

DESARROLLO WEB EN ENTORNO CLIENTE

JAVASCRIPT

DOM EJERCICIOS 2.0

ISAAC SOLER FORES

INTRODUCCIÓN

Esta segunda práctica corresponde al tema 5 del manual que estamos utilizando como guía. En el manual se ve una introducción al DOM pero os adjunto unos apuntes para complementar el tema. Leed primero el manual del curso que es más básico, luego los apuntes que he subido para este tema y cuando los conceptos estén claros realizáis los ejercicios. Se supone para esta práctica y las siguientes un conocimiento mínimo de HTML.

PRACTICA 1- DOM

En las prácticas anteriores hemos visto el objeto document, que es el objeto de más alto nivel dentro del modelo de objetos del documento(DOM), y ¿Qué es el DOM? Es un estándar del W3C que define como accedemos por a los documentos, por ejemplo en HTML y XML. Es una interfaz de programación de aplicaciones (API) que permite a los programas y scripts acceder y actualizar **dinámicamente** su contenido, estructura y estilo. El modelo DOM HTML se define a través de una estructura jerárquica en forma de árbol. Una vez se ha cargado una página, el navegador convierte automáticamente toda la página en esta estructura de árbol formada por elementos a los que se les llama nodos. ¿Cómo podemos acceder a estos nodos? Jerárquicamente o utilizando funciones de acceso directo, que es como se accede habitualmente. A parte de acceder a los nodos también es posible manipularlos, es decir yo puedo en un momento dado, crear un nodo y agregarlo a una página Web que tengo creada. Os pongo un ejemplo: Tenemos una página web con un párrafo inicial (la página web está diseñada estáticamente, es decir, yo sé que va un párrafo en la página con su texto correspondiente) y la creo en HTML de esta forma:

```
<HEAD>
  <TITLE>funciones dom</TITLE>
  <script>
    .....
  </script>
<BODY>
  <div id="ej-aCh">
    <p> Este es mi primer parrafo</p>
  </div>
```

Pero queremos que en un momento dado, el usuario pueda crear un nuevo párrafo y agregarlo al documento HTML que tenemos creado. Crearemos una función dentro del Script, utilizando funciones del DOM, del tipo:

```
function anadir_parrafo(){
  var parrafo = document.createElement('p');
  var texto = document.createTextNode('Párrafo añadido.');
```

parrafo.appendChild(texto);

document.getElementById('ej-aCh').appendChild(parrafo);

```
}
```

A la que llamaríamos, por ejemplo, al pinchar(evento onclick) sobre un botón:

```
<INPUT TYPE="button" VALUE="Añadir nuevo parrafo"
onClick="anadir_parrafo();">
```

