

# **DAWE**

## **Web Storage API**

### **(almacenamiento local)**

Semana 8

LSI Saila

Juanan Pereira <juanan.pereira@ehu.eus>

# Introducción

- En el pasado: cookies
- Web Storage API
- Web Storage API. Ejemplo práctico

# En el pasado: cookies

1

Al servir la página web al navegador, el servidor le envía también una cookie con variables nombre=valor. La cookie servirá para trazar al cliente.



**Cookie:**  
pet=dog;  
age=5;  
color=black



tienda.com

El navegador guardará la cookie y la enviará al servidor cada vez que vuelva a hacer peticiones HTTP

# En el pasado: cookies

2



**Cookie:**  
pet=dog;  
age=5;  
color=black



tienda.com

Cuando el navegador solicite una nueva página a tienda.com, le reenviará la cookie que tiene de ese dominio

Las cookies se envían **en cada petición y respuesta HTTP**. Permiten almacenar datos con una **capacidad total de 4K por cookie**

# En el pasado: cookies

3

Las cookies pueden ser usadas para personalizar la página del cliente, almacenar datos, mantener el estado de un juego...

El servidor sabe, por los datos de la cookie, lo que le interesa a este cliente. En función de eso, puede personalizar la página para este cliente en concreto.



**Cookie:**  
pet=dog;  
age=5;  
color=black



tienda.com

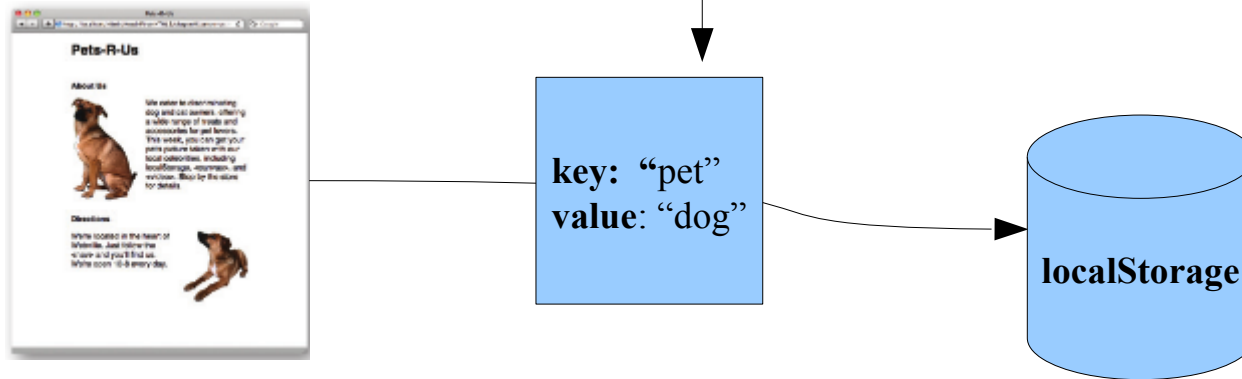
Las cookies se asocian con un dominio (en el ejemplo, tienda.com), por lo que otros dominios (competencia.com) no pueden acceder a esos datos.

# Web Storage API

- API JavaScript ofrecido por HTML5
- Permite el almacenamiento local de pares nombre=valor
- Entre 5 y 10 MB (depende del navegador) por dominio
- Acceso a los datos bajo demanda (reduciendo la necesidad de comunicación con el servidor)
- Especialmente pensado para aplicaciones RIA y aplicaciones web para móviles

# Web Storage API

Usando el API es posible guardar pares clave/valor en el repositorio de almacenamiento local



Este repositorio se encuentra almacenado en el propio navegador. Ofrece entre 5 y 10 MB (por dominio) de almacenamiento persistente (se mantiene tras cerrar el navegador)

# Web Storage API

- Desde JavaScript, es posible acceder al almacenamiento local ofrecido por HTML5 a través del objeto global `localStorage`.

```
interface Storage {  
    getter any getItem(in DOMString key);  
    setter creator void setItem(in DOMString key, in any data);  
};
```

- `localStorage` está soportado por todos los navegadores (¡incluido Internet Explorer!)



# Web Storage API

- Desde JavaScript, es posible acceder al almacenamiento local ofrecido por HTML5 a través del objeto global `localStorage`.

```
interface Storage {  
    getter any getItem(in DOMString key);  
    setter creator void setItem(in DOMString key, in any data);  
};
```

- `localStorage` está soportado por todos los navegadores (¡incluido Internet Explorer!)

# Web Storage API

En lugar de usar getItem y setItem...

```
var valor = localStorage.getItem("clave");  
// ...  
localStorage.setItem("clave", valor);
```

también se puede tratar localStorage como un array asociativo:

```
var valor = localStorage["clave"];  
// ...  
localStorage["clave"] = valor;
```

# Web Storage API

- localStorage también ofrece métodos para borrar un par clave/valor (removeItem)
- o para borrar el repositorio de pares clave/valor entero (clear)

