

1. Crear un script que recoja el valor de la anchura y la altura total de la pantalla del usuario y calcule su diagonal.
2. Crear una página web que establezca las siguientes propiedades del objeto *Document*: color del texto= blanco, color del fondo=gris y título del documento="Modificaciones del objeto Document"
3. Partiendo del ejercicio A4.4 del tema. Esta vez debes desplazar de forma aleatoria la nueva ventana durante un número finito de veces que definas. Cuando termine de moverse tantas veces dicha ventana, terminará cerrándose. Debe imprimirse en la nueva ventana las coordenadas generadas cada vez. Nota: usar el método *setInterval()* para llamar cada segundo al método *desplazarVentana()*.
4. Desde la ventana principal abrir una nueva ventana en la cual no exista ni la ni la barra de estado ni la barra de herramientas.
5. Desde la ventana principal abrir una nueva ventana con las siguientes dimensiones: ancho 800px, alto 500px, sin barra de estado. Además no se debe permitir cambiar el tamaño de la venta al usuario una vez abierta. Probar en internet explorer y en Firefox para ver cómo se comportan.
6. Modifica el color del texto del documento. Selecciona el color deseado a partir de un desplegable que contenga los valores permitidos. Parte de lo siguiente:

```
<body>
  <div id="elementosTag">
    <h3>Modificación de color</h3>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Voluptatibus alias voluptas aut saepe
      laudantium aperiam delectus accusamus, impedit quidem velit sequi dolores obcaecati perspiciatis
      magni aspernatur perferendis ipsam minus, soluta.
    </p>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Magnam fuga veritatis, fugiat. Sed
      fugit, totam ducimus, quod blanditiis deleniti facilis sapiente, numquam ipsa fuga perspiciatis
      nulla expedita aut ea sequi!
    </p>
  </div>
  <div>
    <span>Seleccionar color:</span>
    <select id="cboSize">
      <option id="green">Verde</option>
      <option id="blue">Azul</option>
      <option id="red">Rojo</option>
    </select>
    <button id="actualizar" onclick="modificarColor();">Cambiar</button>
  </div>
</body>
```

7. Mediante la modificación de la propiedad style del elemento HTML oculta y muestra elementos HTML de la página. El texto del botón debe variar de “Ocultar” a “Mostrar”. Parte de lo siguiente:

```
<body>
  <div id="elementosTag">
    <h3>Modificación propiedades css</h3>
    <div>
      <span>Caja de texto</span>
      <input type="text" id="txtColor">
      <button id="actualizar" onclick="mostrar(this);">Ocultar</button>
    </div>
  </div>
</body>
```

8. Partiendo del siguiente documento, inserta texto en cada uno de los párrafos que permita conocer la posición que ocupa dentro del documento. Aplicar a su vez el color especificado en la caja de texto para el texto anterior.

Parrafo posicion 0

Parrafo posicion 1

Parrafo posicion 2

Parrafo posicion 3

```
<body>
  <div id="elementosTag">
    <p></p>
    <p></p>
    <p></p>
    <p></p>
    <div>
      <input type="text" id="txtColor">
      <button id="actualizar" onclick="recorrerTag();">Modificar</button>
    </div>
  </div>
</body>
</html>
```

9. Este ejercicio consiste en añadir de forma dinámica elementos al DOM. Crear una lista en el documento y añadir nuevos elementos en esta al pulsar el botón de la página con el texto escrito en la caja de texto. Parte de lo siguiente:

```
<body>
  <div id="elementosTag">
    <h3>Lista dinámica</h3>
    <ul></ul>
    <hr/>
    <div>
      <span>Nuevo Texto</span>
      <input type="text" id="txt">
      <button id="actualizar" onclick="addItemsLista();">Añadir</button>
    </div>
  </div>
</body>
</html>
```

10. Escribe un método el cual se invoque cada vez que la ventana principal se redimensione. Debe mostrar en la ventana principal las dimensiones de la ventana.

11. Escribe un método que será llamado al pulsar un botón. Dadas tres imágenes, dicho método tendrá que mostrar de forma aleatoria una de ellas cada vez.

12. Define una aplicación web que te permita añadir usuarios a una lista junto con su edad. Aquellos usuarios mayores de edad pero menores de 65 deberán tener un color de letra verde, mientras que los mayores de 65 años presentarán un color rojo. El resto de usuarios presentarán el color por defecto definido en el documento.

13. Sobre el ejercicio anterior, realiza las siguientes mejoras:

- a. Copiar todos los usuarios a una nueva lista de forma que los cambios realizados en la primera lista no afecten a la segunda.
- b. Crea los elementos necesarios que permitan eliminar de la lista de usuarios aquellos cuya edad se corresponda con el valor introducido por teclado. El elemento eliminado deberá figurar con un tipo de letra diferente y subrayado en la segunda lista, de modo que pueda observarse que ha sido eliminado en la lista original.

