

Exercicios

1. Crea unha páxina web que teña un div de tamaño 100x100 e cor de fondo (a que queiras). Engade dous campos numéricos con valores entre 0 e 1000 para representar o alto e ancho do div. Cada vez que se cambie o valor dos campos, debe redimensionarse o div cos valores indicados.
2. Crea unha páxina web que conteña un elemento `<div>` con un tamaño preestablecido.

Cando se pulse no div, este debe ser substituído por un textarea inicializado co texto que había no `<div>` e co mesmo tamaño que o `<div>`.

O elemento textarea debe permitir editar o texto e cando se pulse a tecla “Enter” ou perda o foco, o textarea substitúese polo div, establecendo o contido deste ao texto que había na textarea. É dicir, ao pulsar o div, o seu texto convértese en editable.

3. Crea unha táboa que teña celas editables ao facer clic sobre elas.
 - a. Ao facer clic, a cela convértese en editable (aparece un textarea dentro) e pódese cambiar o HTML. Non debe haber cambios de tamaño e manterase o ancho e alto.
 - b. Só se pode editar unha cela de cada vez. Mentres un `<td>` estea en modo edición, os clics doutras celas serán ignorados.
 - c. Usa un só manexador de eventos para a táboa, é dicir, non engadas un manexador para cada cela da táboa.
 - d. Cando se pulse “Enter” o contido da edición gárdase na cela. Se a cela perde o foco, non se garda o contido editado, senón que se recupera o que había inicialmente.
4. Crea unha páxina web que mostre un reloxo seguindo o formato HH:MM:SS AM/PM. Sempre deben mostrarse dúas cifras para cada grupo. O reloxo debe mostrar as horas de 01-12, polo que se debe indicar se son AM ou PM.

Para executar unha función de forma repetida a intervalos constantes pode usarse o método [`setInterval\(\)`](#), indicando como parámetro intervalo en milisegundos. Este método devolve un ID que identifica de forma única o intervalo, permitindo eliminalo usando o método [`clearInterval\(\)`](#).

```
// imprime a mensaxe en consola despois de 2 segundos
setInterval(imprimirMensaxe, 2000);
function imprimirMensaxe() {
  console.log('Hello World!');
}
```

Tamén se podería usar o método [setTimeout\(\)](#) que permite establecer un temporizador que executa unha función despois dun retardo, en milisegundos, indicado como parámetro.

```
// imprime a mensaxe en consola despois de 2 segundos
setTimeout(imprimirMensaxe, 2000);
function imprimirMensaxe() {
    console.log('Hello World!');
}
```

NOTA: `setTimeout()` devolve o id do timeout, que é un enteiro positivo e pode ser usado para identificar o temporizador. Este valor pode ser pasado a [clearTimeout\(\)](#) para cancelar o temporizador.

5. Crea unha páxina web que permita xogar ao 3 en raia cumprindo as seguintes restricións:
 - a. A páxina debe estar composta, polo menos, dun parágrafo para mostrar información, un taboleiro 3x3 e un botón para reiniciar o xogo. Escolle os estilos CSS ao teu gusto.
 - b. Inicialmente:
 - i. O parágrafo mostra información indicando a quen lle toca xogar, a “X” ou a “O”. A decisión de quen empeza debe ser realizada aleatoriamente.
 - ii. O taboleiro aparece baleiro.
 - c. Cada vez que se fai clic nunha cela, colocárase nela un “X” ou un “O”, dependendo de a quen lle toque. Só se cambiará o valor da cela, se este estaba baleiro. Ademais, debe actualizarse o parágrafo de información indicando a quen lle toca a continuación.
 - d. Despois de cada xogada comprobarase se a partida rematou. Unha partida remata cando gañe “X”, cando gañe “O” ou cando se produza un empate. No caso de rematar a partida, o parágrafo de información debe mostrar un texto indicando quen gañou ou se houbo empate.
 - e. Cada vez que se pulse o botón para reiniciar o xogo, volverase ao estado inicial dos compoñentes.

6. Xogo das parellas.

A continuación indícanse as instrucións para implementar un xogo simple de parellas. Unha vez rematado poden engadírselle funcionalidades adicionais e mellorar a visualización con imaxes, animacións, etc.

A idea base é mostrar 8 cartas en pantalla e buscar as 4 parellas de letras.

- a. Cada vez que se cargue a páxina, crearanse dende JavaScript o código necesario para mostrar as 8 cartas. Debe implementarse de forma que sexa

fácil ampliar o número de cartas, por exemplo, para nun futuro escoller a dificultade do xogo.

Pode ser interesante engadir ao código HTML de cada carta un id que identifique a súa posición.

- b. Crea un array coas letras, por exemplo, da “A” - “D”.
- c. Crea un novo array que conteña dúas veces as letras “A” - “D” e desordénaas de forma aleatoria. A posición das letras no array usarase para gardar que letra hai detrás de cada carta.
- d. Poderase pulsar sobre cada carta para “darlles a volta” e ver que letra esconden. (Unha mellora sería usar imaxes en lugar de letras).
- e. Debes buscar un mecanismo para saber se unha carta está volteada ou non.
- f. Só pode haber dúas cartas volteadas como máximo.
- g. Cando hai dúas cartas volteadas comprobarase se a letra que esconden é igual, producíndose unha parella. Neste caso as cartas quedarán xiradas e non se poderá volver a pulsar sobre elas.
- h. Se a letra que esconden as dúas cartas xiradas non coincide, daráselle a volta a estas cartas despois dun segundos. Despois de pasado este tempo, volveranse a ocultar as letras para seguir buscando un par.
- i. Mentres están xiradas dúas cartas, non se poden xirar máis.