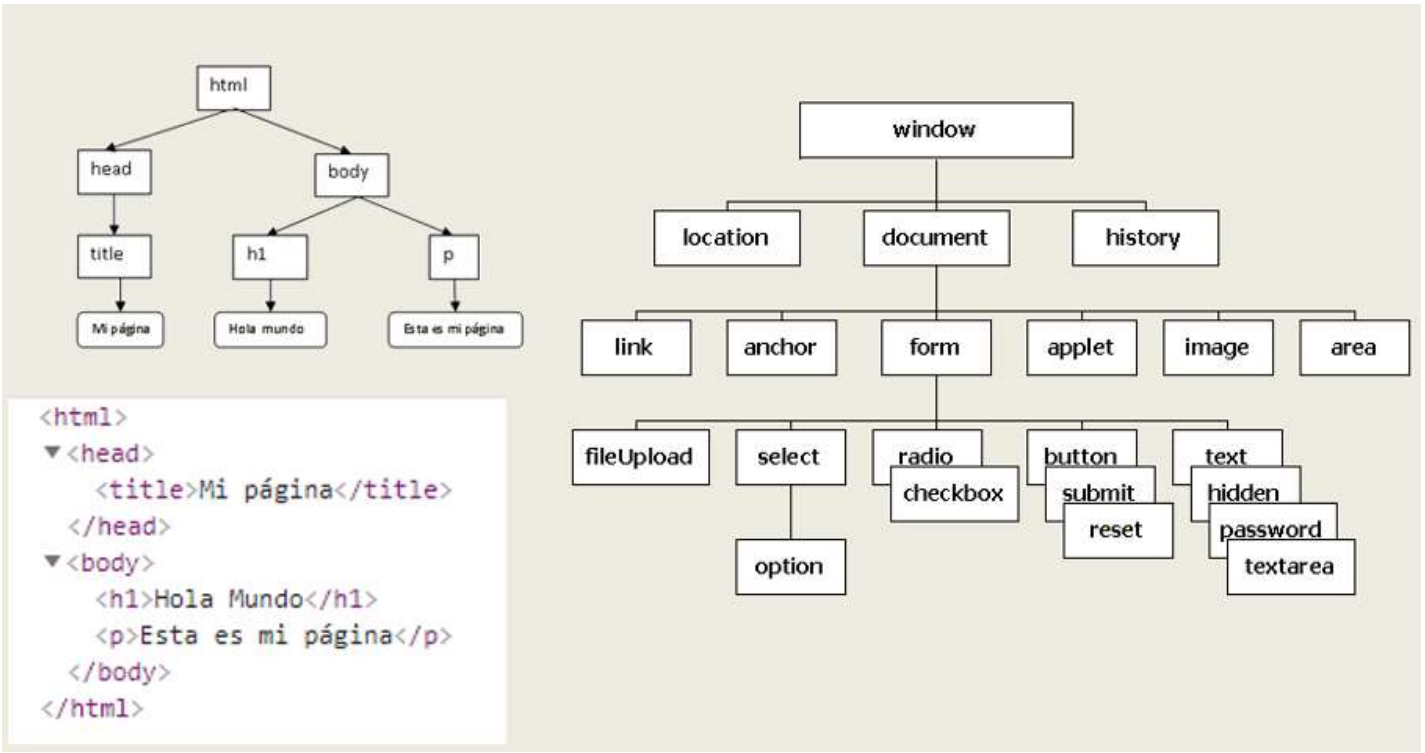
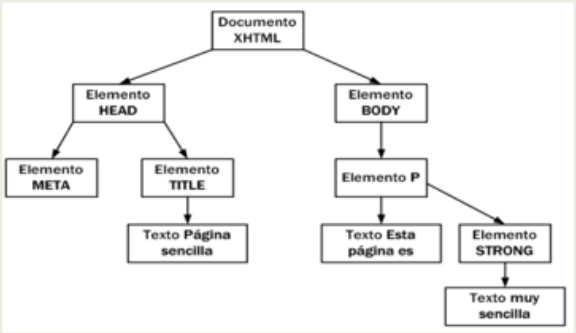
# Unidad de Trabajo 6: Manipulación del DOM

| OBJETIVOS DE LA ACTIVIDAD: |
| --- |
| • DOM y navegadores<br>• Tipos de nodos<br>• Recuperación de elementos |

