

# UF3 – Lista definitiva para el examen (2º DAW)

## 1. DOM

- Selección: getElementById, querySelector, querySelectorAll
- children (solo etiquetas) vs childNodes (todo)
- textContent para leer y escribir
- createElement, appendChild, insertBefore
- remove / removeChild
- appendChild mueve nodos
- cloneNode(true) copia nodos
- Colecciones vivas (children) y estáticas (querySelectorAll)

## 2. Eventos

- addEventListener pasa una función, no la ejecuta
- Objeto event (e) lo crea el navegador
- event.target vs event.currentTarget
- Bubbling (propagación)
- Delegación de eventos en elementos padre
- preventDefault evita el comportamiento por defecto
- stopPropagation detiene la propagación

## 3. Formularios

- document.forms y form.elements
- value para inputs de texto
- checked para checkbox y radio
- Eventos: input y submit
- submit siempre con preventDefault
- El submit pertenece al form, no al botón
- button sin type es submit por defecto

## 4. Template

- La etiqueta no se renderiza sola
- Se usa content.cloneNode(true)

## 5. Web Components

- Etiqueta HTML personalizada
- Clase que extiende HTMLElement
- Registro con customElements.define
- El nombre debe llevar guion

## 6. Geolocation

- navigator.geolocation
- getCurrentPosition obtiene la posición una vez
- watchPosition hace seguimiento
- Requiere permiso y HTTPS

## 7. Storage

- localStorage (permanente)
- sessionStorage (se borra al cerrar)
- setItem, getItem, removeItem, clear
- Storage solo guarda strings
- Objetos con JSON.stringify y JSON.parse

# UF3 – Ejemplos de código mínimos (examen)

## DOM – Selección

```
const caja = document.getElementById("caja");
const item = document.querySelector("li");
const items = document.querySelectorAll("li");
```

## DOM – Crear / Insertar / Borrar

```
const li = document.createElement("li");
li.textContent = "Nuevo";
lista.appendChild(li);

li.remove();
```

## DOM – Copiar vs Mover

```
lista2.appendChild(lista1.children[0]); // mueve

const copia = li.cloneNode(true); // copia
```

## Eventos – click y target

```
ul.addEventListener("click", function(e){
  console.log(e.target.textContent);
});
```

## Eventos – submit y preventDefault

```
form.addEventListener("submit", function(e){
  e.preventDefault();
});
```

## Formularios – document.forms

```
const f = document.forms[0];
f.elements["email"].value;
```

## Formularios – checked (radio / checkbox)

```
if (f.elements["acepta"].checked) {
  console.log("Aceptado");
}
```

## Template

```
const tpl = document.getElementById("tpl");
const clon = tpl.content.cloneNode(true);
document.body.appendChild(clon);
```

## Web Components

```
class MiElemento extends HTMLElement {
  connectedCallback(){
    this.textContent = "Hola";
  }
}
customElements.define("mi-elemento", MiElemento);
```

## Geolocation

```
navigator.geolocation.getCurrentPosition(function(pos){  
  console.log(pos.coords.latitude);  
});
```

## Storage

```
localStorage.setItem("nombre", "Ana");  
const n = localStorage.getItem("nombre");
```

## Storage – Objetos

```
localStorage.setItem("user", JSON.stringify(obj));  
const user = JSON.parse(localStorage.getItem("user"));
```