



# Introducción a JavaScript, a sus tipos y valores

# JavaScript



## ◆ JavaScript

- Diseñado por Netscape en 1995 para ejecutar en un Navegador
  - ◆ Hoy se ha convertido en el **lenguaje del Web y Internet**

## ◆ Norma ECMA (European Computer Manufacturers Association)

- Versión soportada en navegadores actuales:
  - ◆ **ES5: ECMAScript v5**, Dic. 2009, (JavaScript 1.5)
- Navegadores antiguos soportan
  - ◆ ES3: ECMAScript v3, Dic. 1999, (JavaScript 1.3)



◆ Tutorial: <http://www.w3schools.com/js/>

◆ Referencia: <http://www.w3schools.com/jsref/>

◆ Libro: “*JavaScript Pocket Reference*”, D. Flanagan, O’Reilly 2012, 3rd Ed.

# Tipos, objetos y valores

## ◆ Tipos de JavaScript

### ■ **number**

- ◆ números: **32**, **1000**, **3.8**



### ■ **boolean**

- ◆ los valores **true** y **false**

**FALSE**  
**true**



### ■ **string**

- ◆ textos internacionalizados (Unicode) delimitados entre comillas o apóstrofes
  - **"Hola, que tal"**, **'Hola, que tal'**, **'Γεια σου, ίσως'**, **'嘿, 也许'**

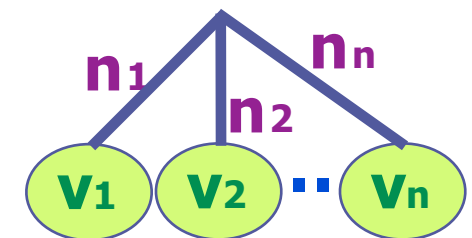
### ■ **undefined**

- ◆ **undefined**: representa **indefinido**

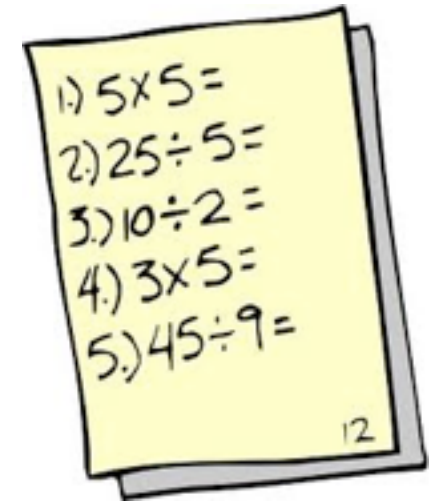
**UNDEFINED**

## ◆ **Objetos**: agregaciones de valores estructurados

- Se agrupan en **clases**: **Object**, **Array**, **Date**, ...
  - ◆ Objeto **null**: valor especial que representa objeto nulo



# Operadores y expresiones



- ◆ JavaScript incluye **operadores** de tipos y objetos
  - Los **operadores** permiten formar **expresiones**
    - ◆ Componiendo **valores** con los operadores
      - Que Javascript evalua hasta obtener un resultado

- ◆ Por ejemplo, con las operaciones aritmeticas +, -, \*, /
  - podemos formar expresiones numéricas

$13 + 7$	$\Rightarrow$	20	// Suma de números
$13 - 7$	$\Rightarrow$	6	// Resta de números
$(8*2 - 4)/3$	$\Rightarrow$	4	// Expresión compleja // Incluyendo paréntesis

# Sobrecarga de operadores

- ◆ Algunos operadores tienen varias semánticas diferentes
- ◆ Por ejemplo, el operador **+** tiene 3 semánticas diferentes
  - **Suma de enteros** (operador binario)
  - **Signo de un número** (operador unitario)
  - **Concatenación de strings** (operador binario)



**13 + 7                      =>    20                      // Suma de números**

**+13                          =>    13                          // Signo de un número**

**"Hola " + "Pepe"   =>   "Hola Pepe"   // Concatenación de strings**



# Conversión de tipos en expresiones

- ◆ JavaScript realiza conversión automática de tipos

- cuando hay ambigüedad en una expresión
  - ◆ utiliza las prioridades para resolver la ambigüedad

- ◆ La expresión **"13" + 7** es ambigua

- porque combina un **string** con un **number**
  - ◆ JavaScript asigna más prioridad al **operador +** de strings, convirtiendo **7** a string

- ◆ La expresión **+"13"** también necesita conversión automática de tipos

- El **operador +** solo está definido para **number**
  - ◆ JavaScript debe convertir el **string "13"** a **number** antes de aplicar operador **+**

13 + 7	=> 20
"13" + "7"	=> "137"
"13" + 7	=> "137"
+"13" + 7	=> 20

Los operadores están ordenados con prioridad descendente. Mas altos más prioridad.

<b>.</b>	<b>Acceso a propiedad o invocar método; índice a array</b>
<b>new</b>	<b>Crear objeto con constructor de clase</b>
<b>()</b>	<b>Invocación de función/método o agrupar expresión</b>
<b>++ --</b>	<b>Pre o post auto-incremento; pre o post auto-decremento</b>
<b>! ~</b>	<b>Negación lógica (NOT); complemento de bits</b>
<b>+ -</b>	<b>Operador unitario, números. signo positivo; signo negativo</b>
<b>delete</b>	<b>Borrar propiedad de un objeto</b>
<b>typeof void</b>	<b>Devolver tipo; valor indefinido</b>
<b>* / %</b>	<b>Números. Multiplicación; división; modulo (o resto)</b>
<b>+</b>	<b>Concatenación de string</b>
<b>-</b>	<b>Números. Suma; resta</b>
<b>&lt;&lt; &gt;&gt; &gt;&gt;&gt;</b>	<b>Desplazamientos de bit</b>
<b>&lt; &lt;= &gt; &gt;=</b>	<b>Menor; menor o igual; mayor; mayor o igual</b>
<b>instanceof in</b>	<b>¿objeto pertenece a clase?; ¿propiedad pertenece a objeto?</b>
<b>== != === !==</b>	<b>Igualdad; desigualdad; identidad; no identidad</b>
<b>&amp;</b>	<b>Operacion y (AND) de bits</b>
<b>^</b>	<b>Operacion ó exclusivo (XOR) de bits</b>
<b> </b>	<b>Operacion ó (OR) de bits</b>
<b>&amp;&amp;</b>	<b>Operación lógica y (AND)</b>
<b>  </b>	<b>Operación lógica o (OR)</b>
<b>?:</b>	<b>Asignación condicional</b>
<b>=</b>	<b>Asignación de valor</b>
<b>OP=</b>	<b>Asig. con operación: += -= *= /= %= &lt;&lt;= &gt;&gt;= &gt;&gt;&gt;= &amp;= ^=  =</b>
<b>,</b>	<b>Evaluación múltiple</b>

## Operadores JavaScript

**3 + 7 => 10**

**"3" + 7 => "37"**

Los operadores están ordenados con prioridad descendente. Mas altos más prioridad.

<b>.</b>	<b>Acceso a propiedad o invocar método; índice a array</b>
<b>new</b>	<b>Crear objeto con constructor de clase</b>
<b>()</b>	<b>Invocación de función/método o agrupar expresión</b>
<b>++ --</b>	<b>Pre o post auto-incremento; pre o post auto-decremento</b>
<b>! ~</b>	<b>Negación lógica (NOT); complemento de bits</b>
<b>+ -</b>	<b>Operador unitario, números. signo positivo; signo negativo</b>
<b>delete</b>	<b>Borrar propiedad de un objeto</b>
<b>typeof void</b>	<b>Devolver tipo; valor indefinido</b>
<b>* / %</b>	<b>Números. Multiplicación; división; modulo (o resto)</b>
<b>+</b>	<b>Concatenación de string</b>
<b>+ -</b>	<b>Números. Suma; resta</b>
<b>&lt;&lt; &gt;&gt; &gt;&gt;&gt;</b>	<b>Desplazamientos de bit</b>
<b>&lt; &lt;= &gt; &gt;=</b>	<b>Menor; menor o igual; mayor; mayor o igual</b>
<b>instanceof in</b>	<b>¿objeto pertenece a clase?; ¿propiedad pertenece a objeto?</b>
<b>== != === !==</b>	<b>Igualdad; desigualdad; identidad; no identidad</b>
<b>&amp;</b>	<b>Operacion y (AND) de bits</b>
<b>^</b>	<b>Operacion ó exclusivo (XOR) de bits</b>
<b> </b>	<b>Operacion ó (OR) de bits</b>
<b>&amp;&amp;</b>	<b>Operación lógica y (AND)</b>
<b>  </b>	<b>Operación lógica o (OR)</b>
<b>?:</b>	<b>Asignación condicional</b>
<b>=</b>	<b>Asignación de valor</b>
<b>OP=</b>	<b>Asig. con operación: += -= *= /= %= &lt;&lt;= &gt;&gt;= &gt;&gt;&gt;= &amp;= ^=  =</b>
<b>,</b>	<b>Evaluación múltiple</b>

