

Ejercicios DOM

CFGs 2º DAW



Curso: 2022/2023

Departamento de Informática

Juan Sevillano Hernández

Ejercicios:

Licencia:





Reconocimiento – No Comercial – Compartir Igual (by-nc-sa):

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

Índice de Contenidos

1. Ejercicio.....	4
2. Ejercicio.....	5
3. Ejercicio.....	6
4. Ejercicio.....	6
5. Ejercicio.....	6
6. Ejercicio.....	6
7. Ejercicio.....	7
8. Ejercicio.....	7
9. Ejercicio.....	8
10. Ejercicio.....	9
11. Ejercicio.....	9
12. Ejercicio.....	9
13. Ejercicio.....	10
14. Ejercicio.....	10
15. Ejercicio.....	10
16. Ejercicio.....	11

 **Importante 1:** no intentes copiar ejercicios ni tan siquiera “ver un poco” código de otros compañeros. Es el mayor error de quien empieza a programar, ya que luego no sabe resolver problemas por sí mismo y da una falsa sensación de aprendizaje.

 **Importante 2:** si en programación algo no sale a la primera... es totalmente normal. Es parte del aprendizaje. ¿Cómo crees que aprendieron los mejores programadores?

1. Ejercicio.

Vamos a cambiar el color de fondo de la página por el color amarillo, accediendo a la propiedad style de la etiqueta body.

Se tiene la siguiente página (puede ver el mismo código en la parte inferior, como lo interpreta el navegador en la parte de abajo a la izquierda y su árbol de objetos a la derecha):

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Prueba</title>

  </head>

  <body>

    <h1>Lenguaje Python</h1>

    <p>Python fue creado a finales de los años ochenta.</p>

    <p>El creador de Python es Guido van Rossum.</p>

  </body>

</html>
```

Lo primero que hay que entender es que el navegador organiza una página web como un árbol de etiquetas HTML. Podemos verla en la parte inferior y comprobar que la etiqueta raíz es 'html'. Luego vemos que tiene dos hijos que son 'head' y 'body'.

La etiqueta 'head' tiene 2 hijos que son 'title' y 'meta'. A su vez la etiqueta 'title' tiene un objeto hijo de tipo texto y la etiqueta 'meta' no tiene hijos.

Por otro lado, la etiqueta u objeto 'body' tiene 3 hijos que son dos de tipo 'p' y uno de tipo 'h1'. A su vez cada uno de estos 3 objetos tienen un hijo de tipo texto.

Debe quedar perfectamente claro que la página tiene 12 objetos estructurados como un árbol (vamos a dejar de lado al objeto <!DOCTYPE html> que normalmente no se lo manipula)

El objeto 'document' tiene una referencia directa al objeto o etiqueta 'body' de la página web. Luego podemos cambiar el color de fondo del cuerpo de la página manipulando el DOM con la siguiente sintaxis:

```
document.body.style.backgroundColor='yellow'
```

Puede sacar los caracteres de comentario en la ventana izquierda y comprobar que el fondo de la página se cambia a color amarillo. Tener en cuenta que el objeto body cuenta con la propiedad style y mediante esta podemos acceder a un conjunto de propiedades del estilo que se aplica a la etiqueta.

Modifique la propiedad color del estilo del body que luego afecta a todas las etiquetas del árbol, esto hace que el título y los dos párrafos cambien al color especificado:

```
document.body.style.color='#f0f'
```

Otras propiedades de style que puede probar:

```
document.body.style.fontFamily='Courier'
```

```
document.body.style.margin='40px'
```

```
document.body.style.border='1px solid #f00'
```

```
document.body.style.padding='20px'
```

Podemos ocultar todo el body modificando la propiedad 'visibility' de 'style' por el valor 'hidden' (luego la podemos hacer visible fijando el valor 'visible'):

```
document.body.style.visibility='hidden'
```

Podemos hacer visible u ocultar la etiqueta body accediendo a la propiedad 'visibility' de 'style', y que se actualice cada 1000 milisegundos:

```
setInterval(() => {  
  if (document.body.style.visibility === 'visible') {  
    document.body.style.visibility = 'hidden'  
  } else {  
    document.body.style.visibility = 'visible'  
  }  
}, 1000)
```

Podemos hacer que el contenido de la página se muestre durante 3 segundos y luego se haga invisible para siempre:

```
setTimeout(() => {  
  document.body.style.visibility = 'hidden'  
}, 3000)
```

2. Ejercicio.

A partir de la página web proporcionada (Ej2.html) y utilizando las funciones DOM, mostrar por pantalla la siguiente información:

1. Número de enlaces de la página
2. Dirección a la que enlaza el penúltimo enlace
3. Numero de enlaces que enlazan a <http://prueba>
4. Número de enlaces del tercer párrafo

3. Ejercicio.

Completar el código JavaScript proporcionado (Ej3.html), para que cuando se pulse sobre el enlace se muestre completo el contenido de texto. Además, el enlace debe dejar de mostrarse después de pulsarlo por primera vez. La acción de pulsar sobre un enlace forma parte de los "Eventos" de JavaScript que se ven en el siguiente capítulo. En este ejercicio, sólo se debe saber que al pinchar sobre el enlace, se ejecuta la función llamada muestra().

4. Ejercicio.

Completar el código JavaScript proporcionado (Ej4.html), para que se añadan nuevos elementos a la lista cada vez que se pulsa sobre el botón. Utilizar las funciones DOM para crear nuevos nodos y añadirlos a la lista existente. Al igual que sucede en el ejercicio anterior, la acción de pinchar sobre un botón forma parte de los "Eventos" de JavaScript que se ven en el siguiente capítulo. En este ejercicio, sólo se debe saber que al pinchar sobre el botón, se ejecuta la función llamada anade().

5. Ejercicio.

Dado el código HTML a continuación (Ej5.html), termina de rellenarlo para tras rellenar los campos de nombre y apellido y hacer clic en enviar datos aparezca por pantalla el texto «hola nombre apellidos, gracias por rellenar el formulario».

Ver imagen adjunta.

Nombre: Apellido:

hola Francesc Ricart, gracias por rellenar el formulario

6. Ejercicio.

Dado el código HTML (Ej6.html), muestra el código html de un documento web con 5 párrafos.

En el interior del script encontrarás las instrucciones:

1. Que en el tercer párrafo haya un enlace con el texto «+ info» creado con javascript y que enlace a <http://www.iesromerovargas.com/>
2. Inserta un párrafo de color rojo entre el tercer y el cuarto párrafos
3. Elimina el segundo párrafo [1]

7. Ejercicio.

Dado el código HTML (Ej7.html), completa el script para realizar las operaciones indicadas en el formulario.

Ingrese primer valor:

Ingrese segundo valor:

☐ Sumar

☐ Restar

Solución:

Ingrese primer valor:

Ingrese segundo valor:

☒ Sumar

☒ Restar

La suma es 11 - La diferencia es -1

8. Ejercicio.

Dado el código HTML (Ej8.html), completa el script para realizar las operaciones indicadas en el formulario.

Ingrese primer valor:

Ingrese segundo valor:

Solución:

Ingrese primer valor:

Ingrese segundo valor:

Sumar ▼

Operar

La suma es 7

Sumar ▲
Restar ▼

Operar

- La diferencia es 1

9. Ejercicio.

Dado el código HTML (Ej9.html), completa el script para contabilizar el número de caracteres que le faltan para completar los caracteres posibles.

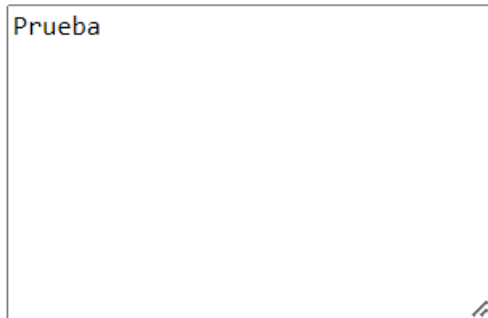
Ingrese sus comentarios

Caracteres posibles: 50

Solución:

Ingrese sus comentarios

Prueba



Caracteres posibles: 44

10. Ejercicio.

Dado el código HTML (Ej10.html), completa el script para realizar las operaciones indicadas en el formulario.

Ingrese email:

Ingrese sitio web:

confirmar

Solución:

Ingrese email: correcto

Ingrese sitio web: correcto

confirmar

11. Ejercicio.