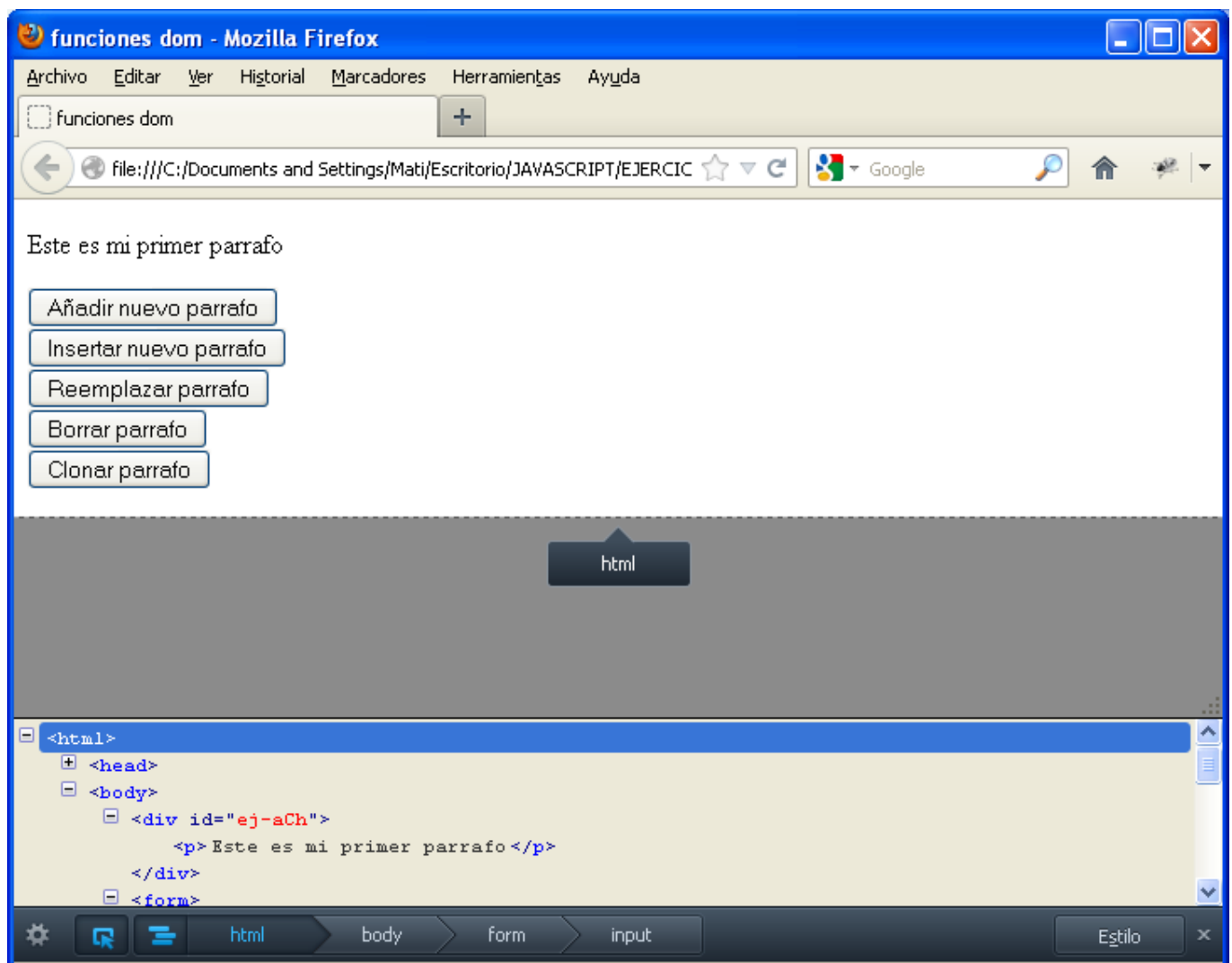
Cuando el usuario pinche sobre este botón, la página web ya no tendrá solamente el párrafo inicial que hemos creado, si no que tendrá tantos párrafos añadidos como veces el usuario haya pulsado el botón.

Una cosa interesante que os puede ayudar a entender el funcionamiento del DOM es utilizar la herramienta del navegador para ver como inicialmente es una página y como es después de realizar una evento con funciones DOM.