## Operadores JavaScript

**8\*2 - 4 => 12**



# Ejemplo: Script Expresión

- ◆ Script: programa JavaScript insertado en una página HTML
  - Delimitado con la marca **<script>** con atributo “type=text/javascript”
    - ◆ Se ejecuta al cargar la página HTML en el navegador
- ◆ Función Alert("mensaje"): genera una ventana desplegable
  - Que muestra el mensaje pasado como parámetro

```

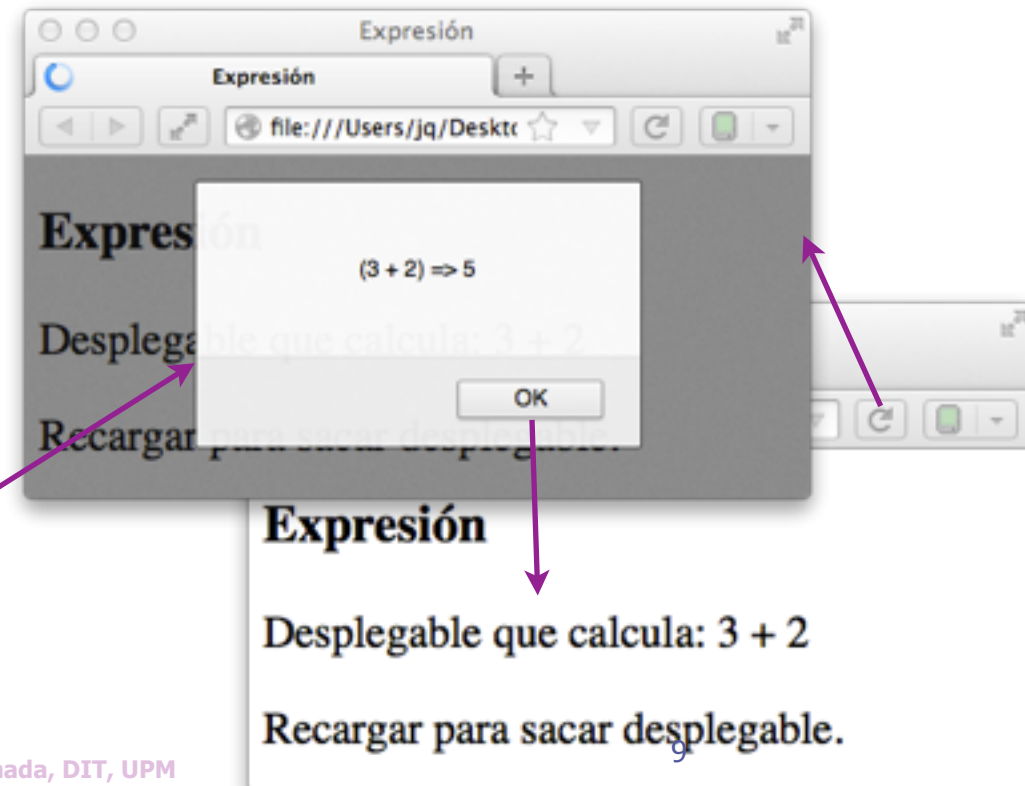
<!DOCTYPE html>
<html><head><title>Expresión</title>
  <meta charset="UTF-8"></head>

<body>
<h3>Expresión</h3>

Desplegable que calcula: 3 + 2
<p>
Recargar para sacar desplegable.

<script type="text/javascript">
  alert("(3 + 2) => " + (3+2));
</script>
</body>
</html>
  
```

© Juan Quemada, DIT, UPM



# Operador typeof

- ◆ El operador **typeof** permite conocer el tipo de un valor
  - Devuelve un string con el nombre del tipo
    - ◆ "number", "string", "boolean", "undefined", "object" y "function"

**typeof 7**

=> "number"



**typeof "hola"**

=> "string"



**typeof true**

=> "boolean"

**FALSE**  
**true**

**typeof undefined**

=> "undefined"

**UNDEFINED**

**typeof null**

=> "object"

**typeof new Date()**

=> "object"



**typeof new Function()**

=> "function"

**f(x)**



# Sentencias, variables y comentarios de JavaScript

# Programa, Sentencia y Comentario

- ◆ Un programa es un conjunto de sentencias que
  - Realizan tareas cuando se ejecutan en un ordenador
    - ◆ Las sentencias se ejecutan en el orden en que han sido definidas
      - Su ejecución provoca cambios en el estado interno o en el exterior del ordenador
- ◆ Los comentarios son mensajes añadidos al programa
  - Solo añaden documentación y facilitan su comprensión
- ◆ En JavaScript hay 2 tipos de comentarios
  - Monolínea: empiezan por `// ..comentario..`
  - Multilínea: Se delimitan con `/* ..comentario.. */`
    - ◆ **OJO!** Los comentarios multilínea tienen problemas con las expresiones regulares

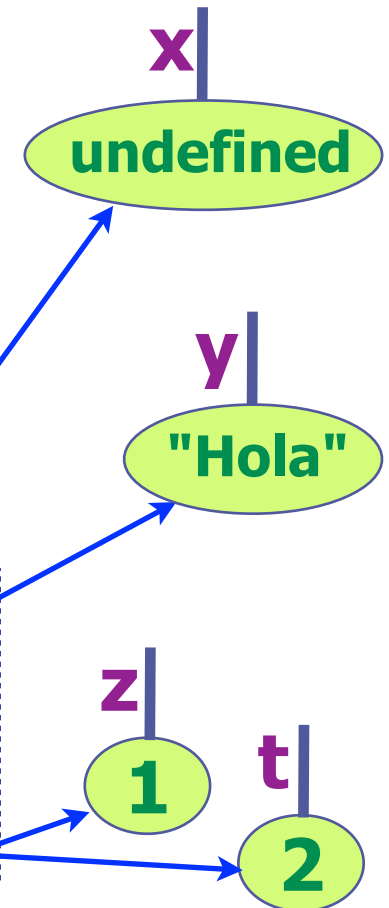
`/* Ejemplo de comentario multilínea que ocupa 2 líneas  
-> al tener ambigüedades, no se recomienda su utilización */`

`alert("Mi mensaje"); // Comentario monolínea que acaba aquí`

# Definición de variables locales

- ◆ La sentencia de definición de variables locales
  - Comienza con la palabra reservada **var**
    - ◆ Seguida de la variable, a la que se puede asignar un valor inicial
- ◆ Se pueden definir varias variables en una sentencia
  - separando las definiciones con coma

```
var x;           // crea la variable x y asigna undefined  
  
var y = "Hola";  // crea y, asignandole el valor "Hola"  
  
var z = 1, t = 2; // crea z e t, asignandosles 1 y 2 respectivamente
```



# Asignación de variables

- ◆ La **ejecución** de una sentencia de definición de variable
  - **Crea la variable** y le asigna un valor inicial (en su caso)
- ◆ Una variable ya creada es un contenedor de valores
  - Una **sentencia de asignación** asigna un nuevo valor a la variable
- ◆ Las variables de JavaScript son **no tipadas**
  - pueden contener **valores de cualquier tipo** de JavaScript

```
var x = 5;    // Crea la variable x y le asigna el valor inicial 5
```

```
x = "Hola";  // Asigna el string (texto) "hola" a la variable x
```

```
x = new Date(); // Asigna objeto Date a la variable x
```

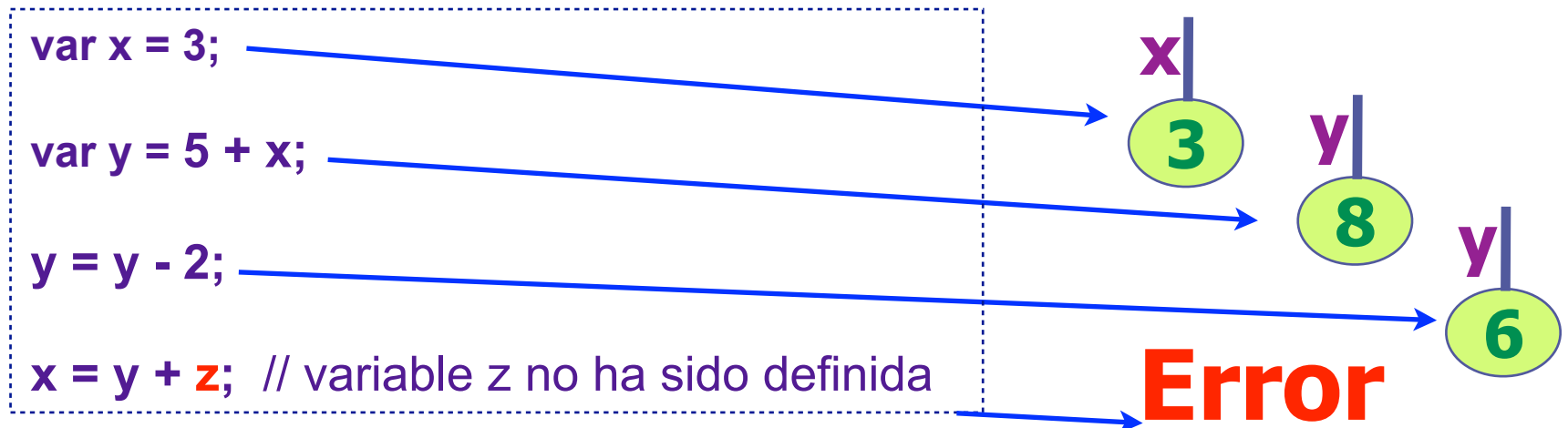
x |  
5

x |  
"Hola"

x |  
Mon Sep 02 2013 09:16:47  
GMT+0200 (CEST)

# Expresiones con variables

- ◆ Una variable representa el valor que contiene
  - Puede ser usada en expresiones como cualquier otro valor
- ◆ Una variable puede utilizarse en la expresión que se asigna a ella misma
  - La parte derecha usa el valor anterior a la ejecución de la sentencia
    - ◆ En `y = y - 2;` la variable `y` tiene el valor 8, por lo que se asigna a `y` un **6** ( $8-2$ )
- ◆ Usar una **variable no definida** en una expresión
  - provoca un **error** y la ejecución del **programa se interrumpe**



# Ejecución de expresiones

- ◆ La sentencia de **ejecución de expresiones** contiene solo una expresión
  - que se calcula al ejecutar la sentencia
    - ◆ Por ejemplo, **(3 + 2)** o **alert("Mi mensaje")** son sentencias correctas (expresiones)
- ◆ Estas sentencias **solo tienen utilidad** si producen **efectos laterales**
  - Como **alert("Mi mensaje")** que muestra una ventana desplegable
- ◆ Una expresión siempre se evalúa en un valor
  - Si ese valor no se guarda en una variable y no hay ningun efecto lateral
    - ◆ La instrucción no tiene ningun efecto en el programa, solo consume recursos

```
var x = 3;
```

```
x = 5 + x;      // sentencia útil, porque la asignación guarda el resultado
```

