

# Programa académico CAMPUS



**Trainer**  
**Ing. Carlos H. Rueda C.**



# JAVASCRIPT LOCALSTORAGE



# JAVASCRIPT LOCALSTORAGE

El objeto **Storage** (API de almacenamiento web) nos permite almacenar datos de manera local en el navegador y sin necesidad de realizar alguna conexión a una base de datos.

# JAVASCRIPT LOCALSTORAGE

**localStorage** es una propiedad que accede al objeto **Storage** y tienen la función de **almacenar datos de manera local** hasta que se decida limpiar los datos del navegador.

# JAVASCRIPT LOCALSTORAGE

LocalStorage provee los siguientes métodos y propiedades:

**setItem(clave, valor)** – almacenar clave/valor.

**getItem(clave)** – obtener el valor

**removeItem(clave)** – eliminar la clave y valor.

**clear()** – borrar todo.

**key(índice)** – obtener la clave de una posición dada.

**length** – el número de ítems almacenados.

# JAVASCRIPT LOCALSTORAGE

## Validar objeto Storage en el navegador

Los navegadores actuales soportan el objeto Storage, pero no está de más validar que realmente se pueda usar.

```
if (typeof(Storage) !==  
'undefined') {  
    // Código cuando Storage es  
compatible  
} else {  
    // Código cuando Storage NO es  
compatible  
}
```

# JAVASCRIPT LOCALSTORAGE

## Guardar datos

Hay dos formas de guardar datos:

- Acceso tipo Mapa
- Acceso tipo objeto

```
//acceso tipo mapa  
localStorage.setItem( 'Nombre',  
  'Miguel Antonio')  
  
//acceso tipo objeto  
localStorage.Apellido = 'Márquez  
Montoya'
```

# JAVASCRIPT LOCALSTORAGE

## Recuperar datos

Hay dos formas de recuperar datos:

- Acceso tipo Mapa
- Acceso tipo objeto

```
// Op 1 -> localStorage.getItem(name, content)
// Op 2 -> localStorage.name

// Obtenemos los datos almacenados
let firstName = localStorage.getItem("Nombre"),
let lastName = localStorage.Apellido;

console.log(`Hola, mi nombre es ${firstName}
${lastName}`);
// Imprime: Hola, mi nombre es Miguel Antonio
Márquez Montoya
```



# JAVASCRIPT LOCALSTORAGE

## Eliminar datos

Para eliminar hacemos:

```
// localStorage.removeItem(name)  
localStorage.removeItem(Apellido)
```

# JAVASCRIPT LOCALSTORAGE

**Limpiar todos los  
datos**

Dejar todo el objeto limpio

```
// localStorage.clear()  
localStorage.clear()
```

# JAVASCRIPT LOCALSTORAGE

## Iterando el LocalStorage

Una opción es utilizar iteración sobre un array:

```
/*Obtener datos almacenados*/  
for(let i=0; i<localStorage.length; i++) {  
    let key = localStorage.key(i);  
    alert(`${key};  
    ${localStorage.getItem(key)}`);  
}
```

# JAVASCRIPT LOCALSTORAGE

## Iterando el LocalStorage

La forma anterior devuelve campos propios del localStorage que no necesitamos. Para solucionar esto podríamos filtrar los campos con `hasOwnProperty`

```
for(let key in localStorage) {  
  if (!localStorage.hasOwnProperty(key)) {  
    // se salta claves como "setItem", "getItem"  
    continue;  
  }  
  alert(`${key}: ${localStorage.getItem(key)}`);  
}
```

# JAVASCRIPT LOCALSTORAGE

## Iterando el LocalStorage

O podemos acceder a las "claves" propias con **Object.Key**.

```
let keys = Object.keys(localStorage);
for(let key of keys) {
    alert(`${key}: ${localStorage.getItem(key)}`);
}
```

# JAVASCRIPT LOCALSTORAGE

## Ejercicio

## Ejemplo - localStorage

---

Nombre almacenado:

Apellido almacenado:

# JAVASCRIPT LOCALSTORAGE

## Ejercicio

```
const myObject = {  
  name : "john doe",  
  age : 32,  
  gender : "male",  
  profession : "optician"  
}  
  
localStorage.setItem("myObject",  
  JSON.stringify(myObject));  
  
let newObject =  
  window.localStorage.getItem("myObject");  
console.log(JSON.parse(newObject));
```

# JAVASCRIPT SESSIONSTORAGE

SessionStorage almacena información mientras la pestaña donde se esté utilizando siga abierta, una vez cerrada, la información se elimina.



# JAVASCRIPT SESSIONSTORAGE

Las propiedades y métodos son los mismos, pero es mucho más limitado:

- ✓ **sessionStorage** solo existe dentro de la pestaña actual del navegador.
  - Otra pestaña con la misma página tendrá un almacenaje distinto.
- ✓ Los datos sobreviven un refresco de página, pero no cerrar/abrir la pestaña.