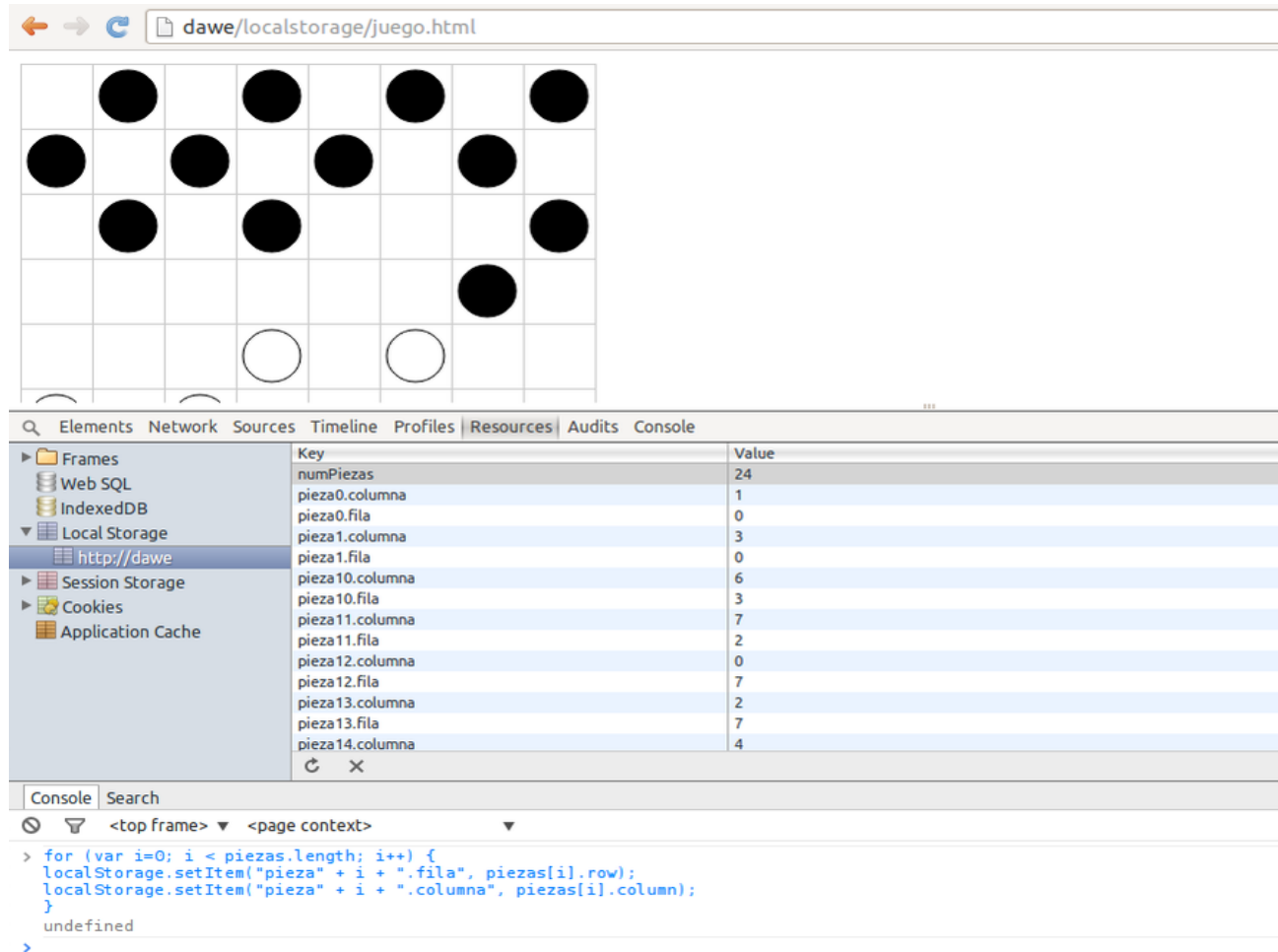
```
interface Storage {  
    delete void removeItem(in DOMString key);  
    void clear();  
};
```

# Web Storage API

- Finalmente, localStorage ofrece la propiedad length para saber cuántos pares clave/valor se almacenan
- y del método key() para iterar sobre los elementos de localStorage usando un índice

```
interface Storage {  
    readonly attribute unsigned long length;  
    getter DOMString key(in unsigned long index);  
};
```

# Web Storage API. Ejemplo de uso



The screenshot displays a web browser window with the address bar showing `dawe/localstorage/juego.html`. The main content area shows a game board, a 5x8 grid. The board contains 15 black circular pieces and 2 white circular pieces. The pieces are distributed as follows:

Row	Col 1	Col 2	Col 3	Col 4	Col 5	Col 6	Col 7	Col 8
1		Black		Black		Black		Black
2	Black		Black		Black		Black	
3		Black		Black				Black
4						Black		
5				White		White		

The developer tools are open, showing the **Local Storage** section. The storage contains 15 items, each representing a piece's position (row and column). The items are as follows:

Key	Value
numPiezas	24
pieza0.columna	1
pieza0.fila	0
pieza1.columna	3
pieza1.fila	0
pieza10.columna	6
pieza10.fila	3
pieza11.columna	7
pieza11.fila	2
pieza12.columna	0
pieza12.fila	7
pieza13.columna	2
pieza13.fila	7
pieza14.columna	4

The console shows the following JavaScript code being executed:

```
> for (var i=0; i < piezas.length; i++) {  
  localStorage.setItem("pieza" + i + ".fila", piezas[i].row);  
  localStorage.setItem("pieza" + i + ".columna", piezas[i].column);  
}  
undefined  
>
```

# Para saber más...

Libro Head First HTML5 (Ed. O'Reilly)  
(Cap. 9: Web Storage)

Libro HTML5 in Action (Ed. Manning)  
(Cap. 5: Mobile Applications: client storage and offline execution)

Libro Dive Into HTML5 (Ed. O'Reilly)  
(Cap. 7: The past, present and future of local storage for web applications)  
(de este último hay una versión libre y online <http://diveintohtml5.info/storage.html>)