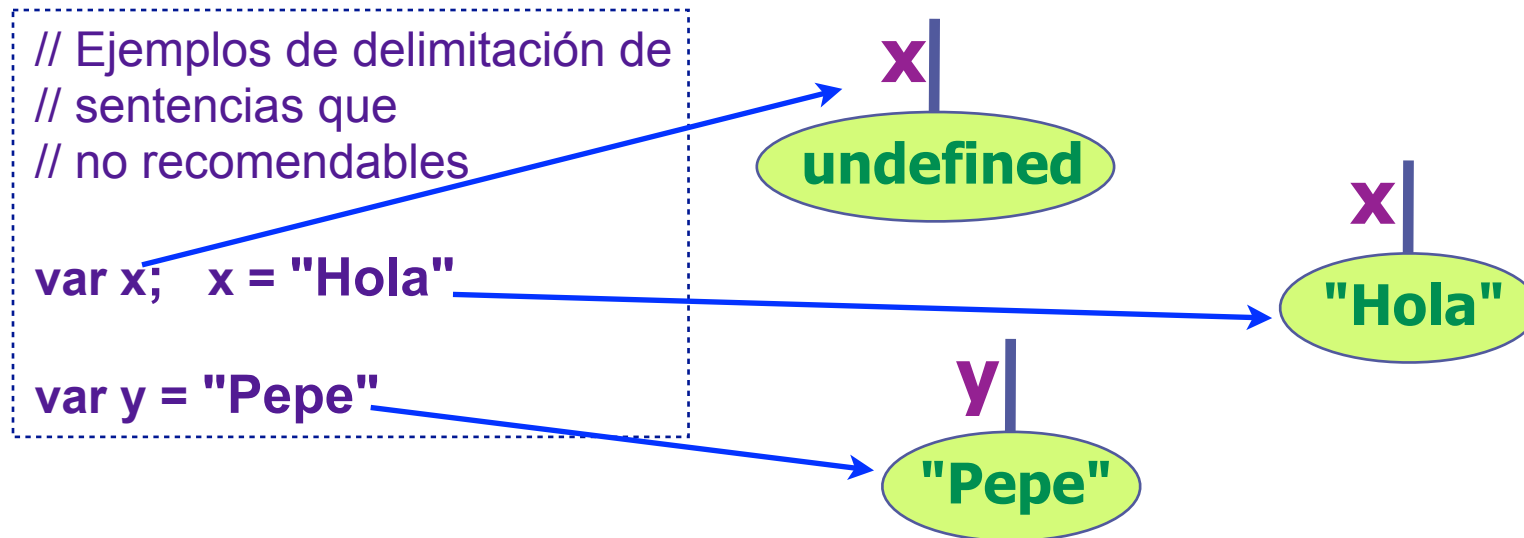
```
x*5 - 2;       // expresión correcta, pero inutil en un programa
```

```
alert("Mi mensaje"); // expresión útil porque tiene efecto lateral
```



# Delimitación de sentencias

- ◆ “;” delimita el final de una sentencia
- ◆ El final de sentencia también puede demitarse con nueva linea
  - Pero hay ambigüedades y no se recomienda hacerlo
- ◆ **Se recomienda terminar siempre con “;”** -> es mas legible y seguro



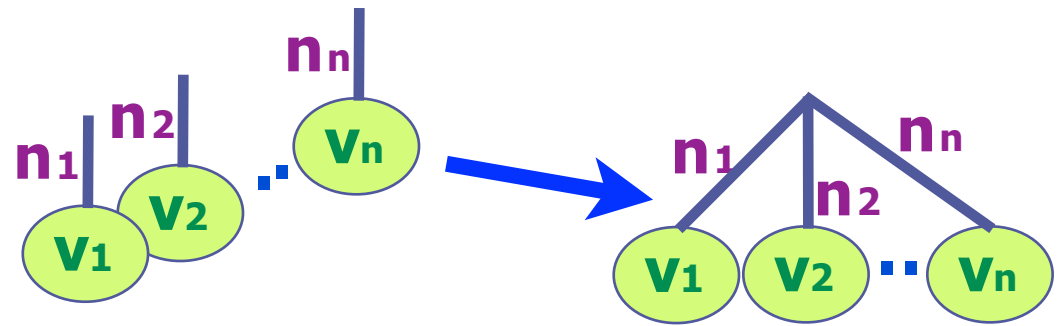
# Nombres de variables

- ◆ El **nombre** (o identificador) de una variable debe comenzar por:
  - **letra**, **\_** o **\$**
    - ◆ El nombre pueden contener además **números**
  - Nombres **bien contruidos**: **x**, **ya\_vás**, **\$A1**, **\$**, **\_43dias**
  - Nombres **mal contruidos**: **1A**, **123**, **%3**, **v=7**, **a?b**,
- ◆ Un nombre de variable
  - **no** debe ser una **palabra reservada** de JavaScript
- ◆ Las variables son sensibles a **mayúsculas**
  - **mi\_var** y **Mi\_var** son variables distintas



# Objetos, propiedades y métodos de JavaScript

# Objetos



- ◆ Los objetos son colecciones de variables
  - agrupadas como un elemento estructurado que llamamos objeto
    - ◆ Las variables se denominan **propiedades**

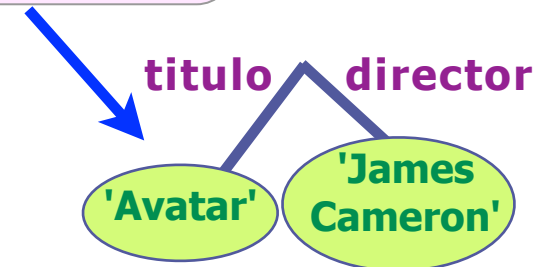
- ◆ Una **propiedad** es un par **nombre:valor** donde
  - los **nombres** deben ser **todos diferentes** en un mismo objeto

- ◆ Se definen con el literal: **{ nombre:valor, ... }**

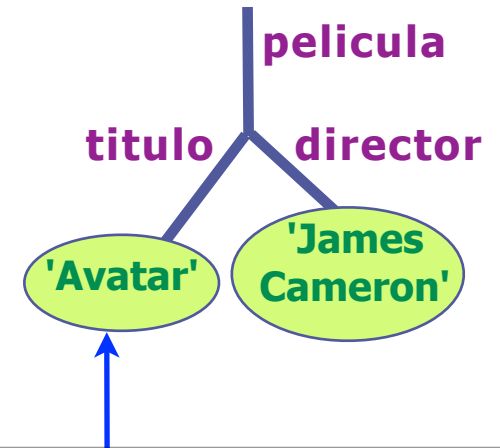
- **Por ejemplo:** {titulo: 'Avatar', director: 'James Cameron'}

- ◆ crea un objeto con 2 propiedades:

- titulo: 'Avatar'
- director: 'James Cameron'

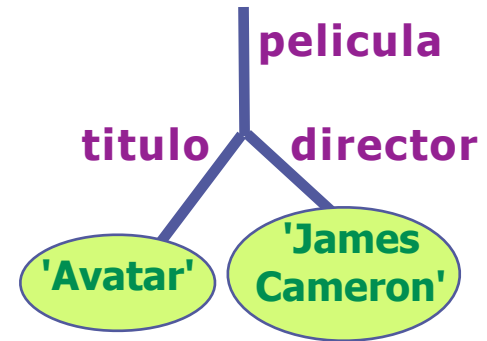


# Propiedades



- ◆ El acceso a propiedades utiliza el operador punto
  - **obj.propiedad**
- ◆ Por ej. en: `var pelicula = {titulo: 'Avatar', director: 'James Cameron'}`
  - **pelicula.titulo**       => "Avatar"
  - **pelicula.director**   => "James Cameron"
  - **pelicula.fecha**       => undefined     // la propiedad fecha no existe
- ◆ Aplicar el operador punto sobre **undefined** o **null**
  - Provoca un **Error\_de\_ejecución** y aborta la ejecución del programa
- ◆ La notación punto solo acepta nombres de propiedades
  - Con la sintaxis de variables: **a, \_method, \$1, ...**
    - ◆ No son utilizables: **"\$43", "?a=1", .....**

# Notación array



- ◆ La notación array permite acceder también a propiedades
  - **objeto["propiedad"]** es equivalente a **objeto.propiedad**
- ◆ en el ejemplo anterior: **var pelicula = {titulo: ..., director: .....}**
  - **pelicula["titulo"]** es equivalente a **pelicula.titulo**
- ◆ La notación array permite utilizar **strings arbitrarios** como nombres
  - **objeto["El director"], pelicula[""]** o **a["%43"]**
    - ◆ **OJO!** es conveniente utilizar siempre nombres compatibles con notación punto
- ◆ Strings arbitrarios como nombres son posibles también en un literal:
  - Por ej. **{"titulo": 'Avatar', "El director": 'James Cameron'}**

# Métodos



- ◆ Un **método** es una operación realizable sobre un objeto
  - Se invoca con la notación punto: **objeto.metodo(..params..)**
- ◆ Un objeto **hereda** métodos de su **clase**
  - Además pueden definirse métodos adicionales
    - ◆ Invocar un **método no heredado ni definido** sobre un objeto da **error\_de\_ejecución**
- ◆ Por ejemplo
  - `document.write("Hola Mundo!")`
    - ◆ inserta string "Hola Mundo!" en página Web
  - `document.writeln("Hola Mundo!")`
    - ◆ inserta string "Hola Mundo!" y nueva linea en la página Web

# Clases y herencia

- ◆ Todos los objetos de JavaScript pertenecen a la **clase Object**
  - Las clases predefinidas en Javascript derivan de Object
    - ◆ **Array, Function, ....**
- ◆ La clase **Object** tiene el constructor **Object()**
  - que permite crear objetos con el operador **new**
    - ◆ **new Object()** crea un objeto vacío equivalente a **{}**
- ◆ Los objetos heredan propiedades y métodos de su clase padre
  - p. e., todo objeto de la clase **Object** hereda el método **toString()**
    - ◆ que devuelve el string **"[object Object]"** cuando se invoca sobre un objeto





# Ejemplos de Scripts JavaScript

# Script I: genera HTML al vuelo

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

```
<h2>Script: genera HTML al vuelo</h2>
```

```
<script type="text/javascript">
```

```
document.write("<b>Fecha:</b> " + (new Date()) + "<p>");
```

```
var pelicula = { titulo: 'Avatar', director: 'James Cameron' };
```

```
document.write("<b>pelicula.titulo =></b> " + pelicula.titulo);
```

```
document.write("<br><b>pelicula['titulo'] =></b> " + pelicula['titulo']);
```

```
</script>
```

```
</body>
```