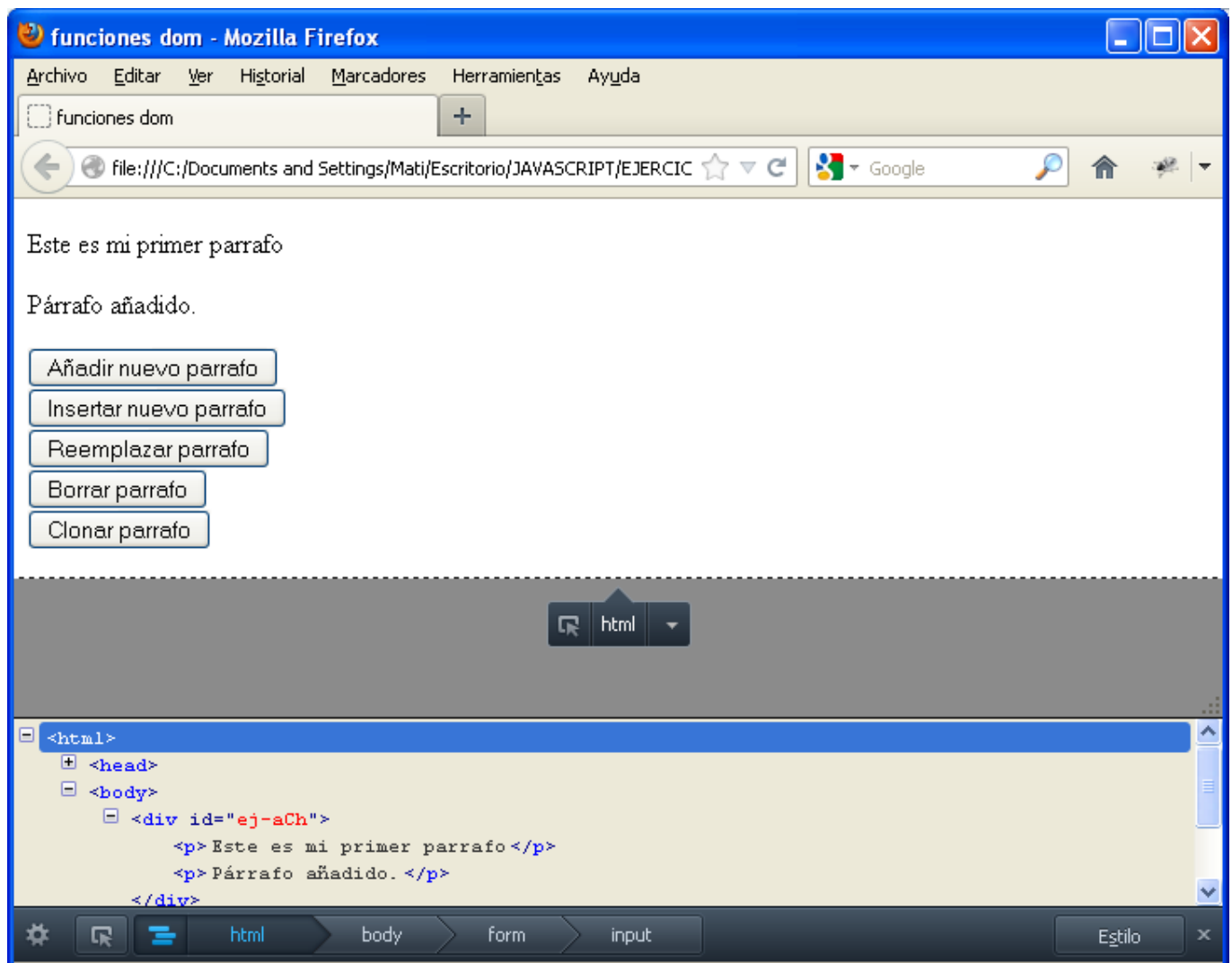
Las capturas de las pantallas que os adjunto son utilizando la herramienta inspeccionar del Menú Herramientas del Firefox (si utilizáis otros navegadores sería buscar la opción similar):

Herramientas > Desarrollador Web > Inspeccionar

Adjunto pantalla previa a pinchar sobre el botón Añadir nuevo párrafo. Mirar en la parte inferior de la pantalla y veréis que sólo aparece el párrafo creado en el código inicial de la página:



Después de pulsar sobre el botón Añadir nuevo párrafo la estructura de la página cambia y se añade el párrafo que he creado en ejecución.



Un comentario, veréis varios ejemplos con InnerHTML. Es una propiedad de Javascript, no forma parte actualmente de la especificación DOM pero, no obstante, la soportan la mayoría de navegadores. Suele usarse con métodos DOM para cambiar el contenido de un elemento

Un ejemplo:

Dentro de un documento HTML tenemos el siguiente div :

```
<div id="párrafo"></div>
```

Y utilizando la función getElementById pondremos dentro de div lo siguiente:

```
document.getElementById('párrafo').innerHTML="Saludos a todos";
```

Lo que estoy diciendo con este código es que acceda al elemento div del documento y ponga dentro del mismo "Saludos a todos".

