

Taller de Programación 1 - 2021-2

HTML - Javascript - DOM

Ph.D. Carlos Cares
carlos.cares@ceisufro.cl

Requerimientos:

- Haber realizado el Tutorial 1 de Programación Avanzada.
- Un editor de texto, idealmente un editor de desarrollo de software.
- Un navegador de Internet.
- Este taller se puede seguir en cualquier plataforma, Linux, Windows, o Mac.

La entrega de los resultados debe ser en el canal privado de Slack del curso, **en el momento en que obtenga las pantallas correctas** de su resultado. Las preguntas y respuestas deben quedar en el canal público. Recuerden que hay puntos porcentuales adicionales por preguntar y por responder las preguntas en el canal público. En cada problema se piden secciones de código y de resultados en el navegador de Internet, cada foto de pantalla (screenshot) debe hacerse con las siguientes características:

Características de los screenshots:

- Deben ser legibles
- Deben incluir una porción del fondo del escritorio
- Deben incluir la parte del escritorio que muestra fecha y hora

Las instrucciones revisadas en el tutorial 1 sumadas a las explicaciones en este Taller son suficientes para realizar estos 5 problemas con éxito. Reporte su avance en la medida que los logre.

Problema 1. (20%).

Se pide que muestre una tabla de 3x3 celdas. La apariencia de la tabla debe ser de un cuadrado conformado por 9 cuadrados más pequeños. Cada celda debe tener un botón. El texto dentro de cada botón debe ser “[]”, es decir, un blanco dentro de paréntesis cuadrados. No hay requerimientos especiales en la forma de desplegar la tabla pero se recomienda que lo haga en HTML directo (sin generarla en javascript).

El comportamiento del clic sobre cada botón debe ser la generación de una ventana de alerta con el mensaje correspondiente a la celda presionada identificada correlativamente, de este modo la primera celda debe decir “hola celda 1”, la segunda debe decir “hola celda 2”, así hasta la celda 9.

Debe mostrar el código HTML del despliegue de la tabla más la función gatillada por el evento clic. Como requerimiento de calidad de código se pide que la función que maneja el evento sea la misma, de modo que se pide que use parámetros para diferenciar cuál fue la celda seleccionada.

A mostrar:

- TP1-P1-A Imagen del código HTML/Javascript
- TP1-P1-B Salida de la tabla en el navegador después del clic en la 2da celda. Se debe ver la ventana emergente.
- TP1-P1-C Salida de la tabla en el navegador después del clic en la última celda. Se debe ver la ventana emergente.

Problema 2. (20%).

En base al problema anterior, se pide ahora que identifique cada botón con un identificador. Si usa la etiqueta `<button>` entonces el correspondiente objeto DOM tendrá la propiedad `innerHTML`. Entonces, se pide que, alternadamente, el clic escriba una X como texto del botón (es decir [X]) y, la segunda vez escriba una O (es decir [O]), la tercera vez escriba una X, etc.. Las ventanas emergentes ya no deben aparecer.

Note que el comportamiento esperado es como el juego del “gato”, también conocido como “tres en línea”. Sin embargo, NO ES parte de este problema el resolver que una casilla puede recibir más de un clic y cambiar una “jugada” ya hecha.

A mostrar:

- TP1-P2-A Imagen del código HTML/Javascript
- TP1-P2-B Salida de la tabla en el navegador después del clic en la 2da celda. Se deb ver el símbolo X.
- TP1-P2-C Salida de la tabla en el navegador después del clic en la tercera celda. Se debe ver el símbolo O.

Problema 3. (20%).

En base al problema anterior, se pide ahora que evite que un botón ya elegido se vuelva a elegir. Para esto use el atributo `enabled` (que es booleano) de los botones, de modo que ya no sería un botón seleccionable.

ADICIONALMENTE, debe verificar que alguno de los dos jugadores (X o O) ha ganado la partida. La forma de hacer esto es poner un mensaje en una sección (DIV) BAJO el tablero (la tabla), diciendo si ya ganó X o ya ganó O. Además todos los botones deben quedar inhabilitados (`enabled=false`) incluso los vacíos. Recuerde que para poder escribir en este DIV desde código javascript tendrá que asignarle un identificador (id) para poder usar el método `getElementById` del objeto `document`.

A mostrar:

- TP1-P3-A Imagen del código HTML/Javascript
- TP1-P3-B Salida de la tabla en el navegador después de hacer ganar a X en una diagonal. Debe mostrar el mensaje que ganó X
- TP1-P3-C Salida de la tabla en el navegador después de hacer ganar a O en una columna. Debe mostrar el mensaje que ganó O.
- TP1-P3-D Adjunte el archivo HTML en el canal de slack.

Problema 4. (20%).

En base al problema anterior, se pide ahora que el comportamiento sea exactamente el mismo, pero ahora, la línea ganadora debe ser marcada con celdas (botones) de color verde. El resto debe quedar del color original de los botones.

Es parte de este problema el que, para realizar el pintado del fondo de varios botones, use una función que reciba un arreglo de los identificadores de los objetos a pintar, por ejemplo ["btn1","btn5","btn9"] y tenga además un segundo parámetro que sea el color (string). Esta función debe llamarse `pintaFondos`.

A entregar:

- TP1-P4-A Imagen del código HTML/Javascript incluyendo la nueva función. Si el código es muy largo use dos screenshots.
- TP1-P4-B Salida de la tabla en el navegador después de hacer ganar a X en una diagonal. Debe mostrar el mensaje que ganó X y además la diagonal de color diferente.
- TP1-P4-C Salida de la tabla en el navegador después de hacer ganar a O en una fila. Debe mostrar el mensaje que ganó O y además la fila en un color diferente.
- TP1-P4-D Adjunte el archivo HTML en el canal de slack.

Problema 5. (20%).

En base al problema anterior, ahora se pide que se extienda la solución con las siguientes características. En el despliegue, abajo de todo (del DIV) debe existir un botón con el texto “Reiniciar”. Este botón debe hacer lo siguiente, todos los botones deben volver a su color original (gris?) y todos deben quedar habilitados y vacíos otra vez (osea con el texto “[]”).

ADICIONALMENTE, el DIV de los mensajes debe ser histórico-acumulativo, es decir, que debe tener una secuencia como por ejemplo:

Ganó X

Ganó X

Ganó O

Etc..

PERO cuando se presione el botón Reiniciar, y no hay un ganador previo, entonces, se debe agregar el mensaje “Empate X-O”. Ayuda: El salto de línea en HTML se especifica con la etiqueta `
` (no requiere fin). Dado que el operador “+” se usa para concatenar en javascript entonces, si x es un objeto con `innerHTML`, entonces:

```
x.innerHTML += "<br>Otra línea";
```

agrega otra línea de HTML a ese objeto.

A entregar:

- TP1-P5-A Imagen del código HTML/Javascript incluyendo la nueva función. Si el código es muy largo use dos screenshots.
- TP1-P5-B Salida de la tabla en el navegador después de hacer ganar a X en una diagonal. Debe mostrar el histórico.
- TP1-P5-C Salida de la tabla en el navegador después de 4 jugadas adicionales con el histórico. Lo último debe ser un empate y deber estar reportado en el histórico.
- TP1-P5-D Adjunte el archivo HTML en el canal de slack.