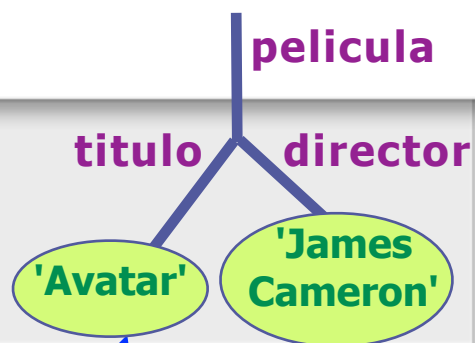
```
</html>
```

## Script: genera HTML al vuelo

**Fecha:** Mon Jan 20 2014 18:37:15 GMT+0100 (CET)

**pelicula.titulo =>** Avatar

**pelicula['titulo'] =>** Avatar



# Script II: texto formateado con <pre>

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

  <!-- &, < y > generan los caracteres
       &, < y >, que no están permitidos en HTML -->

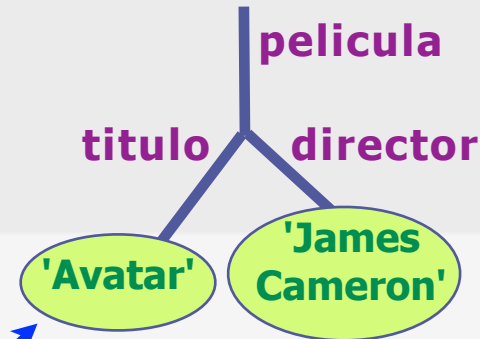
<h2>Script: texto formateado con <pre></h2>

  <!-- Blancos y nuevas líneas del string
       son visibles dentro de un bloque <pre> -->

<pre>
<script type="text/javascript">
  document.writeln("Fecha: " + (new Date()));

  var pelicula = { titulo: 'Avatar', director: 'James Cameron' };

  document.writeln();
  document.writeln("pelicula.titulo      => " + pelicula.titulo);
  document.writeln("pelicula['titulo']   => " + pelicula['titulo']);
</script>
</pre></body></html>
```





# Aplicaciones de cliente

## ◆ Programa que se ejecuta en el navegador

- Se identifica por el URL de la página Web que los contiene
  - ◆ Por ejemplo: <http://upm.es/misapps/webapp.html>
- Se descarga con HTTP del servidor

1) Cliente solicita WebApp identificada con URL

2) Script se ejecuta al cargar la página Web en el navegador:



Solicitud HTTP asociada a un URL

Respuesta HTTP: página Web con script



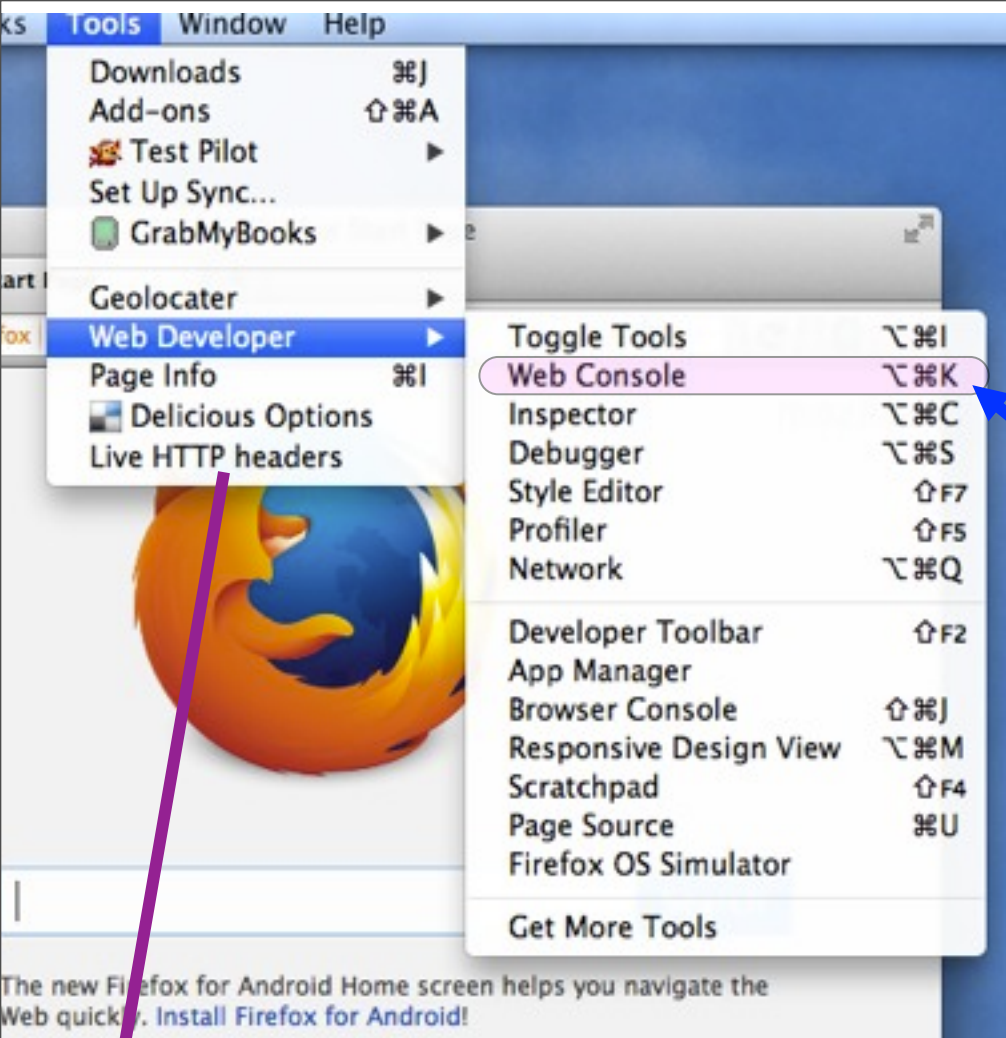
Servidor sirve fichero identificado por URL

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

<h2>Script: genera HTML al vuelo</h2>

<script type="text/javascript">
  document.write("Fecha: " + (new Date()) + "<op>");

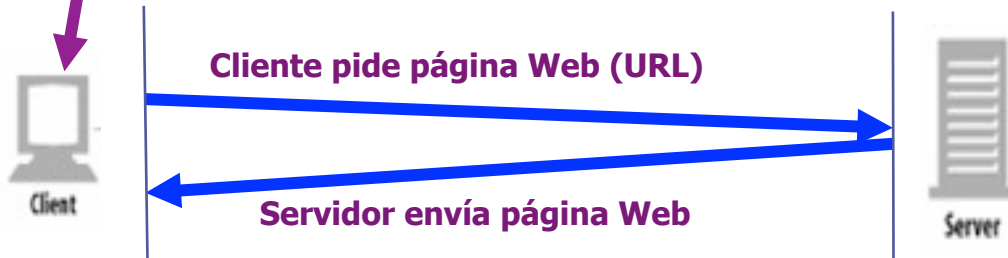
  var pelicula = { titulo: 'Avatar', director: 'James Cameron' };
  document.write("pelicula.titulo => " + pelicula.titulo);
  document.write("<br>pelicula['titulo'] => " + pelicula['titulo']);
</script>
</body>
</html>
```



# Firefox - HTML5 Web Developer

- ◆ Firefox tiene herramientas de desarrollo muy eficaces, en
  - Tools -> Web Developer -> \*\*
- ◆ **Consola JavaScript**
  - Permite ejecutar sentencias
  - Muestra errores en scripts
  - .....
- ◆ Los demás navegadores
  - tienen herramientas similares
- ◆ Explicadas en Screencasts

