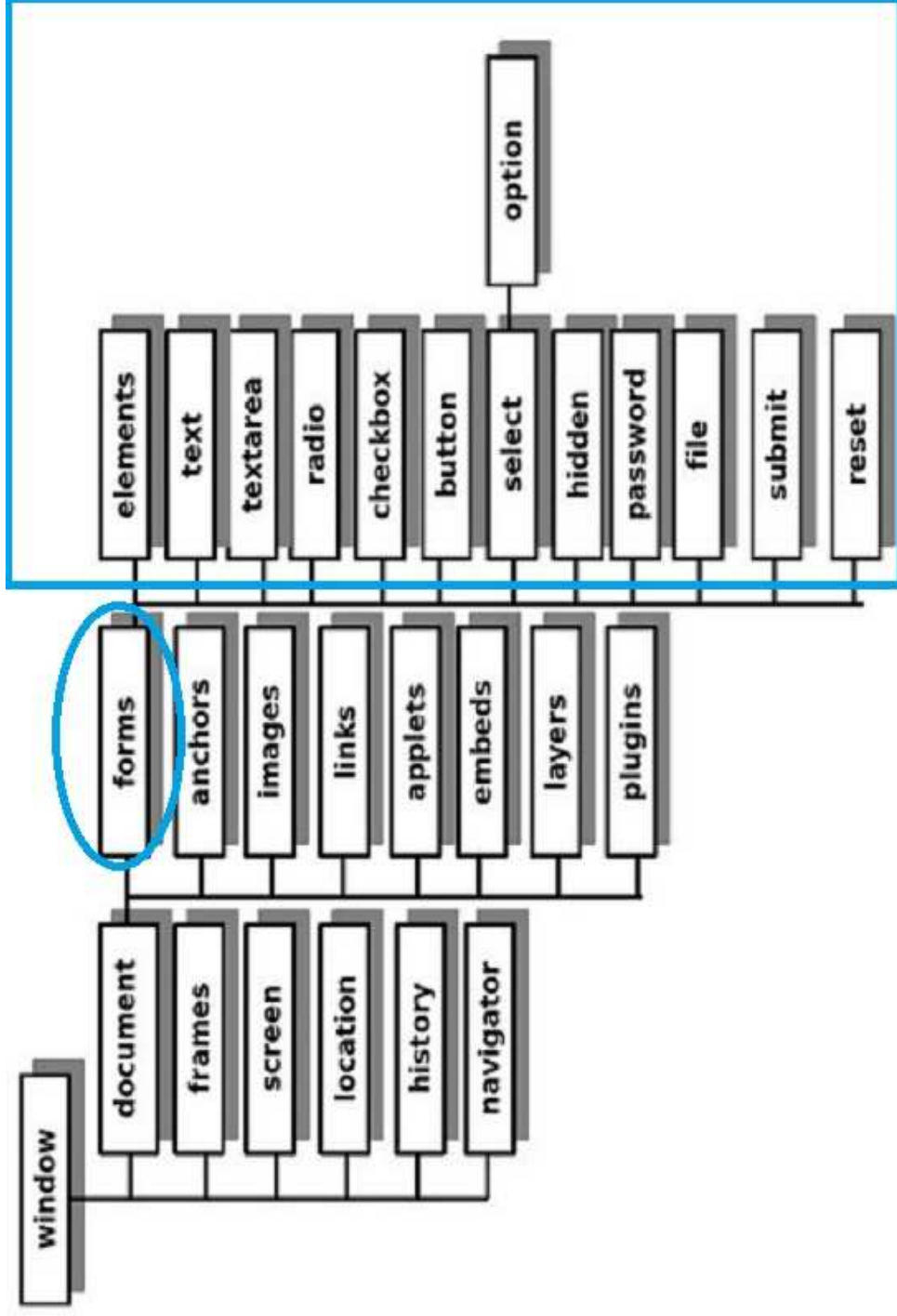
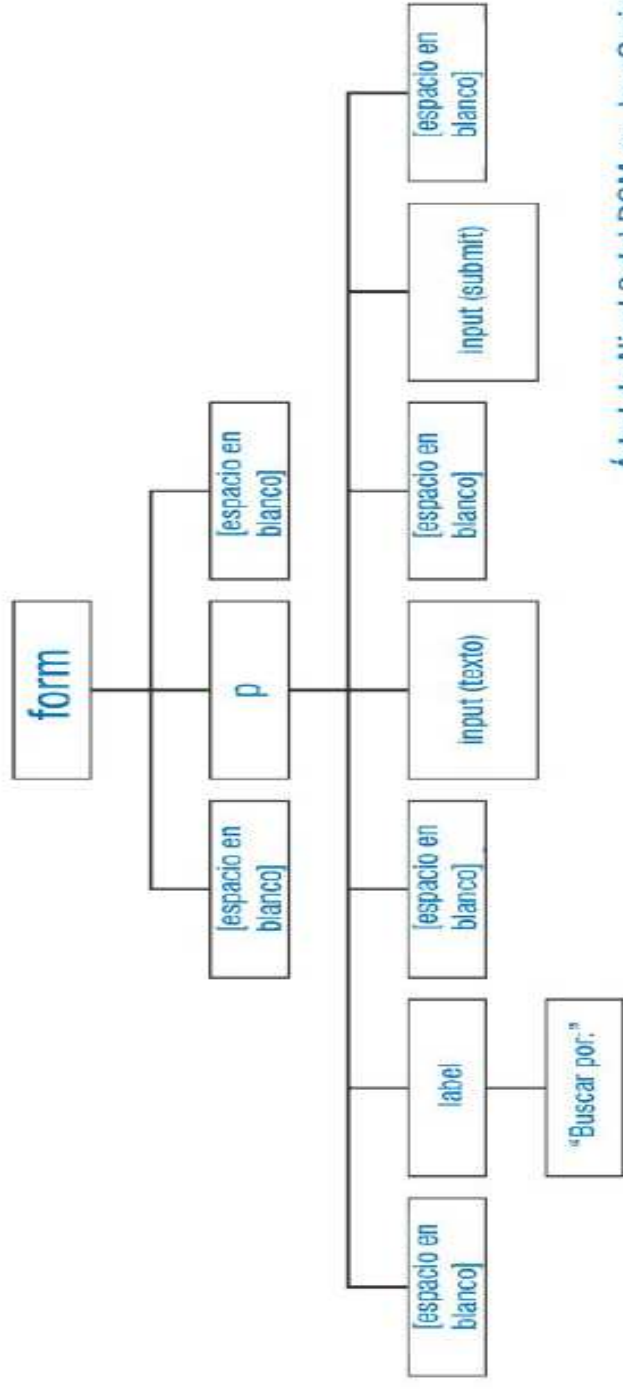