## Análisis del DOM en los navegadores



```
<!DOCTYPE html>
<html lang="es">
<head>
   <meta charset="utf-8" />
   <title>Página sencilla</title>
</head>
<body>
   <p>Esta página es <strong>muy sencilla</strong></p>
</body>
</html>
```




El DOM representa HTML como una estructura de árbol de etiquetas.

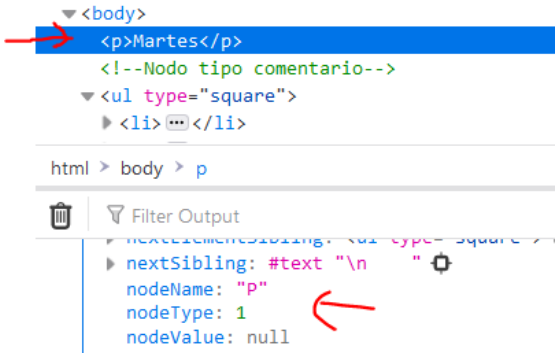
Si el documento HTML está mal escrito, por ejemplo faltan las etiquetas de definición del documento (<head>, <body> ….), el navegador realiza correcciones de forma automática al construir el DOM.

**a)** Analiza en las herramientas para desarrolladores la generación del DOM para los siguientes ejemplos:

| | | |
|---|---|---|
| `<p>Martes`<br>`<li>AWDC`<br>`<li>AWDC`<br>`<li>AWDS`<br>`<li>AWDS`<br>`<li>AWDS` | `<table id="table">`<br>`  <tr>`<br>`  <td>Martes</td>`<br>`  </tr>`<br>`</table>` | `<p>Martes`<br>`<li>AWDC`<br>`<li>AWDC`<br>`<li>AWDS`<br>`<li>AWDS`<br>`<li>AWDS`<br>`<table id="table">`<br>`  <tr>`<br>`  <td>Martes</td>`<br>`  </tr>`<br>`</table>` |

Puedes probar también la herramienta: [Live DOM Viewer (hixie.ch)]

b) Tipos de nodos




Identifica distintos tipos de nodos en el siguiente ejemplo:

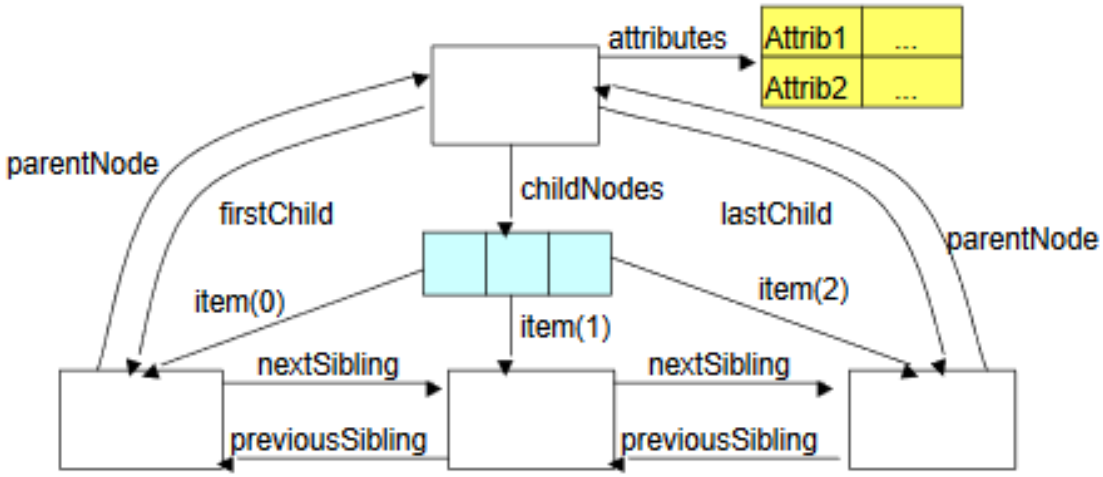
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Tipos de nodos</title>
</head>
<body>
    <p>Martes</p>
    <!-- Nodo tipo comentario -->
    <ul type="square">
        <li>AWDC
            <li>AWDC
            <li>AWDS
            <li>AWDS
            <li>AWDS
    </ul>
</body>
</html>
```

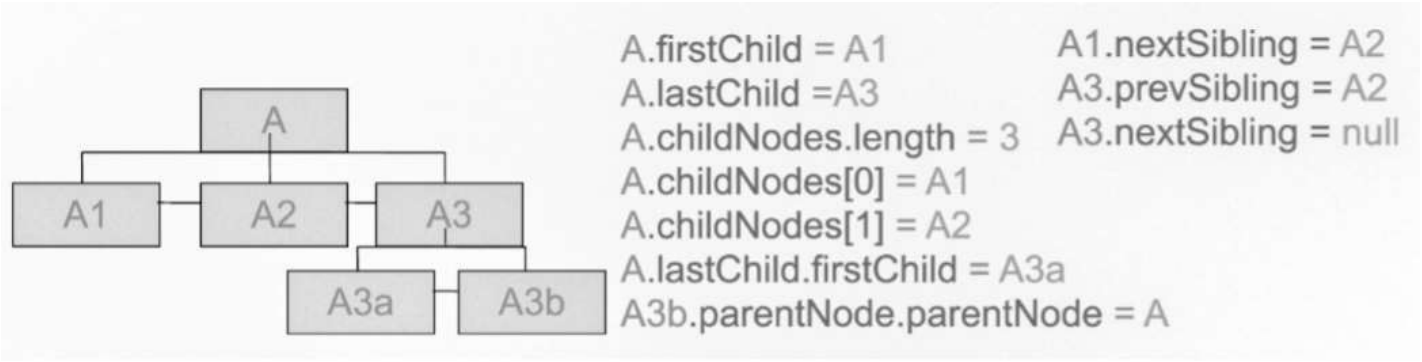
c) Analiza el siguiente script e indica las diferencias principales entre NodeList y HTMLCollection.

```
<!DOCTYPE HTML>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <title>Ejemplo DOM</title>
</head>
<body>
<div>children -> HTMLCollection</div>
<div>childNodes -> NodeList</div>
<script>
    const mostrarContenido = (lista, encabezado) => {
      console.log("\nContenido de "+encabezado+ " (length="+lista.length+"):");
```

```
      for(let i=0; i<lista.length; i++) {
        console.log("  "+i+":"+lista[i]);
      }
    }
    const elementos = document.body.children; //HTMLCollection
    const nodos = document.body.childNodes; //NodeList
    mostrarContenido(elementos, "children [HTMLCollection]");
    mostrarContenido(nodos, "childNodes [NodeList]");
</script>
</body>
</html>
```

Recorrido por el DOM:




Analiza el siguiente código e indica la diferencia entre el resultado obtenido por children y childNodes.

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Recorriendo el DOM</title>
</head>
<body>
```

```
<ul id="ul">
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
</ul>
<script>
    console.log("document.body.children");
    for (let elemento of document.body.children) {
        console.log(`Elemento:${elemento.nodeType}, ${elemento.nodeName}, ${elemento.nodeValue}`);
    }
    console.log(`total elementos con children: ${document.body.children.length}`);
    console.log("document.body.childNodes");
    for (let elemento of document.body.childNodes) {
        console.log(`Elemento:${elemento.nodeType}, ${elemento.nodeName}, ${elemento.nodeValue}`);
    }
    console.log(`total elementos con childNodes: ${document.body.childNodes.length}`);
</script>
</html>
```