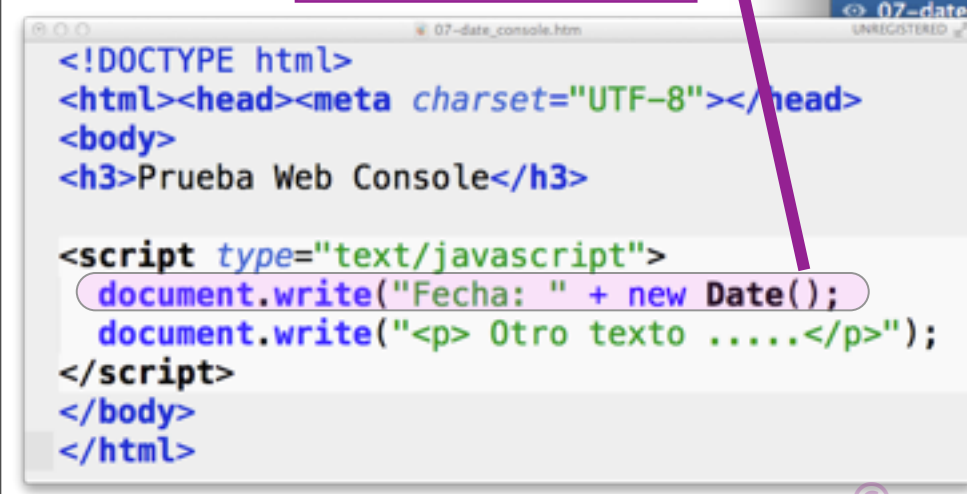
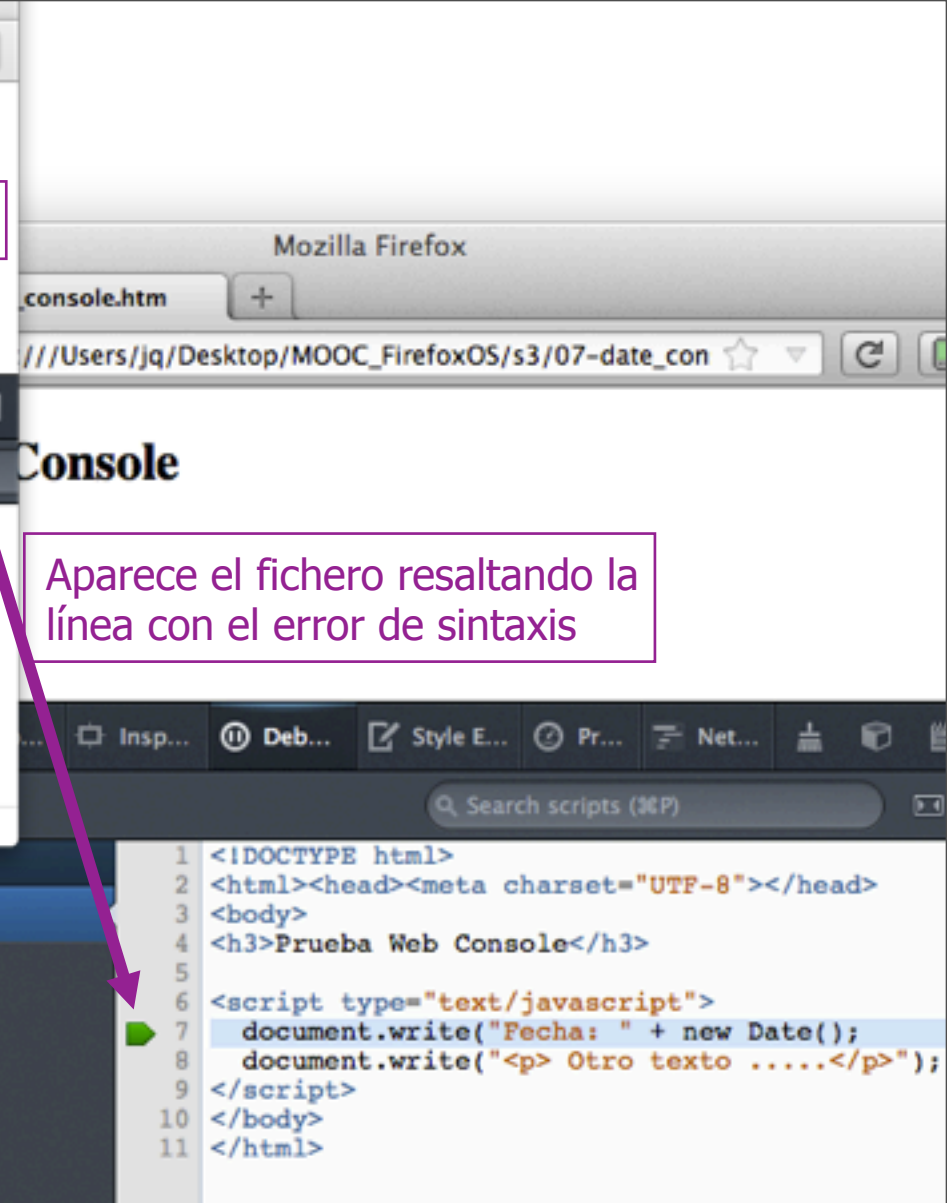
The new Firefox for Android Home screen helps you navigate the Web quickly. Install Firefox for Android!

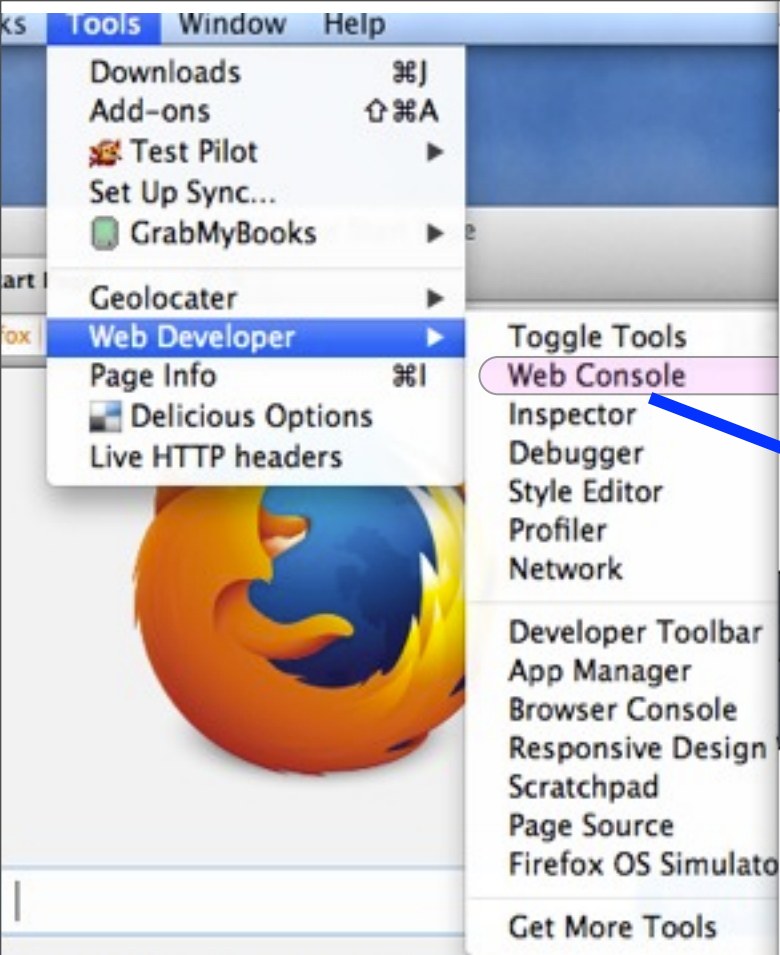






Firefox detecta que falta el paréntesis, aborta la ejecución y lo notifica en la consola





file:///Users/jq/Desktop/MOOC\_I

# Prueba Web Console

Fecha: Thu Jan 30 2014 16:34:29 GMT+0100 (CET)

Otro texto .....

Developer Toolbar

- App Manager
- Browser Console
- Responsive Design
- Scratchpad
- Page Source
- Firefox OS Simulator

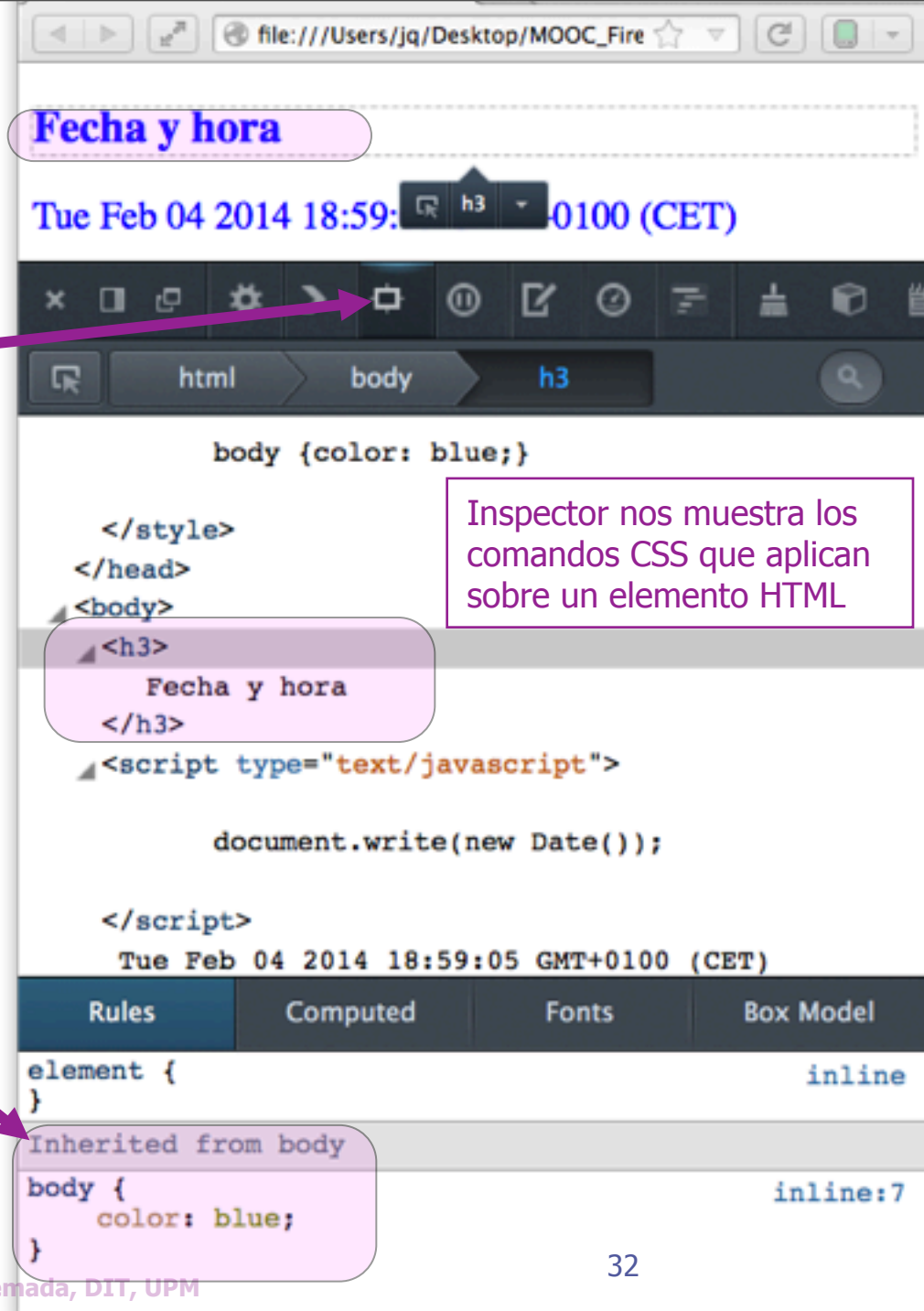
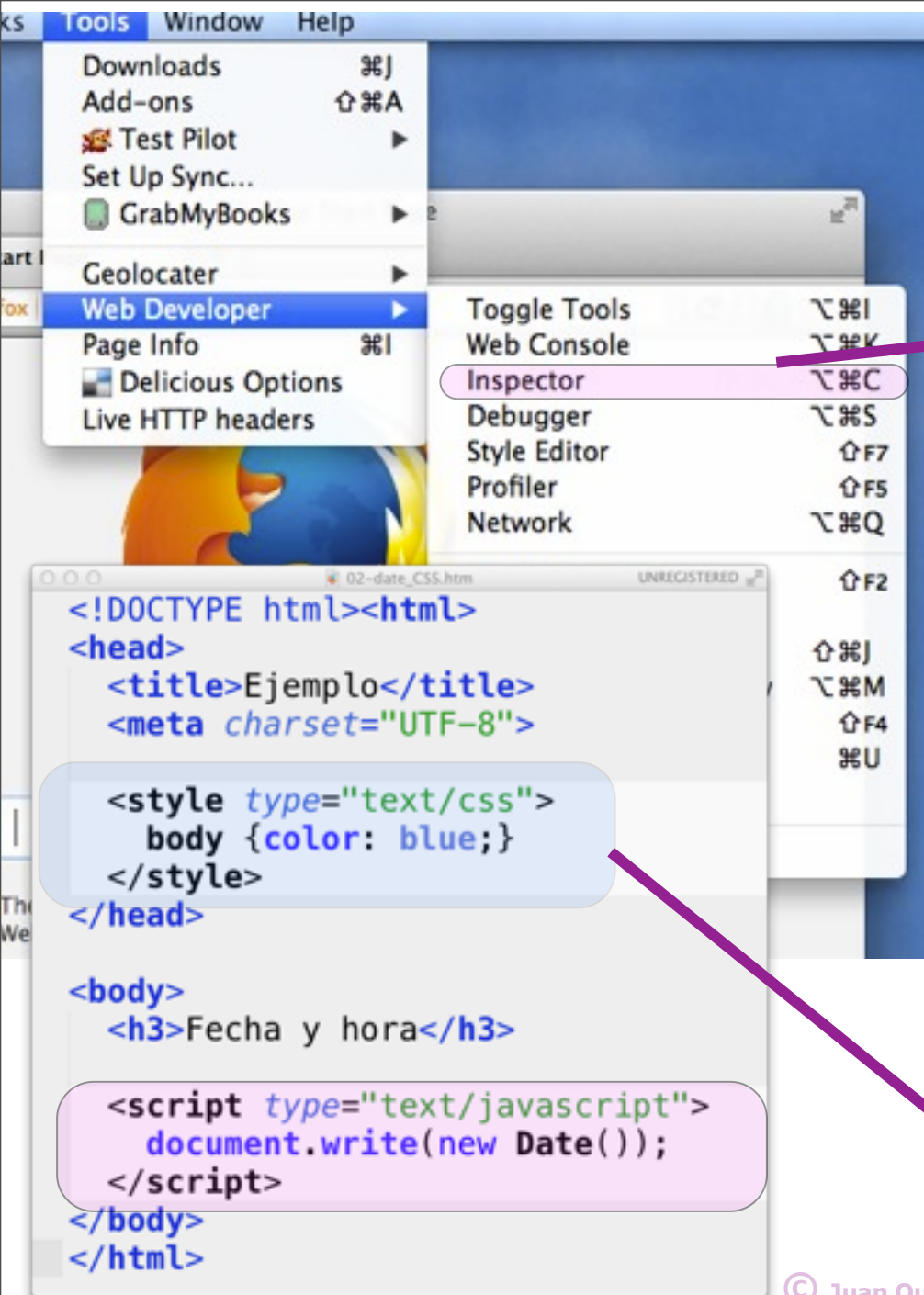
Get More Tools