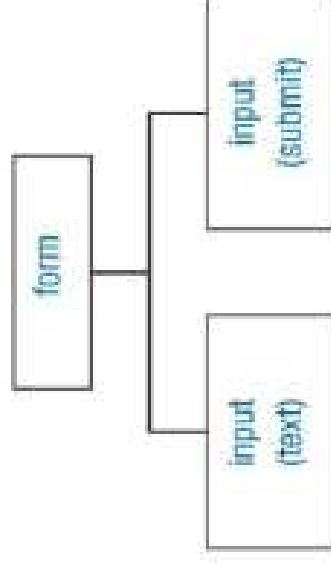
## 2.- Formularios

- Un formulario web sirve para enviar, tratar y recuperar datos que son enviados y recibidos entre un cliente y un servidor web.
- En JavaScript el objeto form depende en la jerarquía del objeto document.
- JavaScript dispone de numerosas propiedades y funciones que facilitan la programación de aplicaciones que manejan formularios.





Árbol de Nivel 2 del DOM en JavaScript



## 2.- Formularios

- Estructura de un formulario:
  - Los formularios se definen con etiquetas.
  - La etiqueta principal es `<form> </form>`.
  - Para que sea funcional, la etiqueta `<form>` necesita inicializar dos atributos:
    - `action` – Contiene la URL donde se redirigen los datos del formulario.
    - `method` – Indica el método por el cual el formulario envía los datos. Puede ser `POST` o `GET`.

## 2.- Formularios

- Ejemplo:

```
<html>
  <head><title>Ejemplo de formulario</title>
</head>
  <body>
    <h3>Formulario</h3>
    <form action="www.web.es/formulario.php"
      method="post">
      ...
    </form>
  </body>
</html>
```

## 2.- Formularios

- Cuando se carga una página web, el navegador crea automáticamente un array llamado `forms`, que contiene la referencia a todos los formularios de la página.
- Para acceder al array `forms`, se utiliza el objeto `document`, por lo que `document.forms` es el array que contiene todos los formularios de la página.
- Para acceder al primer formulario de la página:

```
document.forms[0];
```

## 2.- Formularios

- Además del array de formularios, el navegador crea automáticamente un array llamado `elements`, que contiene la referencia a todos los elementos (cuadros de texto, botones, listas desplegables, etc.) de ese formulario.
- Cada formulario de la página tendrá su array `elements` que nos permitirá acceder a sus elementos.
- Para acceder al primer elemento del primer formulario de la página:

```
document.forms[0].elements[0];
```