14. Dado un array de imágenes, implementar una función `insertImage` que reciba como parámetro dicho array, y añada dichas imágenes al documento, cada una de las cuáles debe estar en un párrafo independiente y :

```
var imgArray = new Array();
imgArray[0] = new Image();
imgArray[0].src = 'firstimage.jpg';
imgArray[1] = new Image();
imgArray[1].src = 'secondimage.jpg';
imgArray[2] = new Image();
imgArray[2].src = 'thirdimage.jpg';
function insertImage (imgArray) {
    // Code
}
```

15. Dado el documento HTML correspondiente (*wp15.rar*), sin modificar sus etiquetas, añadir el código javascript necesario en los selectores (rellenar los "" vacíos), para que de forma dinámica cada 5 segundos vaya rotando la imagen presentada entre las 3 posibles.

```
<body>
  <div id="slide">
    <article class="prev"></article>
    <article class="current"></article>
    <article class="next"></article>
  </div>
</body>
```

16. Modificar el ejercicio anterior para que aquellas imágenes que estén “detrás”, no aparezcan. Para ello cambiar el atributo *hidden* y establecerlo a *true* cuando haga falta. Utiliza el método **.setAttribute()*.

17. Sin modificar el body del HTML, modificar el fondo de las filas pares para que aparezcan con el color #eeffee. Utiliza el método `*.setAttribute()`. Recuerda que el documento tiene que estar cargado antes de modificar cualquier cosa.

```
<ul id="thelist">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```

18. Dado el siguiente *body* y contenido del *head*, añadir la funcionalidad requerida para que cuando el usuario haga *click* sobre una de las filas, ésta se borre. Utiliza el método `*.removeChild()`.

```
<body>
  <table id="mixed">
    <tbody>
      <tr>
        <th>Value One</th>
        <th>Value two</th>
        <th>Value three</th>
      </tr>
    </tbody>
    <tbody>
      <tr>
        <td>123.45</td>
        <td>apple</td>
        <td>true</td>
      </tr>
      <tr>
        <td>65</td>
        <td>banana</td>
        <td>>false</td>
      </tr>
      <tr>
        <td>1034.99</td>
        <td>cherry</td>
        <td>>false</td>
      </tr>
    </tbody>
  </table>
  <div id="result"></div>
</body>
```

```
<style>
  table {
    border-collapse: collapse;
  }
  td, th {
    padding: 5px;
    border: 1px solid #ccc;
  }
  tr:nth-child(2n+1) {
    background-color: #eeffee;
  }
</style>
<script>
  window.onload=function(){
    var e=document.getElementsByTagName("tr");
    for (var i = 0; i < e.length; i++) {
      //Add an EventListener for the click event of every row
      //
    };

    function borrar(){
      //Code
      //
    }
  }
</script>
```

19. Partiendo del siguiente código, tienes que conseguir que cuando se cargue el documento se abra un *popup*.

Hay dos clases en el *style*, uno para la capa que ocupará todo el documento, y otra por encima para el mensaje/popup. Cuando el usuario haga *click* sobre el *popup*, este se tiene que cerrar.



Sigue los pasos que se marcan en los comentarios. Cada punto es una sentencia.

```

4      <style>
5      .overlay
6      {
7          background-color: #000;
8          opacity: .5;
9          filter: alpha(opacity=50);
10         position: fixed; top: 0; left: 0;
11         width: 100%; height: 100%;
12         z-index: 1;
13     }
14     .msg
15     {
16         position: absolute;
17         background-color: #eeffee;
18         padding: 10px;
19         width: 200px;
20         height: 200px;
21         font-size: 2em;
22         z-index: 2;
23         top: 50%;
24         left: 50%;
25         margin-left: -100px;
26         margin-top: -100px;
27     }
28     </style>

```

```

29 <script>
30   window.onload=function() {
31     mostrarPopup();
32     document.getElementById("msg").addEventListener("click",quitarPopup);
33   }
34   function mostrarPopup() {
35     // 1. crea un overlay que ocupa todo el documento. Será de tipo div. Para ello usa el método *.createElement
36     var overlay =
37     //2. Asigne un atributo id=overlay.
38     //...
39     //3. Asigne un atributo class=overlay.
40     //...
41     //4. Utilizando *.appendChild(), añade overlay al body.
42     //...
43
44     // 5. crea un overlay para el mensaje. Será de tipo div. Para ello usa el método *.createElement
45     var msg =
46     // 6. crea un nodo de texto, mediante *.createTextNode con el siguiente texto: "Please join our mailing list! (Click to
47     // close.)"
48     var txt =
49     //7. Usando *.appendChild(), añade txt al msg.
50     //...
51     //8. Asigne a msg un atributo id=msg.
52     //...
53     //9. Asigne a msg un atributo class=msg.
54     //...
55     //10. Utilizando *.appendChild(), añade msg al body.
56     //...
57   }
58   function quitarPopup() {
59     //Usando removeChild, elimina el elemento con id=overlay
60     //document.body.removeChild(...);
61     //Usando removeChild, elimina el elemento con id=msg
62     //...
63   }
64
65 </script>
66 </head>
67 <body>
68   <p>Welcome!</p>
69 </body>

```