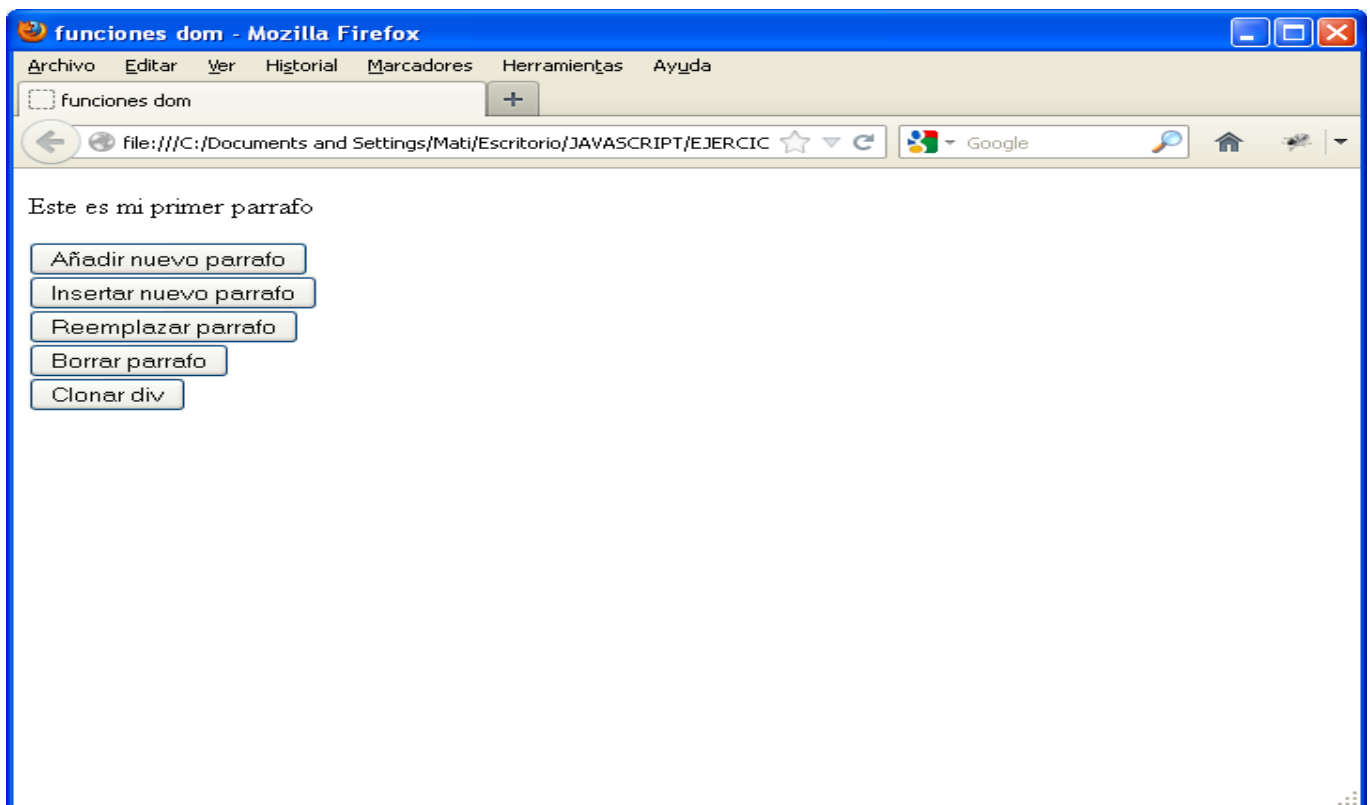
Es una breve introducción a DOM pero creo que tenéis suficiente material para trabajar. Como os he comentado anteriormente este tema tiene la dificultad añadida de entender los conceptos. Una vez los hayáis asimilado, el trabajo con las propiedades y métodos que utiliza el DOM tampoco es complejo. Inicialmente

tendréis que entender el concepto del Nodo y luego trabajaréis con él(insertar un nodo, borrarlo, reemplazarlo...), el acceso a los mismos, con los métodos `getElementsByTagName()`, `getElementByName()`, `getElementById()` y a sus atributos .

Ejercicios a realizar:

1.- Crear una página HTML con un párrafo inicial con el texto “Este es mi primer párrafo” y cuatro botones.

