

Es un **framework** HTML, CSS y JS que facilita el desarrollo de proyectos responsive, pensado para que sus páginas se visualicen sobre todo en dispositivos móviles (**mobile first**). Es fácil de usar, y es compatible con las últimas versiones de todos los navegadores.

Se descarga en <http://getbootstrap.com/>

Genera las tres carpetas necesarias en un proyecto bootstrap:

## **bootstrap/**

### **css/**

- bootstrap.css
- bootstrap.css.map
- bootstrap.min.css
- bootstrap.min.css.map
- bootstrap-theme.css
- bootstrap-theme.css.map
- bootstrap-theme.min.css
- bootstrap-theme.min.css.map

### **js/**

- bootstrap.js
- bootstrap.min.js

### **fonts/**

- glyphicons-halflings-regular.eot
- glyphicons-halflings-regular.svg
- glyphicons-halflings-regular.ttf
- glyphicons-halflings-regular.woff
- glyphicons-halflings-regular.woff2

Hay plantillas de Bootstrap que se pueden descargar en:

<http://bootstrapzero.com/>

# Mi primera página

En el HTML, deberemos tener estos tres enlaces:

```
<!--Bootstrap CSS -->
```

```
<link rel="stylesheet" href="css/bootstrap.min.css">
```

```
<!--jQuery -->
```

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

```
<!--Bootstrap JavaScript -->
```

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

Al ser un entorno **Mobile first**, tendremos que corregir la escala y ancho de inicio del dispositivo:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

# Estructura Bootstrap

Bootstrap requiere un elemento contenedor general para todo el contenido de la página.

Existen dos tipos identificados por su clase:

Un contenedor de ancho centrado y dependiente del ancho de la ventana del navegador.

```
<div class="container">    ...</div>
```

Contenedor que ocupa todo el espacio de la ventana del navegador.

```
<div class="container-fluid">    ...</div>
```

container-fluid

Párrafo en un contenedor fluido

container

Párrafo en un contenedor normal

# Grid

Bootstrap facilita el diseño empleando una rejilla oculta, un grid de 12 columnas que permite múltiples combinaciones.



El grid tiene 4 tipos de distribución:

**xs** (eXtra Small. teléfonos <768px)

**sm** (Small. tabletas ≥768px)

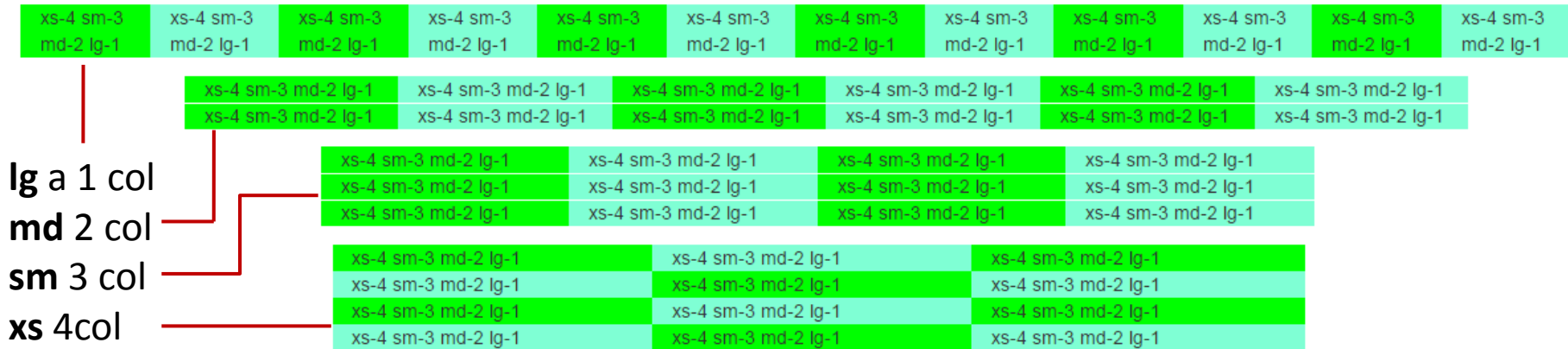
**md** (Medium. escritorio ≥992px)