Net CSS JS Security Logging

```
16:38:41.253 < var pelicula = { titulo: 'Avatar', director: 'James Cameron' };
16:38:41.256 > undefined
16:38:45.077 < pelicula.titulo
16:38:45.079 > "Avatar"
16:38:50.703 < pelicula.director
16:38:50.706 > "James Cameron"
16:39:03.134 < pelicula.fecha
16:39:03.136 > undefined
16:39:30.404 < 8*4 - 2
16:39:30.406 > 30
>> alert("Mensaje")
```

La consola nos va mostrando el resultado de ejecutar las sentencias JavaScript

Aquí se introduce la sentencia





# Publicar en la nube (Google Drive)

# Recursos públicos y de acceso restringido

## ◆ Una página Web o un recurso en Internet

- puede ser **público** o de **acceso restringido**

## ◆ Recurso **público**

- Para ser accedido en **modo lectura** por cualquiera en Internet
  - ◆ Una página Web, una foto, una película, etc.

## ◆ Recurso de **acceso restringido**

- Utilizado en aplicaciones Web, como Google Drive
  - ◆ Puede ser creado, editado o borrado, pero solo por usuarios autorizados
- Autorización: se concede si el usuario se autentica y tiene derecho
  - ◆ Autenticación: por usuario y contraseña, eDNI, huella dactilar, ....

# Publicación en la nube: Google Drive

- ◆ Se debe utilizar una cuenta de Google
  - Para publicar páginas Web y Apps en **Google Drive**
    - ◆ <http://drive.google.com>
- ◆ Google Drive permite crear **directorios** con acceso **público**
  - El directorio se crea y edita con **acceso restringido**
    - ◆ Pero el directorio y sus recursos tienen además **acceso público**
      - Con **otro URL** que da acceso en modo solo lectura
- ◆ Este segundo **URL público** permite
  - **Probar** que los ejercicios funcionan en Internet
    - ◆ accediendo a ellos desde el PC, móviles y otros terminales
  - **Entregar** los ejercicios funcionando, para que otros los corrijan

# Crear un directorio público I

◆ Seguir pasos 1 y 2 para crear un directorio en Google Drive

The screenshot shows the Google Drive web interface. The browser tabs include 'Recibidos (1) - juan.que...' and 'Ejercicios - Google Drive'. The address bar shows the URL 'https://drive.google.com/?tab=mo&authuser=0#folders/OB48KCWfVwCIEUHjk...'. The Google logo is at the top left, followed by a search bar and user information '+Juan'. The 'Drive' header is on the left. Below it, there are 'CREATE' and 'upload' buttons. The left sidebar shows 'My Drive' with a folder named 'Ejercicios' and subfolders 't2', 'Shared with Me', 'Starred', 'Recent', and 'More'. A 'Connect Drive to your desktop' button is also visible. The main area shows 'My Drive > Ejercicios'. A 'New folder' dialog box is open in the center, with the text 'Please enter a name for the new folder:' and a text input field containing 'Ejercicios'. Below the input field are 'Create' and 'Cancel' buttons. Annotations with arrows point to specific elements: '0. En modo cuadrícula' points to the grid view icon in the top right; '1. clicar para crear nueva carpeta' points to the '+ New' button in the top left; '2. Escribir nombre y clicar para crear la carpeta' points to the 'Create' button in the dialog box.

0. En modo cuadrícula

1. clicar para crear nueva carpeta

2. Escribir nombre y clicar para crear la carpeta

36

# Crear un directorio público II

◆ Seguir pasos 1, 2 y 3 para acceder al cambio de permisos

The screenshot shows the Google Drive web interface. On the left, the 'My Drive' sidebar lists folders including 'Ejercicios'. A purple box labeled '1. clicar para seleccionar' points to a checkmark icon next to the 'Ejercicios' folder. A purple box labeled '2. clicar para compartir (share)' points to a share icon (person with a plus sign) in the top toolbar. On the right, the 'Sharing settings' panel is open. A purple box labeled '3. clicar para cambiar Change)' points to a 'Change...' button in the 'Who has access' section.

Drive

CREATE

My Drive

Ejercicios

Shared with Me

Starred

Recent

More

Connect Drive to your desktop

1.38 GB (9%) of 15 GB used

Manage

My Drive > Ejercicios

2. clicar para compartir (share)

1. clicar para seleccionar

Sharing settings

Link to share (only accessible by collaborators)

[//drive.google.com/folderview?id=0B48KCWFVwCIEU2ISNUk&usp=sharing](https://drive.google.com/folderview?id=0B48KCWFVwCIEU2ISNUk&usp=sharing)

Share link via:

Who has access

Private - Only the people listed below can access

Juan Quemada (you) [juan.quemada@gmail.com](mailto:juan.quemada@gmail.com) Is owner

3. clicar para cambiar Change)

# Crear un directorio público III

- ◆ Seguir pasos 1 y 2 para finalizar el cambio de permisos
  - La carpeta Ejercicios ya es pública en Internet

The screenshot shows the Google Drive web interface. In the left sidebar, the 'Ejercicios' folder is highlighted with a yellow background and a checkmark icon. In the main content area, the 'Ejercicios' folder is also highlighted with a yellow background and a checkmark icon. To the right, the 'Sharing settings' dialog is open. It shows 'Visibility options' with 'Public on the web' selected. Below this, 'Access' is set to 'Anyone (no sign-in required)' with a 'Can view' dropdown. At the bottom, there are 'Save' and 'Cancel' buttons. Two purple callout boxes with arrows point to the 'Public on the web' option and the 'Save' button.

1. clicar para seleccionar (Public on the Web)

2. clicar para guardar selección

# Acceso público con nuevo URL

- ◆ Todos los accesos anteriores **están restringidos**
  - Al usuario que es **dueño del disco** (cuenta) de Google Drive
    - ◆ Que solo puede acceder si está autenticado
- ◆ Estos accesos utilizan URLs de acceso restringido, como
  - <https://drive.google.com/?tab=Xo&authuser=0#/0B48KCWfVwCIEcEdDcEVSTS1MS00>
- ◆ Carpetas y ficheros públicos poseen además un URL público
  - Permite acceder a otros pero solo en modo **solo lectura**
    - ◆ <https://googledrive.com/host/0B48KCWfVwCIEcEdDcEVSTS1MS00/>

# Encontrar URL de acceso público

- ◆ Siguiendo los pasos 1, 2 y 3 accedemos a los detalles del directorio
  - Tal y como se muestra aquí

The screenshot shows the Google Drive web interface. In the left sidebar, the 'Ejercicios' folder is selected, indicated by a checkmark icon and a purple box with the label '1. seleccionar'. The main area shows the 'Ejercicios' folder. A purple box with the label '2. desplegar funciones adicionales' points to the 'More' button in the top toolbar. Another purple box with the label '3. clicar para ver mas detalles' points to the 'Details' option in the dropdown menu that appears after clicking 'More'.

Google Drive interface showing the 'Ejercicios' folder selected and the 'Details' menu option highlighted.

