

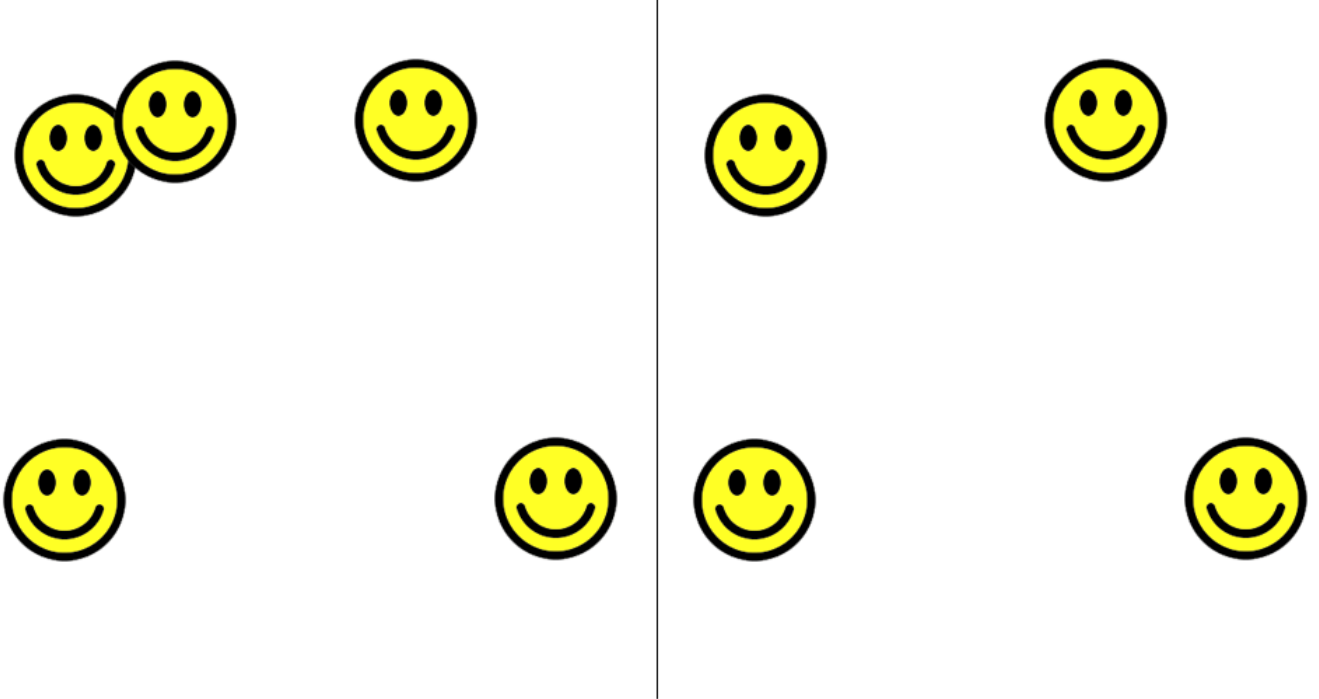
Matching game

Objetivo

Se trata de programar un juego con las siguientes características. Al cargar la página HTML se mostrarán en el navegador cinco caras en la parte de la izquierda y cuatro en la de la derecha.

Matching Game

Click on the extra smiling face on the left.