**lg** (Large. escritorios grandes (≥1200px))



# Grid

## Ejemplo de página con estructura:



```
<div class="container">
```

```
<div class="row">
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
<div class="col-xs-4 col-sm-3 col-md-2 col-lg-1">xs-4 sm-3 md-2 lg-1</div>
```

```
</div>
```

# Menú

Los menús utilizan la clase **.nav** y una clase modificadora: **.nav-tabs** para pestañas, **.nav-pills** para pildoras, se añade la clase **.nav-justified** para justificar el menú al ancho de la caja y **.nav-stacked** para opción vertical.

## Tabs



```
<ul class="nav nav-tabs">
```

## Menú Pills



```
<ul class="nav nav-pills">
```

## Menú Pills Justificado



```
<ul class="nav nav-pills nav-justified">
```

## Menú Pills vertical



```
<ul class="nav nav-pills nav-stacked">
```

Opción 2

Opción 3

Opción 4

# Menú dropdown

Los menús desplegables, muestran una lista de enlaces secundarios al pulsar.

```
<li role="presentation" class="dropdown">  
  <a class="dropdown-toggle" data-toggle="dropdown" href="#" >Opción 3  
  <span class="caret"></span> </a>  
  <ul class="dropdown-menu">  
    <li><a href="#">O3 Opción 1</a></li>  
    <li><a href="#">O3 Opción 2</a></li>  
    <li><a href="#">O3 Opción 3</a></li>  
    <li><a href="#">O3 Opción 4</a></li>  
  </ul> </li>
```

Opción 1

Opción 2

Opción 3 ▼

Opción 4

O3 Opción 1

O3 Opción 2

O3 Opción 3

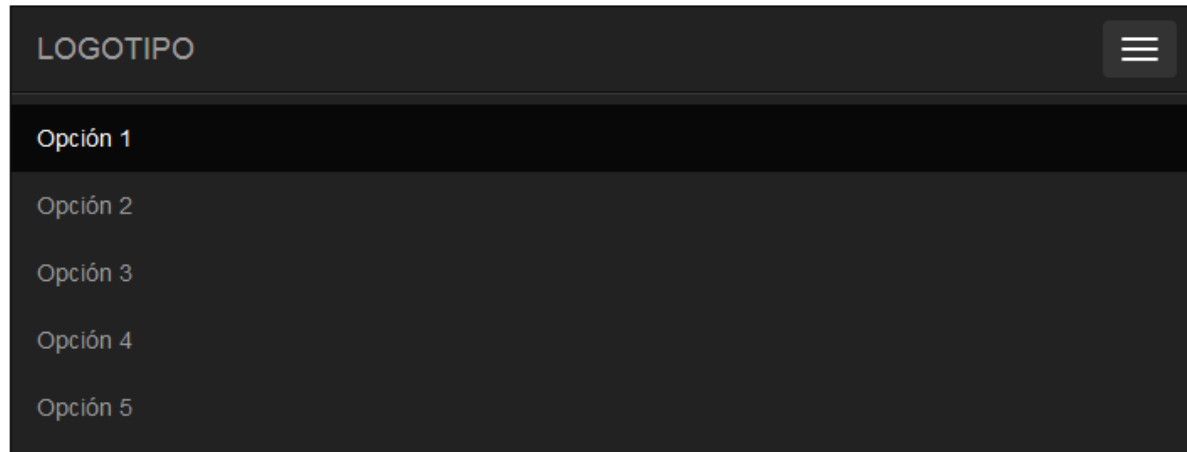
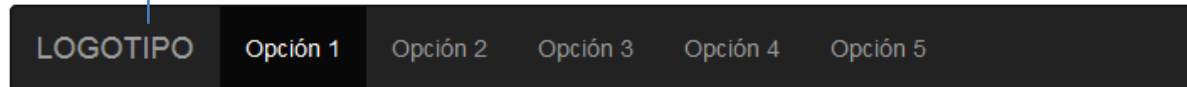
O3 Opción 4