# JAVASCRIPT SESSIONSTORAGE

## Ejercicio

## Ejemplo - sessionStorage

---

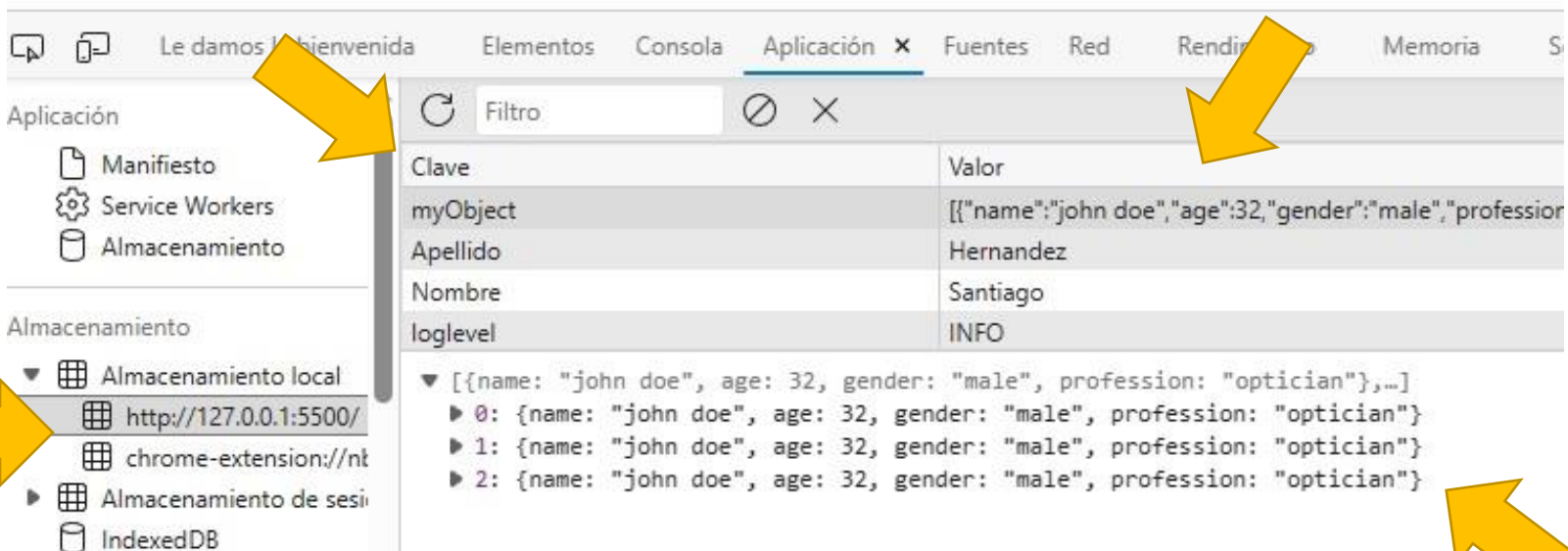
Nombre almacenado:

Apellido almacenado:

# JAVASCRIPT SESSIONSTORAGE

## Pestaña Aplicación

### Ejemplo localStorage



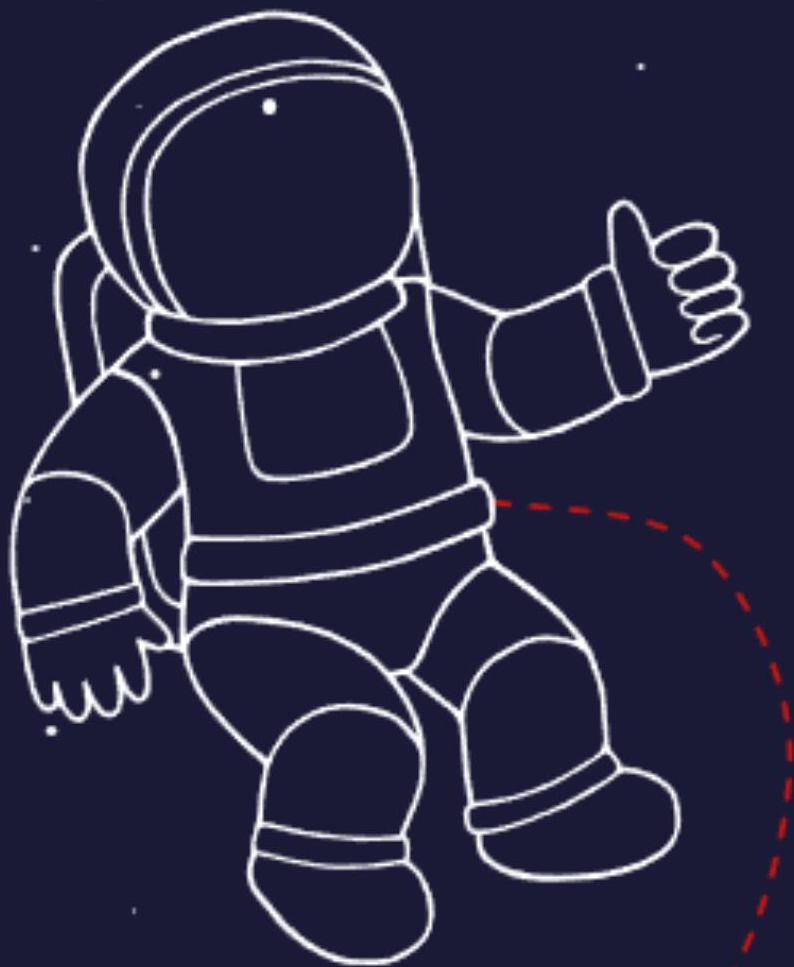
The screenshot shows the Chrome DevTools 'Aplicación' (Application) tab. The left sidebar has a tree view with 'Almacenamiento local' (Local Storage) selected. The main pane displays a table of stored items and a detailed view of the first item.

Clave	Valor
myObject	[{"name": "john doe", "age": 32, "gender": "male", "profession": "optician"}, ...]
Apellido	Hernandez
Nombre	Santiago
loglevel	INFO

▼ [{"name": "john doe", "age": 32, "gender": "male", "profession": "optician"}, ...]

- ▶ 0: {"name": "john doe", "age": 32, "gender": "male", "profession": "optician"}
- ▶ 1: {"name": "john doe", "age": 32, "gender": "male", "profession": "optician"}
- ▶ 2: {"name": "john doe", "age": 32, "gender": "male", "profession": "optician"}



# Programa académico CAMPUS



**Trainer**  
**Ing. Carlos H. Rueda C.**