Realiza un programa que cuando se pulse un botón con el texto “Nuevo número”, añada un elemento con un número aleatorio a una lista desordenada.

12. Ejercicio.

Realiza un programa que cree dinámicamente una tabla de 100x100. Cada elemento de la tabla tendrá un número único, que empezará en 1 y se irá incrementando de 1 en 1.

Esta página además tendrá un botón que será “Calcular número casi primos”. Este botón hará que todas las celdas de la tabla que tengan números “Casi primos” se pongan con un fondo amarillo.

Definimos aquí que es un “Número casi primo”:

- Número casi primo: es un número que solo es divisible por sí mismo, la unidad y por un solo número que no sea ni la unidad ni si mismo.

Ejemplo:

2 no es un número casi primo, porque es divisible por 1 y por 2, pero no por otro número.

4 es un número casi primo, porque es divisible por 1, por 4 y por 2.

8 no es un número casi primo, porque es divisible por 1, por 8 y por 2, pero además también es divisible por 4.

13. Ejercicio.

Realiza un programa que cree 100 elementos “checkbox” con números aleatorios. Además, la página tendrá un botón “Marcar todos” y un botón “Desmarcar todos”, con su correspondiente funcionalidad.

14. Ejercicio.

Realiza un programa que tenga 3 elementos <p> y al hacer clic sobre ellos desaparezcan (se oculten) y al hacer doble clic (los elimine del DOM). También deberá tener un botón “Reaparecer” que hará que aparezcan todos los elementos desaparecidos (pero no los eliminados).

15. Ejercicio.

A partir de la página web proporcionada (Ej15.html), utiliza las funciones DOM:

1. Se debe modificar el protocolo de todas las direcciones de los enlaces. De esta forma, si un enlace apuntaba a `http://prueba`, ahora debe apuntar a `https://prueba`
2. Los párrafos de la página cuyo atributo `class` es igual a "importante" deben modificarlo por "resaltado". El resto de párrafos deben incluir un atributo `class` igual a "normal".
3. A los enlaces de la página cuyo atributo `class` sea igual a "importante", se les añade un atributo "name" con un valor generado automáticamente y que sea igual a "importante"+i, donde i es un valor numérico cuyo valor inicial es 0 para el primer enlace.

16. Ejercicio.

A partir de la página HTML que se proporciona (Ej16.html), completar el código JavaScript definido para realizar la siguiente aplicación sencilla:

|| Genera !!

<< Comparar >>

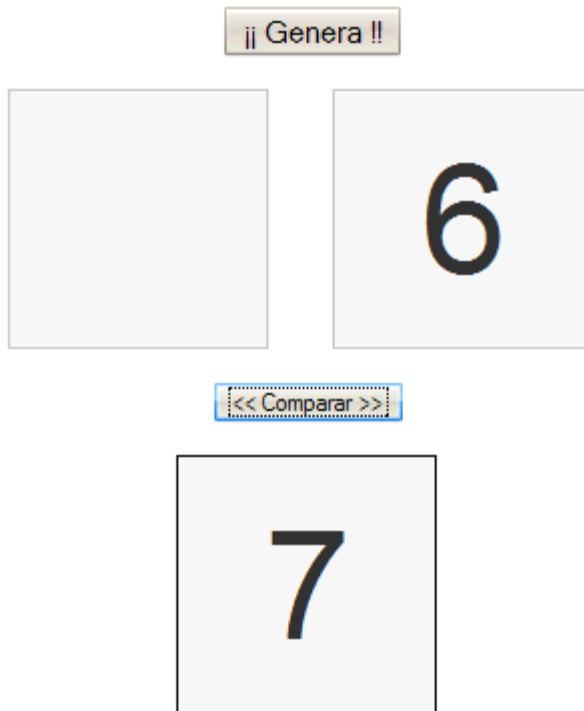
|| Genera !!

7

6

<< Comparar >>

Resultado:



Inicialmente, la aplicación cuenta con tres cajas vacías y dos botones. Al presionar el botón de "Genera", se crean dos números aleatorios. Cada número aleatorio se guarda en un elemento `<p>`, que a su vez se guarda en una de las dos cajas superiores.

Una vez generados los números, se presiona el botón "Comparar", que compara el valor de los dos párrafos anteriores y determina cual es el mayor. El párrafo con el número más grande, se mueve a la última caja que se utiliza para almacenar el resultado de la operación.

La página que se proporciona contiene todo el código HTML y CSS necesario. Además, incluye todo el código JavaScript relativo a la pulsación de los botones y que se estudiará con detalle en el siguiente capítulo.

En el ejercicio se deben utilizar entre otras, las funciones `getElementById()`, `createElement()`, `createTextNode()`, `appendChild()` y `replaceChild()`.