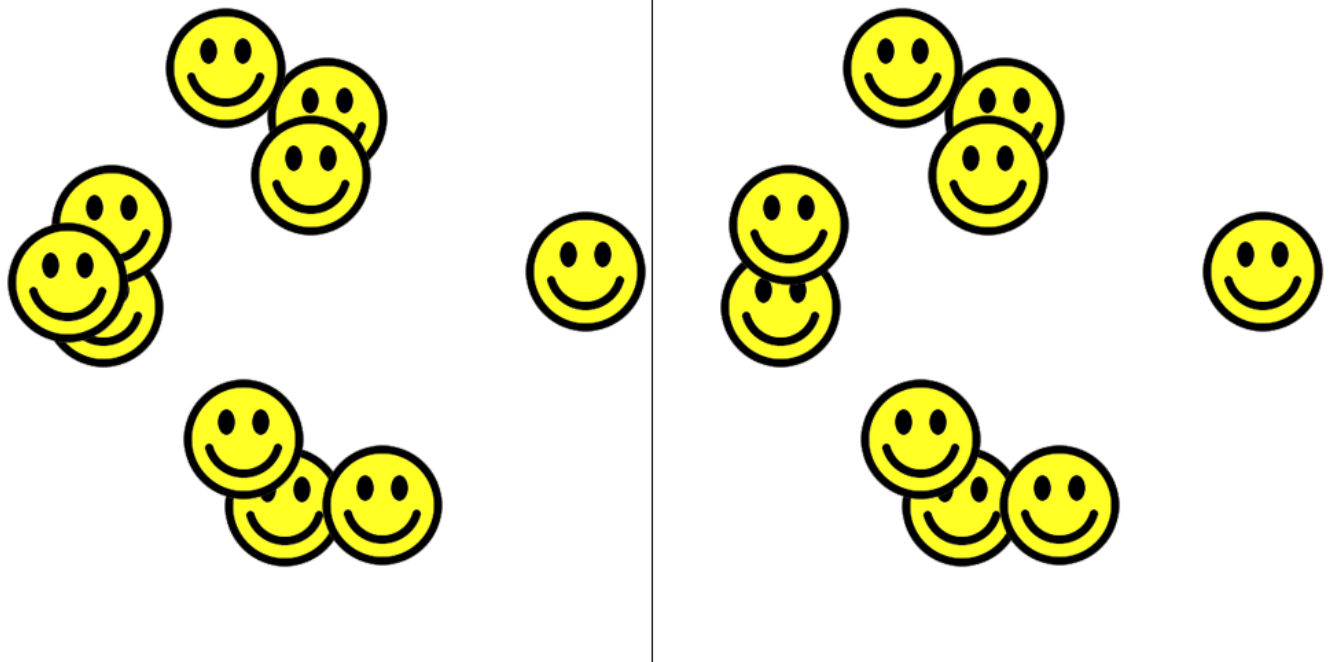


La parte de la izquierda siempre tiene una cara más. El usuario tiene que hacer click en esta cara extra. Si se pulsa cualquier cosa, excepto la cara correcta, se mostrará un mensaje de Game Over. Si se pulsa la cara correcta todas las caras se borran y se muestra un nuevo conjunto con cinco caras más posicionadas aleatoriamente. Siempre habrá una cara extra en la parte de la izquierda.

Después de pulsar la cara correcta en el ejemplo anterior se mostrará algo similar a lo siguiente:

Matching Game

Click on the extra smiling face on the left.



Después de jugar durante un rato en la pantalla habrá un montón de caras, tal y como se muestra en la siguiente figura:

Matching Game

Click on the extra smiling face on the left.



Aspectos técnicos

Hay dos elementos div en el body de la página HTML:

```
<div id="leftSide"></div>
```

```
<div id="rightSide"></div>
```

Las caras se crean dinámicamente con JavaScript. Todas las caras se generan en la parte de la izquierda bajo el elemento div con id 'leftSide'. A continuación se usa la función *cloneNode(true)* para clonar todas las caras al div con id 'rightSide', así habrá una copia exacta en la derecha. El último hijo en esta nueva rama del DOM se borra, así habrá una cara más en la izquierda.

Es necesario utilizar la gestión de eventos con JavaScript. Hay dos eventos:

1. Un evento *onclick* se aplica a la cara extra de la izquierda sobre la que el usuario tiene que pulsar. Cuando se lanza el evento una variable que contiene el número de caras que se generarán se incrementa en 5 y, entonces, el proceso de generación de caras empieza de nueva.
2. Otro evento *onclick* se aplica al body. Si se lanza esta función significa que el jugador no ha pulsado la cara correcta el juego se termina. Cuando se lanza este evento hay que mostrar un mensaje de game over y las dos funciones que gestionan los eventos se borran.

La cara sonriente está adjunta en el fichero del enunciado.