- **Añadir párrafo:** Añadirá el párrafo con el texto “Párrafo añadido” detrás del último párrafo creado.
- **Insertar párrafo:** Insertará el párrafo con el texto “Párrafo insertado” detrás del primer párrafo (“Este es mi primer párrafo”).(Utilizar la función `insertBEfore()`)
- **Reemplazar párrafo:** Cambiará el texto del segundo párrafo (el siguiente a “Este es mi primer párrafo”) por “Párrafo reemplazado”. (Utilizar la función `replaceChild()`)
- **Borrar párrafo:** Eliminará el párrafo que haya detrás de “Este es mi primer párrafo”.
- **Clonar div:** Duplicará (clonará) toda la estructura div que tengamos creada en el momento que pinchemos el botón. (Utilizar la función `cloneNode()`)

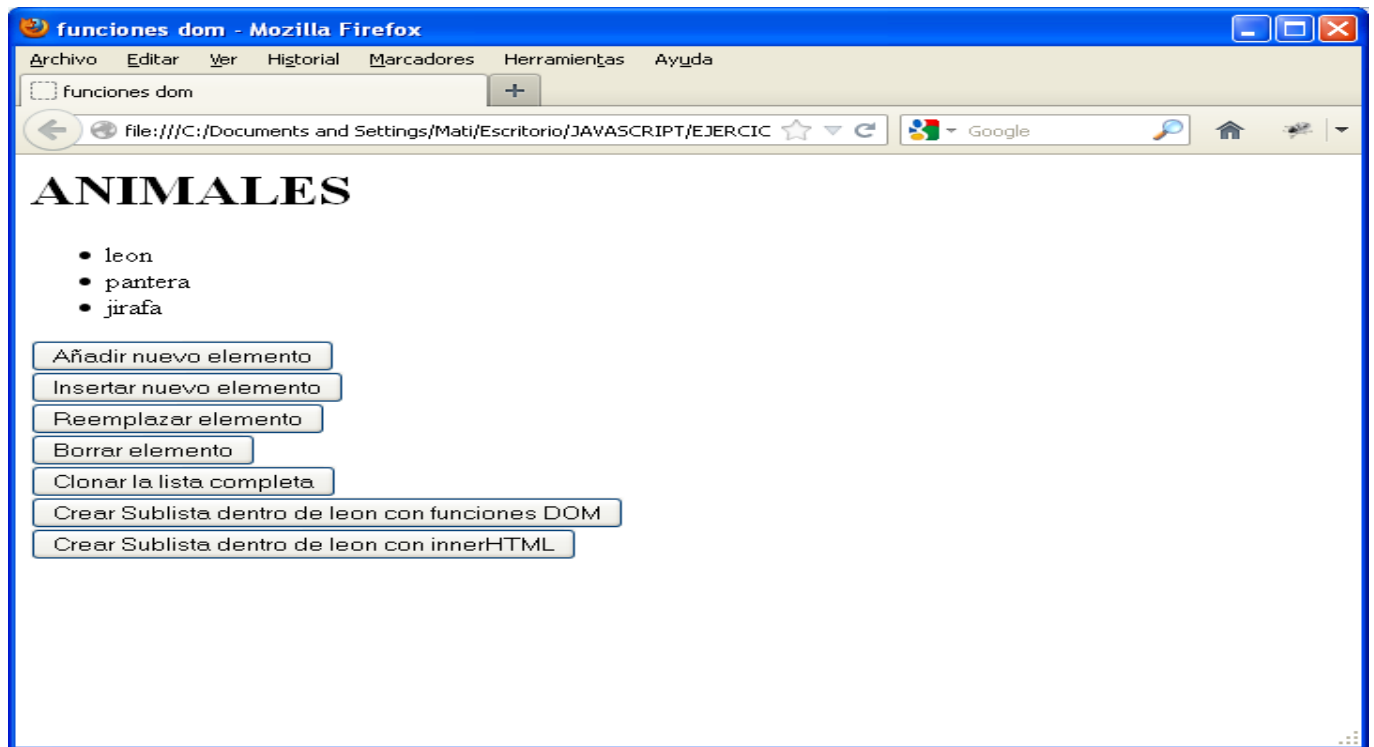


Estructura inicial de la página:

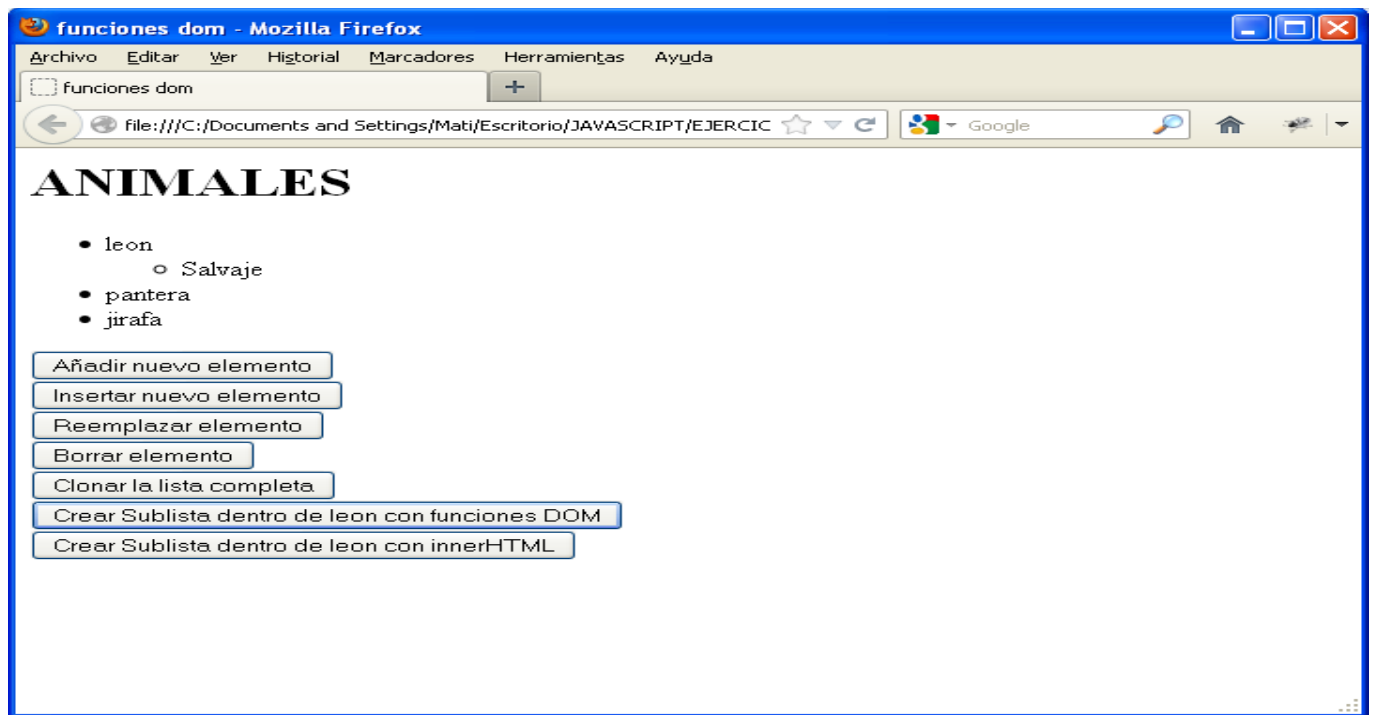
```
<HTML>
<HEAD>
  <TITLE>ejercicio 1</TITLE>
  <script>
    .....
  </script>
</HEAD>
```

```
<BODY>
  <div id="divisor">
    <p> Este es mi primer parrafo</p>
  </div>
  .....
</BODY>
</HTML>
```

2.- Crear una página HTML con una lista desordenada inicial con tres animales (ver captura de pantalla). Los botones funcionarán igual que en el ejercicio anterior pero con los elementos de la lista(el nombre del animal a añadir o reemplazar es libre). El botón “Crear Sublista dentro de león con funciones DOM” creará una nueva lista desordenada dentro del primer elemento (ver captura de pantalla). El botón “Crear Sublista dentro de león con InnerHTML” hará lo mismo que el anterior pero utilizando esta propiedad.



Pantalla inicial



(pantalla después de pinchar sobre el botón Crear Sublista dentro de león con funciones DOM)

Estructura inicial de la página:

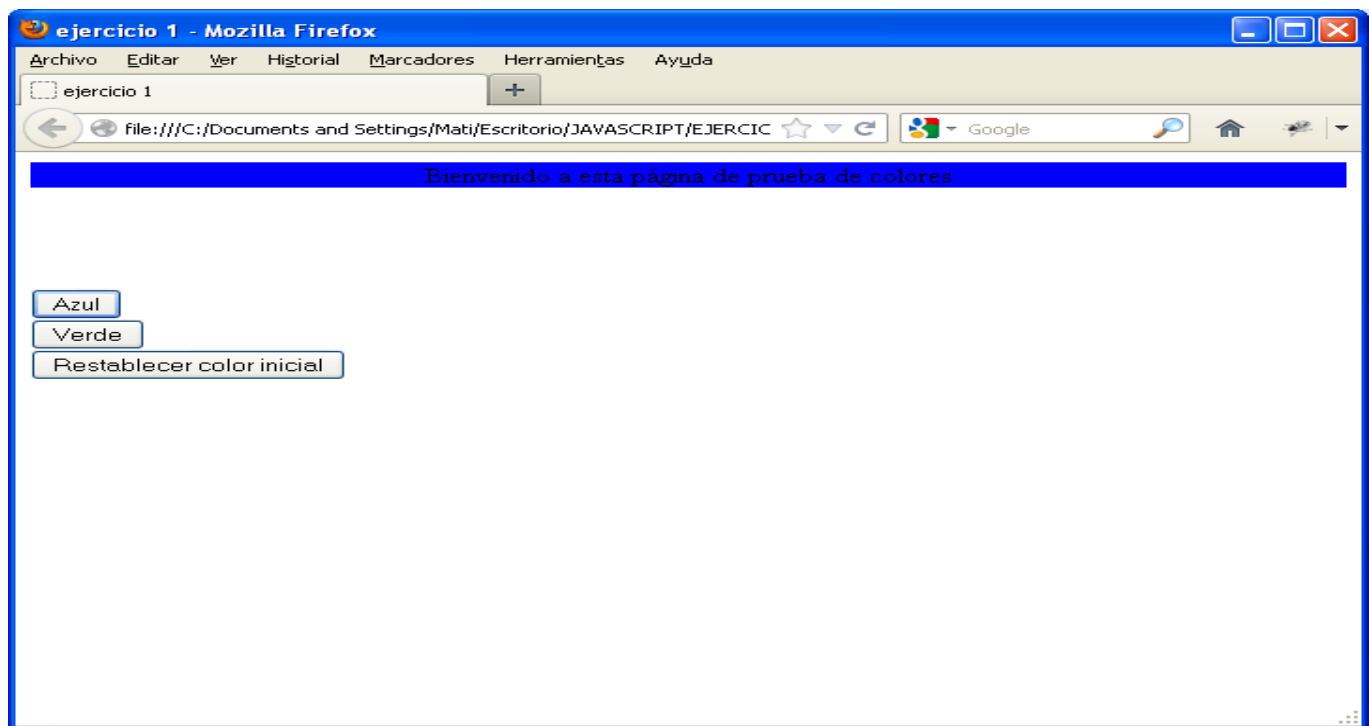
```
<HTML>
  <HEAD>
    <TITLE>ejercicio 2</TITLE>
    <script>
      .....
    </script>
  </HEAD>
  <BODY>
    <H1>ANIMALES</H1>
    <UL id="listaelemen">
      <li>leon
      <li>pantera
      <li>jirafa
    </UL>
    .....
  </BODY>
</HTML>
```

3.- Crea una página HTML con un div con el texto “ Bienvenido a esta página de prueba de colores” que aparecerá inicialmente en blanco. La página contendrá tres botones y al pinchar sobre ellos cambiará el fondo del div.

Azul: Cambiará el fondo de la pantalla a azul.

Verde: Cambiará el fondo de la pantalla a verde.

Restaurar color inicial: Dejará el fondo de la pantalla a blanco.



Ej:Pantalla que aparece cuando pulsamos el botón azul.

4.- Crea una página HTML que contenga varias imágenes y enlaces. La página tendrá un botón con el texto “Número de Imágenes” y otro con el texto “Número de enlaces”. Al pinchar sobre el primer botón debe salir un mensaje con alert que nos diga el número total de imágenes que tiene la página y al pinchar sobre el segundo botón debe salir un mensaje con alert que nos diga el número total de enlaces.

5.- Crea un elemento (el que tú quieras, por ejemplo una caja de texto) y dale atributos.

Nota: Hay que evitar el uso de la etiqueta innerHTML(excepto en el segundo ejercicio que pido utilizarla expresamente), hay que usar las funciones DOM para creación y acceso a los nodos.