

Exhibiendo en un alert el nombre digitado

Carla estaba ayudando a sus colegas a prepararse para la próxima prueba de lógica de la programación. Le pidieron que resolviera la siguiente pregunta que tenía probabilidad de aparecer en la prueba: "lee del mundo HTML un texto escrito por el usuario, y con el clic de un botón muestra lo que se escribió".

Ella comenzó con la estructura mínima de todo el programa:

```
<meta charset="UTF-8">
```

```
<script>
```

```
</script>
```

[COPIA EL CÓDIGO](#)

Luego, agregó un campo de entrada y un botón con los tag `<input/>` y `<button>`, respectivamente:

```
<meta charset="UTF-8">
```

```
<input/>
```

```
<button>Mostrar texto escrito</button>
```

```
<script>
```

```
</script>
```

[COPIA EL CÓDIGO](#)

Luego, buscó en el mundo JavaScript usando

`document.querySelector()` la etiqueta `<input/>` del mundo HTML .

Explicó que esto era necesario, porque solo así JavaScript podría leer lo que está escrito en `<input>` :

```
<meta charset="UTF-8">

<input/>
<button> Mostrar texto escrito </button>

<script>
    var input = document.querySelector("input");
</script>
```

COPIA EL CÓDIGO

Luego explicó que se necesita programar dentro de una función que instrucciones deben ejecutarse cuando se hace clic en el botón. El problema es que creó solo la función, no las instrucciones. Su intención es terminarlo al final:

```
<meta charset="UTF-8">

<input/>
<button>Mostrar texto escrito</button>

<script>
    var input = document.querySelector("input");

    function mostrarTexto() {
```

```
}  
</script>
```

COPIA EL CÓDIGO

Luego, con mucha calma, dijo que la función `mostrarTexto` debe llamarse con el botón y que, primero, necesita obtener el botón del mundo HTML y hacer la asociación:

```
<meta charset="UTF-8">  
  
<input/>  
<button>Mostrar texto escrito</button>  
  
<script>  
    var input = document.querySelector("input");  
  
    function mostrarTexto() {  
  
    }  
  
    var button = document.querySelector("button");  
    button.onclick = mostrarTexto;  
  
</script>
```

COPIA EL CÓDIGO

Ten en cuenta que en este último cambio, ella asoció la función con `button.onclick`. Con esta asociación, cada vez que se hace clic en el botón, se ejecutará `mostrarTexto`, es como si, por detrás el navegador hiciera `mostrarTexto()` cada vez que se hace clic en el botón.

Pero cuando Carla comenzó a explicar el código de la función `mostrarTexto` , recibió una llamada y tuvo que salir con urgencia y dejó a medias la explicación.

¿Y ahora? ¿Pudieron sus amigos terminar el código?

¿Cuál de las siguientes opciones tiene el código completo de la función `mostrarTexto` ? Recuerda que el código debe funcionar como se esperaba, que es mostrar una alerta al usuario cuando hace clic en el botón.

A

```
<meta charset="UTF-8">

<input/>
<button>Mostrar texto escrito</button>

<script>
  var input = document.querySelector("input");

  function mostrarTexto() {
    alert(value);
  }

  var button = document.querySelector("button");
  button.onclick = mostrarTexto;
</script>
```

B

```
<meta charset="UTF-8">

<input/>
<button>Mostrar texto escrito</button>

<script>
    var input = document.querySelector("input");

    function mostrarTexto() {

        alert(input);
    }

    var button = document.querySelector("button");
    button.onclick = mostrarTexto;
</script>
```



```
<meta charset="UTF-8">

<input/>
<button>Mostrar texto escrito</button>

<script>
    var input = document.querySelector("input");

    function mostrarTexto() {

        alert(input.value);
    }

    var button = document.querySelector("button");
    button.onclick = mostrarTexto;
</script>
```



¡Correcto! Cuando se llama a la función `mostrarTexto`, pasará como parámetro de la función `alert` el valor `input.value`. No podemos simplemente pasar `input`, porque el `input` corresponde a la etiqueta y queremos que su valor provenga de esa etiqueta. Por eso, pasamos `input.value`.