# Menú collapse

Los menús **collapse**, se ocultan cuando en pantallas pequeñas, mostrando un botón a la derecha.

```
<a class="navbar-brand" href="#">LOGOTIPO</a>
```

## Menú collapse



```
<button type="button"
class="navbar-toggle" data-
toggle="collapse" data-
target="#miMenu">
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
  <span class="icon-bar"></span>
</button>
```



# Botones

Bootstrap facilita unos estilos para los botones, y cuatro tamaños diferentes.

`btn-default` | `btn-primary` | `btn-success` | `btn-info` | `btn-warning` | `btn-danger` | `btn-link`

## Botones



## Tamaños



`btn-lg` | `btn-md` | `btn-sm` | `btn-xs`

# Imágenes

Las imágenes con bootstrap tienen varias clases predefinidas.

**.img-rounded**, para redondear las esquinas, lo hace con un border-radius de 6px.

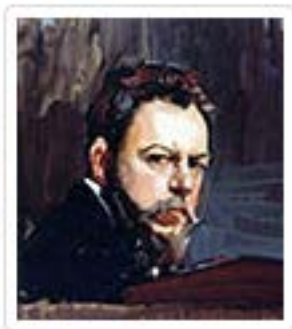
**.img-circle**, pone un borde redondo del 50%

**.img-thumbnail**, muestra la foto con marco y borde

## Imágenes responsive

Para modificar el tamaño de las imágenes y adaptarlo a los cambios del tamaño de las pantallas, para ello usaremos la clase