- Para acceder al ultimo elemento del primer formulario de la página:

```
document.forms[0].elements[document.forms[0].elements.length-1];
```

## 2.- Formularios

- Con el método de acceso a formularios anterior, si cambia el diseño de la página y el orden de los formularios, tendríamos que cambiar el código y el mantenimiento resulta tedioso.
- Accederemos a los formularios de una página a través de su nombre (atributo name) o a través de su atributo id.

```
var formPrincipal = document.mi_formulario;  
var formSecundario = document.otro_formulario;  
...
```

```
<form name="mi_formulario">  
</form>  
<form name="otro_formulario">  
</form>
```



## 2.- Formularios

- A través del método `getElementById()` del DOM nos permite acceder a un objeto a través de su atributo ID.

```
var fPri = document.getElementById("mi_formulario");  
var apellid = document.getElementById("apellidos");
```

```
...
```

```
<form id="mi_formulario">  
  <input type="text" id="apellidos" />  
</form>
```

## 2.- Formularios

- En resumen, para acceder a cualquier objeto dentro de nuestro documento o formulario:

```
document.getElementById("id-control");
```

```
document.nombreFormulario.name-del-control;
```

```
<form id="formBusqueda" action="">
  <input type="text" id="entrada" name="cEntrada" />
  <input type="submit" id="enviar" name="enviar" />
</form>
```

- Las siguientes referencias al campo de texto entrada, serán todas válidas:

```
document.getElementById("entrada");
document.formBusqueda.cEntrada;
document.formBusqueda.elements[0];
document.forms["formBusqueda"].cEntrada;
document.forms["formBusqueda"].elements["cEntrada"];
```

## 2.- Formularios: Elementos

- Cuadro de texto y textarea:
  - El valor de texto mostrado por estos elementos se obtiene y se establece mediante la propiedad `value`.

```
<input type="text" id="texto" />  
var valor = document.getElementById("texto").value;
```

```
<textarea id="parrafo"></textarea>  
var txt = document.getElementById("parrafo").value;
```

## 2.- Formularios: Elementos

- **Checkbox:**

- Para saber si un campo de tipo checkbox está o no marcado, disponemos de la propiedad `checked`.

```
<p>Colores favoritos</p>  
<input type="checkbox" id="rojo" name="rojo" value="ro"/> Rojo  
<input type="checkbox" id="azul" name="azul" value="az"/> Azul  
<input type="checkbox" id="verde" name="verde" value="ve"/> Verde
```

Colores favoritos

☐ Rojo

☐ Azul

☐ Verde

- Para marcar un elemento mediante código:

```
document.getElementById("verde").checked = true;
```

## 2.- Formularios: Elementos

- Radiobutton:

- Debemos asignar el mismo atributo `name` a cada uno de los botones del grupo.
- Lo importante es conocer cuál de todos los radiobuttons se ha seleccionado mediante la propiedad `checked`, la cuál devuelve `true` para el radiobutton seleccionado y `false` en cualquier otro caso.

Género

```
</br><input type="radio" name="genero" value="M"> Hombre  
</br><input type="radio" name="genero" value="F"> Mujer
```

Género

☐ Hombre

☐ Mujer

## Actividad 4



Haciendo uso de un formulario, realiza un cuestionario al usuario para que seleccione su actor favorito de entre tres facilitados cualesquiera.

## 2.- Formularios: Elementos

- Select (listas desplegables):
  - En general, lo que se requiere es obtener el valor del atributo `value` de la opción seleccionada por el usuario.

```
<select id="opciones" name="opciones">  
  <option value="1">Primer valor</option>  
  <option value="2">Segundo valor</option>  
  <option value="3">Tercero valor</option>  
</select>
```

- Para poder obtener el valor seleccionado se utilizarán las siguientes propiedades:

## 2.- Formularios: Elementos

- Select (listas desplegables):
  - `options`, es un array creado automáticamente por el navegador para cada lista desplegable y que contiene la referencia a todas las opciones de esa lista.
  - Para acceder a la primera opción de una lista:

```
document.getElementById("id-lista").options[0]
```

- `selectedIndex`, devuelve el índice de la opción seleccionada.

```
document.getElementById("id-lista").selectedIndex;
```

(El valor de `selectedIndex` para la primera opción de la lista será 0.)



## Actividad 5



Utiliza una lista desplegable para que el usuario elija una provincia de entre cuatro y muestra mediante un mensaje de aviso la opción seleccionada.