1. seleccionar

2. desplegar funciones adicionales

3. clicar para ver mas detalles



# URL de acceso público

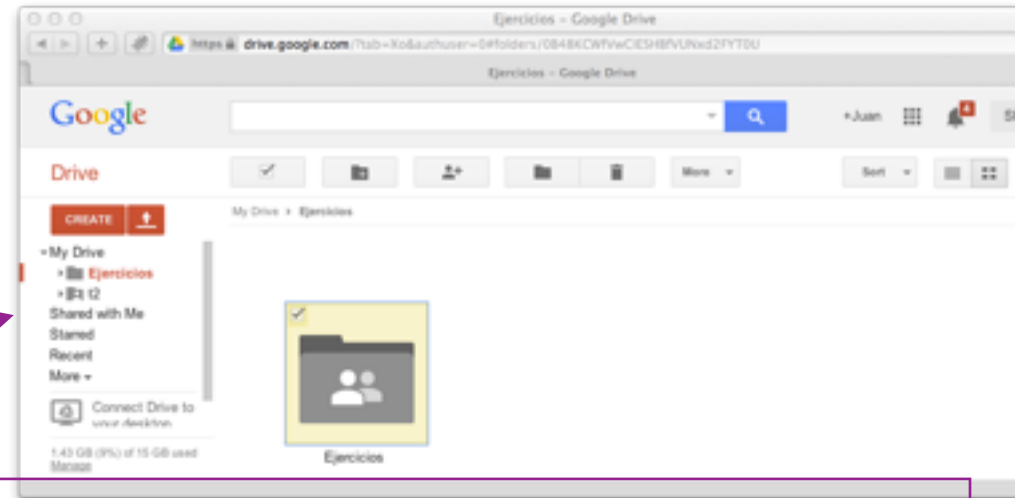
- ◆ Los detalles (details) del recurso
  - muestran el URL público de acceso en modo lectura

The screenshot shows the Google Drive web interface. At the top, there are two browser tabs: 'Recibidos (1) - juan.quemada' and 'Ejercicios - Google Drive'. The address bar shows a private URL: <https://drive.google.com/?tab=mo&authuser=0#folders/0B48KCWFvVwCIEUHjk...>. A purple arrow points from the text 'URL privado' to this address bar.

The main content area shows the 'Ejercicios' folder. A purple arrow points from the text 'URL público' to the sharing details panel on the right. The sharing details panel shows the following information:

- SHARING**
- Visibility: Public on the web
- Juan Quemada (owner)
- DESCRIPTION**
- None
- FOLDERS**
- Ejercicios
- HOSTING**
- <https://googledrive.com/host/0B48KCWFvVwCIEUHjk...>
- GENERAL INFO**
- Modified by me 6:33 pm

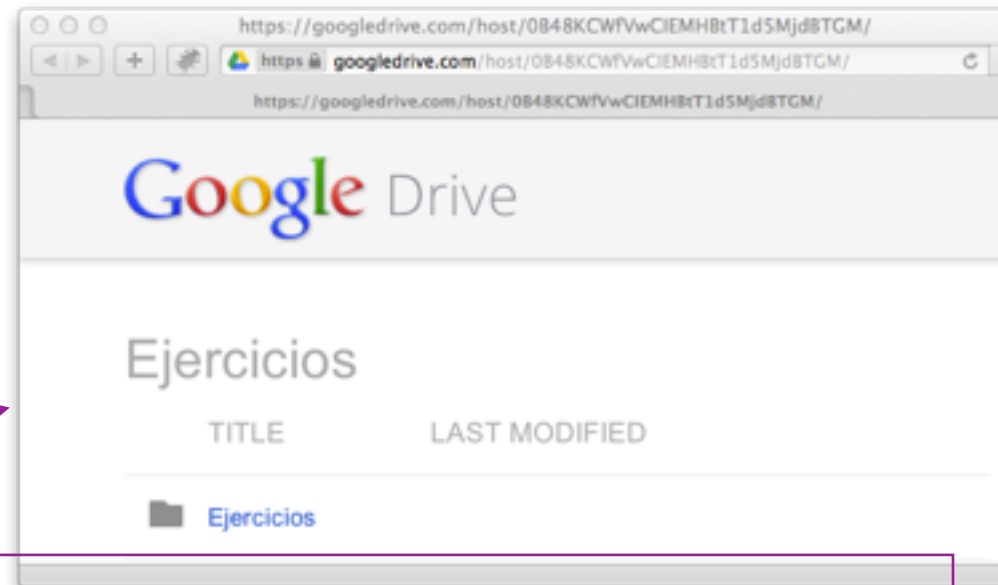
# Acceso autenticado: lectura, modificación, borrado, ..



**URL privado:**

**<https://drive.google.com/?tab=Xo&authuser=0#folders/0B48KCWfVwCIEcEdDcEVSTS1MS00>**

# Acceso público: solo lectura



**URL público:**

**<https://googledrive.com/host/0B48KCWfVwCIEcEdDcEVSTS1MS00/>**



Final del tema  
Muchas gracias!



# Entrega de ejercicios en Google Drive

# Publicación de ejercicios

- ◆ El **directorio público** creado para esta asignatura
  - será utilizado para publicar el ejercicio de la entrega semanal
- ◆ El **ejercicio** deberá subirse al **directorio público**
  - Y el **URL público** deberá incluirse en la entrega en Moodle
- ◆ El URL público debe utilizarse también para
  - Probar y ver las páginas y aplicaciones Web
    - ◆ En nuestro PC, teléfonos móviles y tabletas
- ◆ Se recomienda guardar el URL a la carpeta pública
  - En los favoritos del navegador utilizado: PC, móvil, tableta, ...
    - ◆ Para probar los ejercicios realizados en todos los dispositivos con facilidad

# Subir un ejercicio al directorio I

◆ Seleccionar **Files...** para subir fichero

The screenshot shows the Google Drive web interface. The browser address bar displays the URL: `https://drive.google.com/?tab=Xo&authuser=0#folders/0B48KCWFVwCIEMHBtT1d5MjdBTGM`. The page title is "Ejercicios - Google Drive". The Google logo is in the top left, followed by a search bar and user information "+Juan". The "Drive" header is on the left. Below it, the "CREATE" button is highlighted with a red circle. A dropdown menu is open, showing "Files..." and "Folder...". A purple arrow points from the text "1. Desplegar y clicar Files para subir fichero" to the "Files..." option. To the right, the view toggle buttons (list, grid, settings) are shown. A purple arrow points from the text "0. En modo cuadrícula" to the grid view button. The left sidebar shows the folder structure: "My Drive" > "Ejercicios" > "Ejercicios".

0. En modo cuadrícula

1. Desplegar y clicar Files para subir fichero

# Subir un ejercicio al directorio II

◆ Seguir pasos 1 y 2 para subir fichero

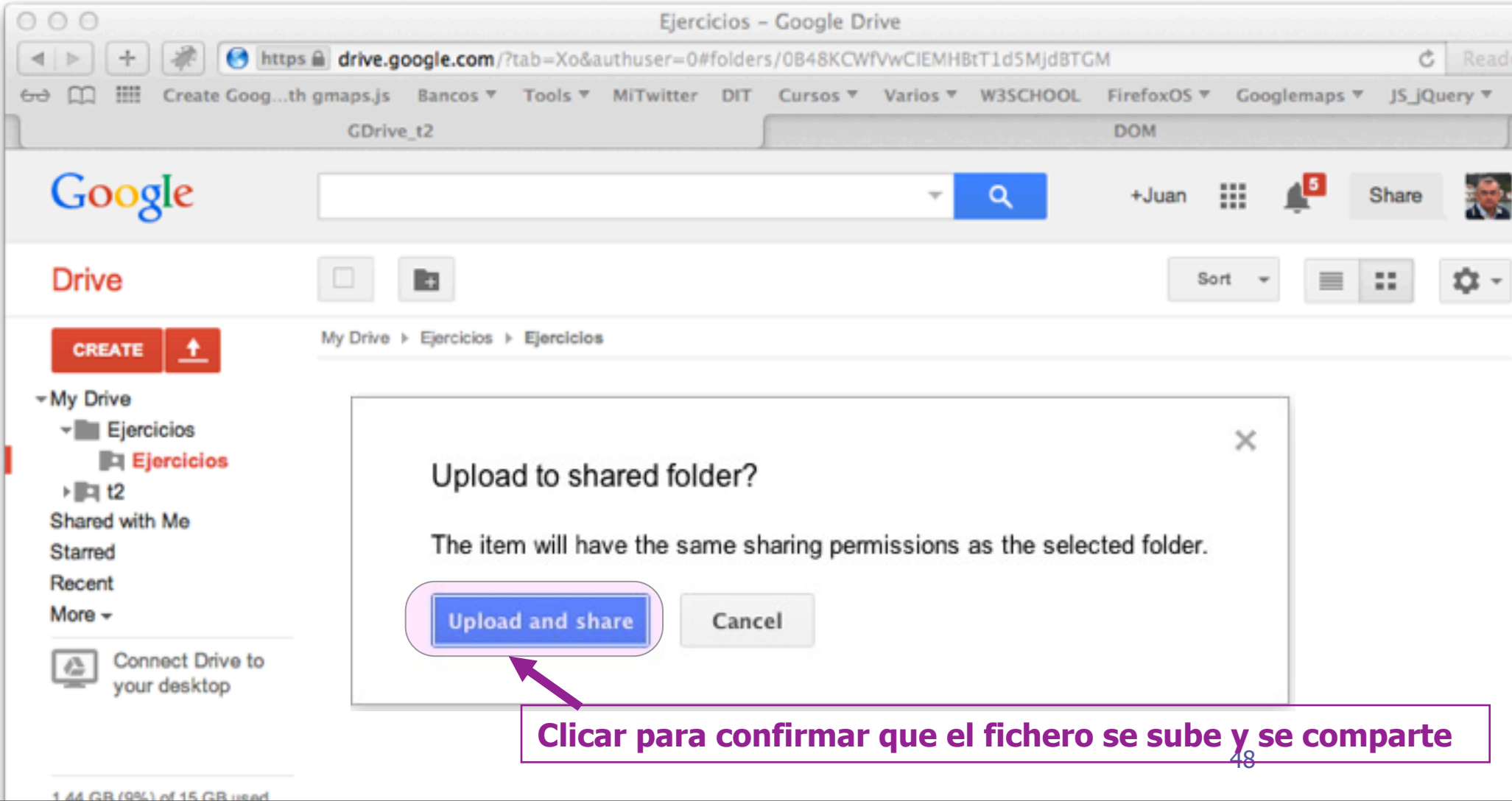
The screenshot shows the Google Drive web interface. The browser address bar displays the URL: <https://drive.google.com/?tab=Xo&authuser=0#folders/0B48KCWFVwCIEMHBtT1d5MjdBTGM>. The page title is "Ejercicios - Google Drive". The left sidebar shows the "Drive" logo, "CREATE" button, and a list of folders including "My Drive", "Ejercicios", and "t2". The main content area shows a folder named "s2-mod2\_ejemplos" containing a list of files. The file "T2\_3-2\_lista.html" is selected, highlighted with a purple box, and an arrow points to it from a text box labeled "1. clicar para seleccionar fichero". The file list table is as follows:

Name	Date Modified	Size
T2_1-1_importancia.html	Sep 5, 2013 12:24 PM	309 bytes
T2_1-2_valores_pordefecto_CSS.html	Aug 2, 2013 10:11 AM	2 KB
T2_2_herencia.html	Sep 5, 2013 1:59 PM	220 bytes
T2_3-1_br.html	Sep 11, 2013 1:41 PM	5 KB
T2_3-2_cita.html	Sep 11, 2013 12:29 PM	612 bytes
T2_3-2_lista.html	Sep 11, 2013 12:16 PM	315 bytes
T2_3-3_parrafo.html	Sep 11, 2013 12:14 PM	255 bytes
T2_3-4_pre.html	Sep 11, 2013 1:20 PM	226 bytes

At the bottom right, there are "Cancel" and "Choose" buttons. An arrow points from a text box labeled "2. clicar para subir fichero" to the "Choose" button.

# Subir un ejercicio al directorio III

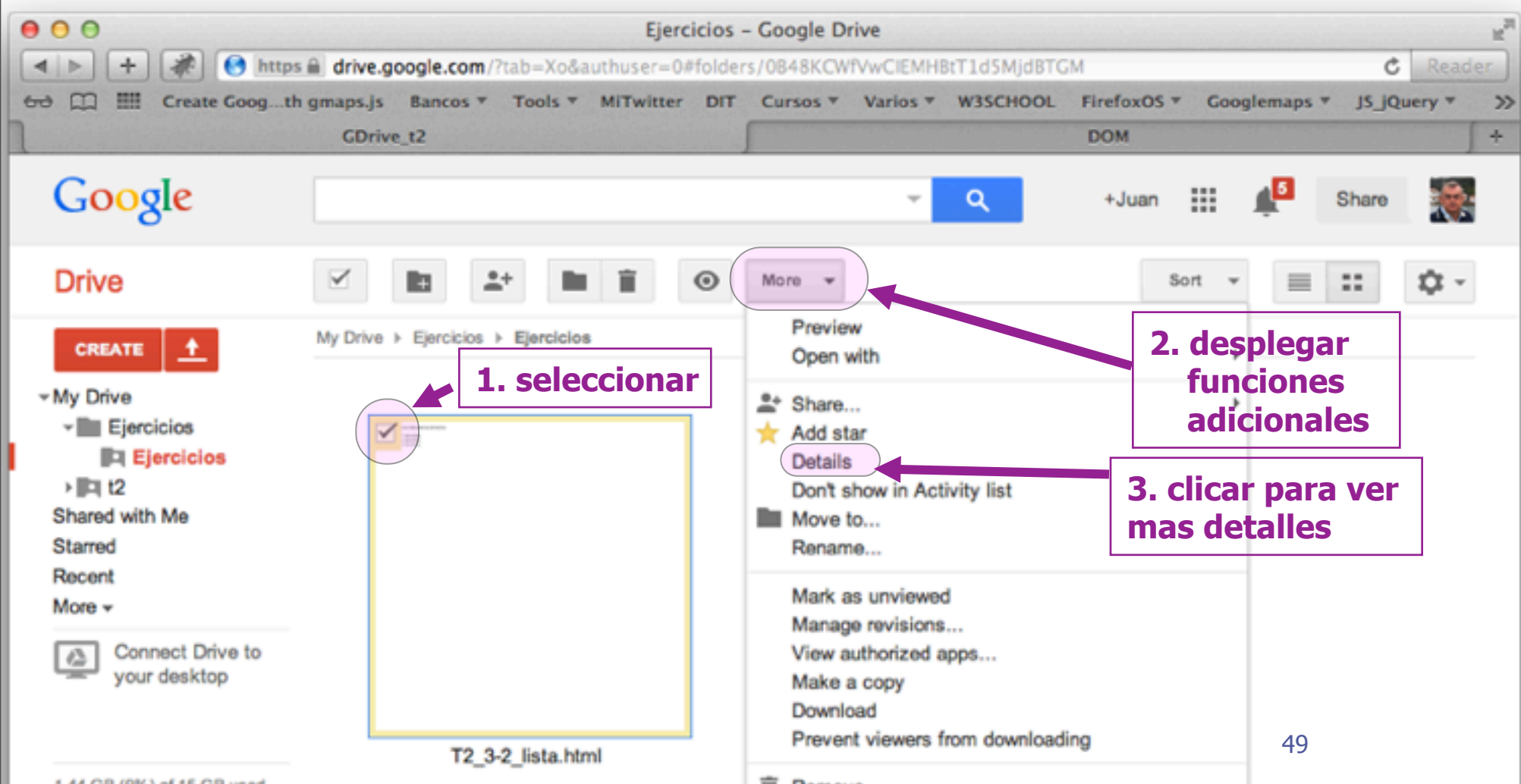
- ◆ Confirmar que el fichero se sube a un directorio público
  - El fichero también será público y será visible como pagina Web en Internet





# Encontrar el URL público I

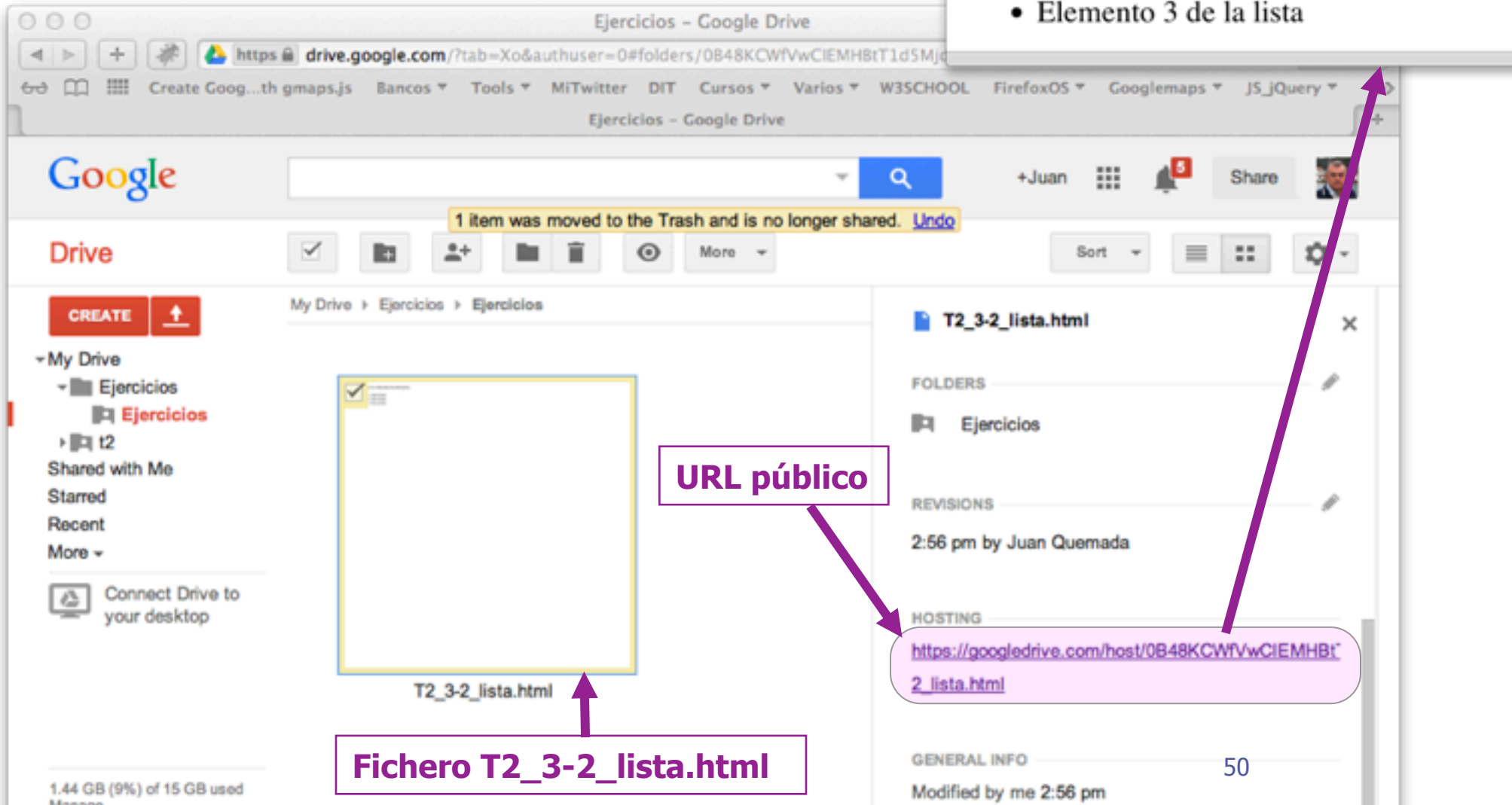
- ◆ El fichero T2\_3-2\_lista.html ya esta en directorio público
  - Pasos 1, 2 y 3 nos llevan a detalles del recurso (ver transp. siguiente)



## Encontrar el URL público II

◆ Haciendo clic en el URL público

- Vemos T2\_3-2\_lista.html como página Web





Final del tema  
Muchas gracias!