**.img-responsive**, aplica el ancho máximo del 100% hasta el tamaño real de la imagen y alto automático.



# BOOTSTRAP

Responsive



# Carrusel con bootstrap

# Carrusel con bootstrap

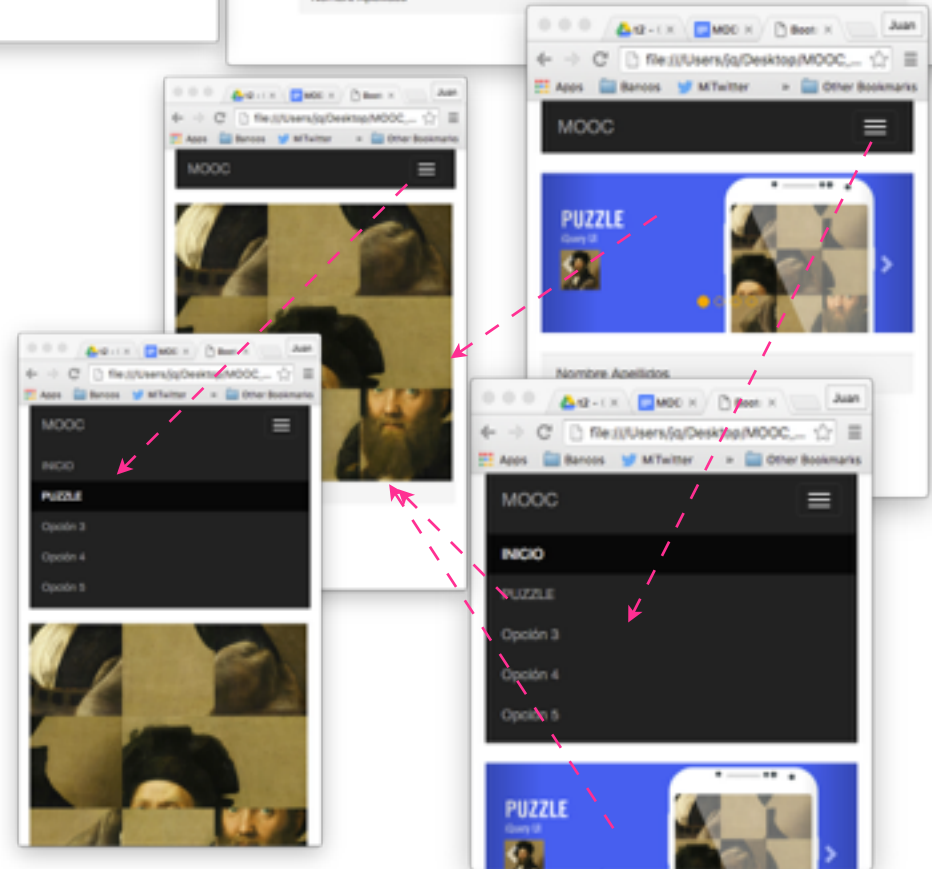


## ◆ Ejemplo de un carrusel bootstrap

- Soporte como componente estándar
  - ◆ en [bootstrap.js/bootstrap.css](http://bootstrap.js/bootstrap.css)
- Componentes bootstrap utilizados son:
  - ◆ <http://getbootstrap.com/javascript/#carousel>
  - ◆ <http://getbootstrap.com/javascript/#collapse>
  - ◆ <http://getbootstrap.com/components/#navbar-brand-image>
  - ◆ <http://getbootstrap.com/css/#overview-container>

## ◆ Ejercicio de este modulo

- Integrar otras apps, además del puzzle



# Barra de navegación

```
<!DOCTYPE html><html lang="en">
<head> <title>Bootstrap Example</title> <meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="css/bootstrap.css" rel="stylesheet">
<script src="js/jquery-1.11.2.min.js"></script>
<script src="js/bootstrap.js"></script>
```

Definir viewport e importar bootstrap y jQuery.

```
<style type="text/css">
.carousel-inner .item img,
.carousel-inner .item img
{ width: 100%;}
.container .carousel-indicators .active { background-color:orange;}
.container .carousel-indicators li { border: 1px solid orange;}
.container .panel-footer { margin-top:20px;}
</style>
```

Estilo CSS adicional al que da bootstrap.

```
</head>
```

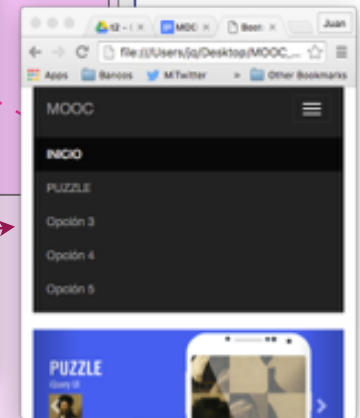
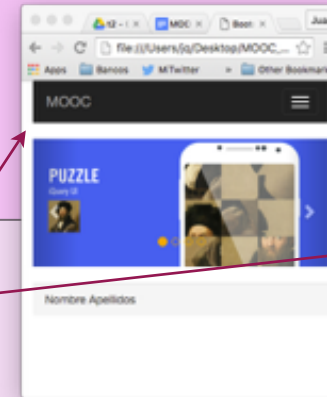
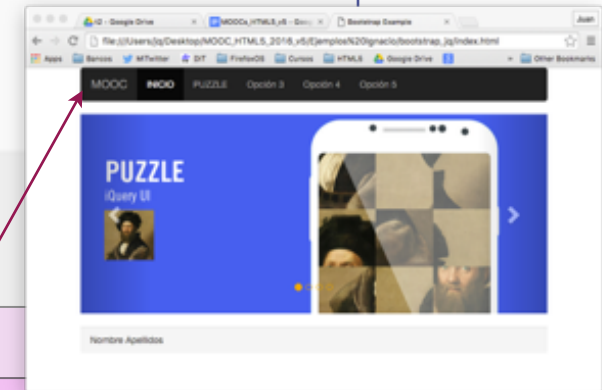
```
<body>
<div class="container">
```

Barra de navegación horizontal

```
<nav class="navbar navbar-inverse ">
<div class="container-fluid">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#miMenu">
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
<a class="navbar-brand" href="index.html">MOOC</a>
</div>
```

```
<div class="collapse navbar-collapse" id="miMenu">
<ul class="nav navbar-nav">
<li class="active"><a href="index.html">INICIO</a></li>
<li><a href="ejercicio_puzzle.html">PUZZLE</a></li>
<li><a href="#">Opción 3</a></li>
<li><a href="#">Opción 4</a></li>
<li><a href="#">Opción 5</a></li>
</ul>
</div>
```

```
</div>
</nav>
```



Barra de navegación colapsada



# Carrusel

Botones de selección del carrusel

Figuras en rotación del carrusel bootstrap. La 1a enlaza con el puzzle adaptado. Los enlaces de las demás se enlazarán a medida que se adapten los ejemplos.

```
<div id="myCarousel" class="carousel slide" data-ride="carousel">
```

```
<!-- menú -->
```

```
<ol class="carousel-indicators">
```

```
<li data-target="#myCarousel" data-slide-to="0" class="active"></li>
```

```
<li data-target="#myCarousel" data-slide-to="1"></li>
```

```
<li data-target="#myCarousel" data-slide-to="2"></li>
```

```
<li data-target="#myCarousel" data-slide-to="3"></li>
```

```
</ol>
```

```
<!-- slides -->
```

```
<div class="carousel-inner" role="listbox">
```

```
<div class="item active">
```

```
<a href="ejercicio_puzzle.html">
```

```

```

```
</a>
```

```
</div>
```

```
<div class="item">
```

```

```

```
</div>
```

```
<div class="item">
```

```

```

```
</div>
```

```
<div class="item">
```

```

```

```
</div>
```

```
</div>
```

```
<a class="left carousel-control" href="#myCarousel" role="button" data-slide="prev">
```

```
<span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
```

```
<span class="sr-only">Siguiente</span>
```

```
</a>
```

```
<a class="right carousel-control" href="#myCarousel" role="button" data-slide="next">
```

```
<span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
```

```
<span class="sr-only">Anterior</span>
```

```
</a>
```

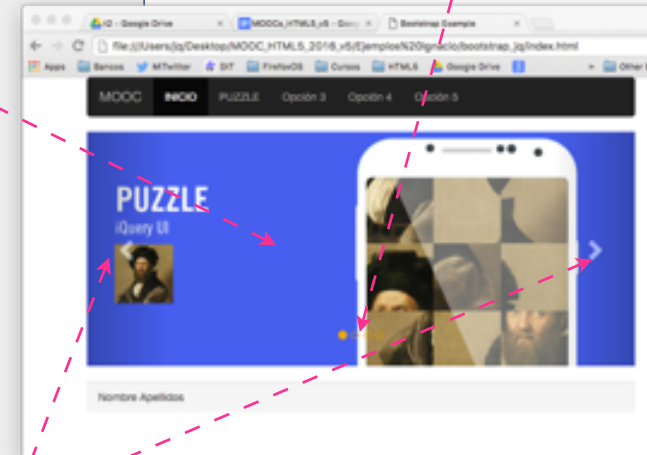
```
</div>
```

```
<div class="panel-footer">Nombre Apellidos</div>
```

```
</div>
```

```
</body>
```

```
</html>
```



Definición de las flechas (izquierda y derecha) de paso a la siguiente aplicación del carrusel.

Bootstrap carousel: <http://getbootstrap.com/javascript/#carousel>

# Puzzle en carrusel

```

<!DOCTYPE html><html lang="en"><head>
<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="css/bootstrap.css" rel="stylesheet">
<style type="text/css">
* { user-select: none;
  -webkit-touch-callout: none;
  .....
}
@media (orientation:portrait) { #sortable {width:100%;}}
</style>
<script src="js/jquery-1.11.2.min.js"></script>
<script src="js/jquery-ui.min.js"></script>
<script src="js/jquery.ui.touch-punch.min.js"></script>
<script src="js/bootstrap.js"></script>

<script>
$(function() {
  $( "#sortable" ).sortable();
  $( "#sortable" ).disableSelection();
});
</script>
</head>
<body>
<div class="container">
<nav class="navbar navbar-inverse ">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#miMenu">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="index.html">MOOC</a>
    </div>
    <div class="collapse navbar-collapse" id="miMenu">
      <ul class="nav navbar-nav">
        <li><a href="index.html">INICIO</a></li>
        <li class="active"><a href="#">PUZZLE</a></li>
        <li><a href="#">Opción 3</a></li>
        <li><a href="#">Opción 4</a></li>
        <li><a href="#">Opción 5</a></li>
      </ul>
    </div>
  </div>
</nav>

<div class="ejercicio">
  <ul id="sortable">
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
  </ul>
</div>
<div class="clear"> </div>
</div>
<div class="panel-footer">Nombre Apellidos</div>
</div>
</body>
</html>

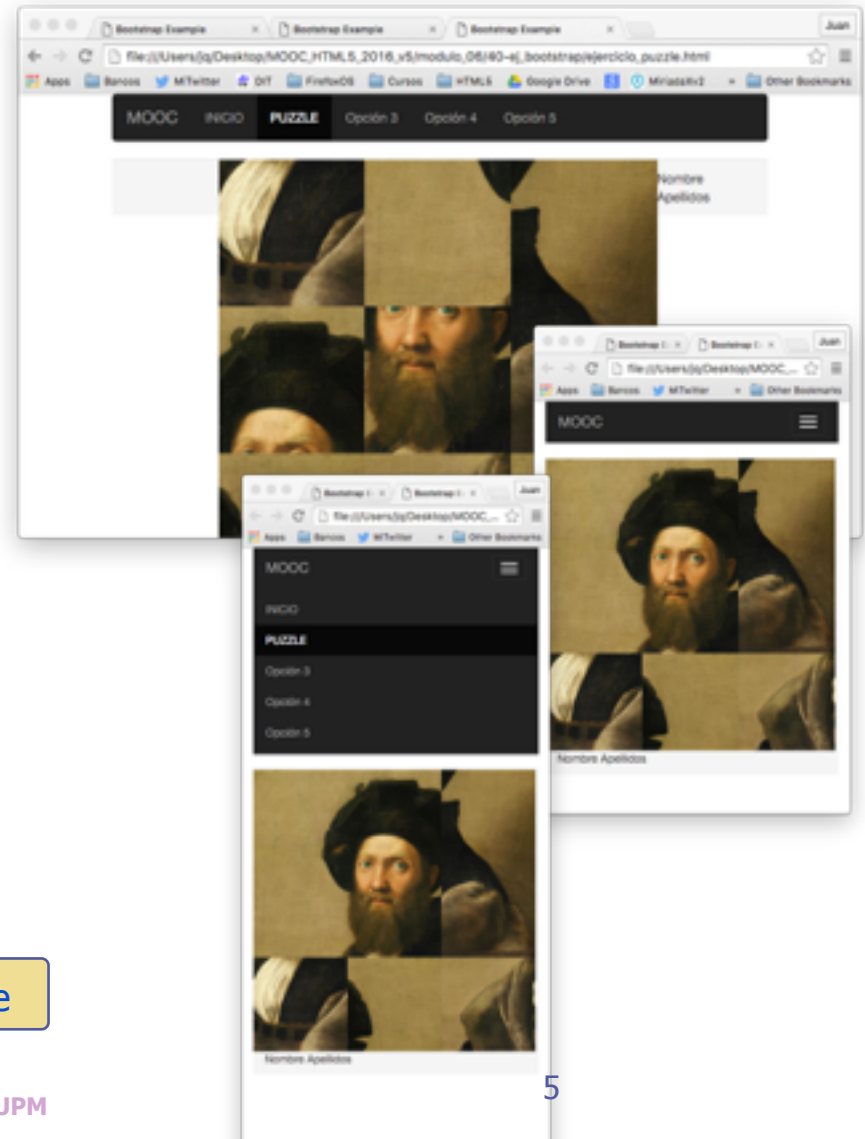
```

Pertenece a carrusel

Pertenece a puzzle

Pertenece a carrusel

Pertenece a puzzle







Final del tema  
Muchas gracias!