Requerimientos

El ejercicio consta de cuatro partes:

Part 1 – desarrollo de la página web sin JavaScript

Sólo se mostrará el texto con el nombre del juego y la línea del medio que separa los dos div:

Matching Game

Click on the extra smiling face on the left.



Texto

- El título 'Matching Game' debe ser un elemento *h1*, *h2*, o *h3*
- El mensaje 'Click on the extra smiling face on the left.' debe ser un elemento *p*.

Dos divs

- `<div id="leftSide"></div>`
- `<div id="rightSide"></div>`

Style

- Se necesitan al menos tres reglas de estilo.
- `position: absolute` se aplicará a todos los elementos `img`. Esto se hace para fijar más tarde la posición exacta de una imagen.
- `position: absolute` se aplicará a todos los elementos `div`, aunque realmente sólo necesita ser posicionado el div con id `rightSide`. Los dos divs tienen que tener el mismo ancho y largo: `width: 500px; height: 500px`
- Para la línea que los separa añadiremos un borde al lado izquierdo del div con id *rightSide*.

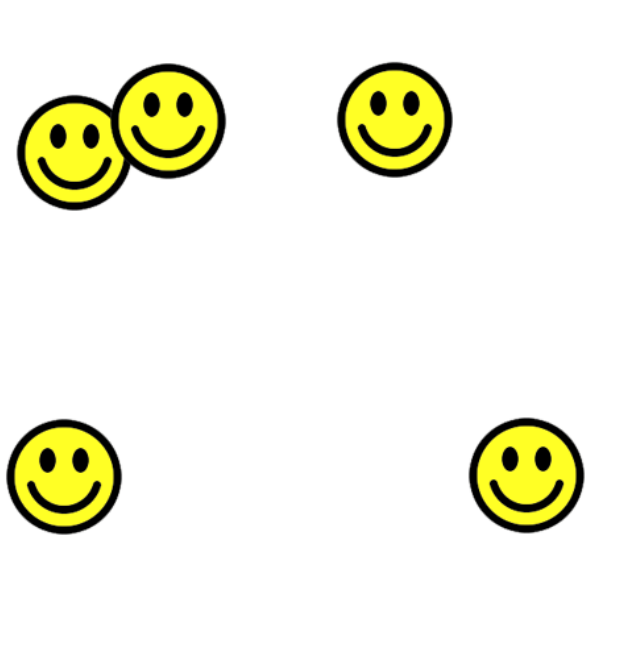
```
#rightSide { left: 500px;
border-left: 1px solid black }
```

Part 2 – generación de las imágenes de la izquierda

- Mediante JavaScript añadiremos código para generar *numberOfFaces* imágenes en la izquierda. En esta parte aún no hay imágenes en la derecha.

Matching Game

Click on the extra smiling face on the left.



El código JavaScript se ejecuta cuando la página se carga y hace lo siguiente:

Variables

- Se crea una variable llamada *numberOfFaces* inicializada a 5.
- Se crea una variable llamada *theLeftSide* que apunta al div de la parte izquierda:

```
var theLeftSide = document.getElementById("leftSide");
```

Creación de una función *generateFaces()*

- Cuando se cargue la página se llama a la función *generateFaces()* para ejecutarla.
- En esta función se crean las caras mediante un bucle que se ejecuta *numberOfFaces* veces. En cada iteración:
 - Se crea un elemento *img* usando la función *createElement()*.
 - El atributo *src* de *img* tendrá el nombre de la imagen que se mostrará:
`img.src="smile.png";`
 - La posición de una imagen se controla por los valores *top* y *left*. El valor *top* será un valor aleatorio dentro de un rango. Por ejemplo, si la altura del div es 500 y la altura de la imagen es 100, el rango de *top* estará entre 0 y 400. Usa *style.top* para establecer esta propiedad.

- El valor `left` de `img` será un valor aleatorio dentro de un rango. Por ejemplo, si el ancho del `div` es 500 y el de la imagen es 100, el rango de `left` estará entre 0 y 400. Usa `style.left` para establecer esta propiedad.
- Añade la nueva cara al `div` con `id leftSide` usando la función `appendChild()`.

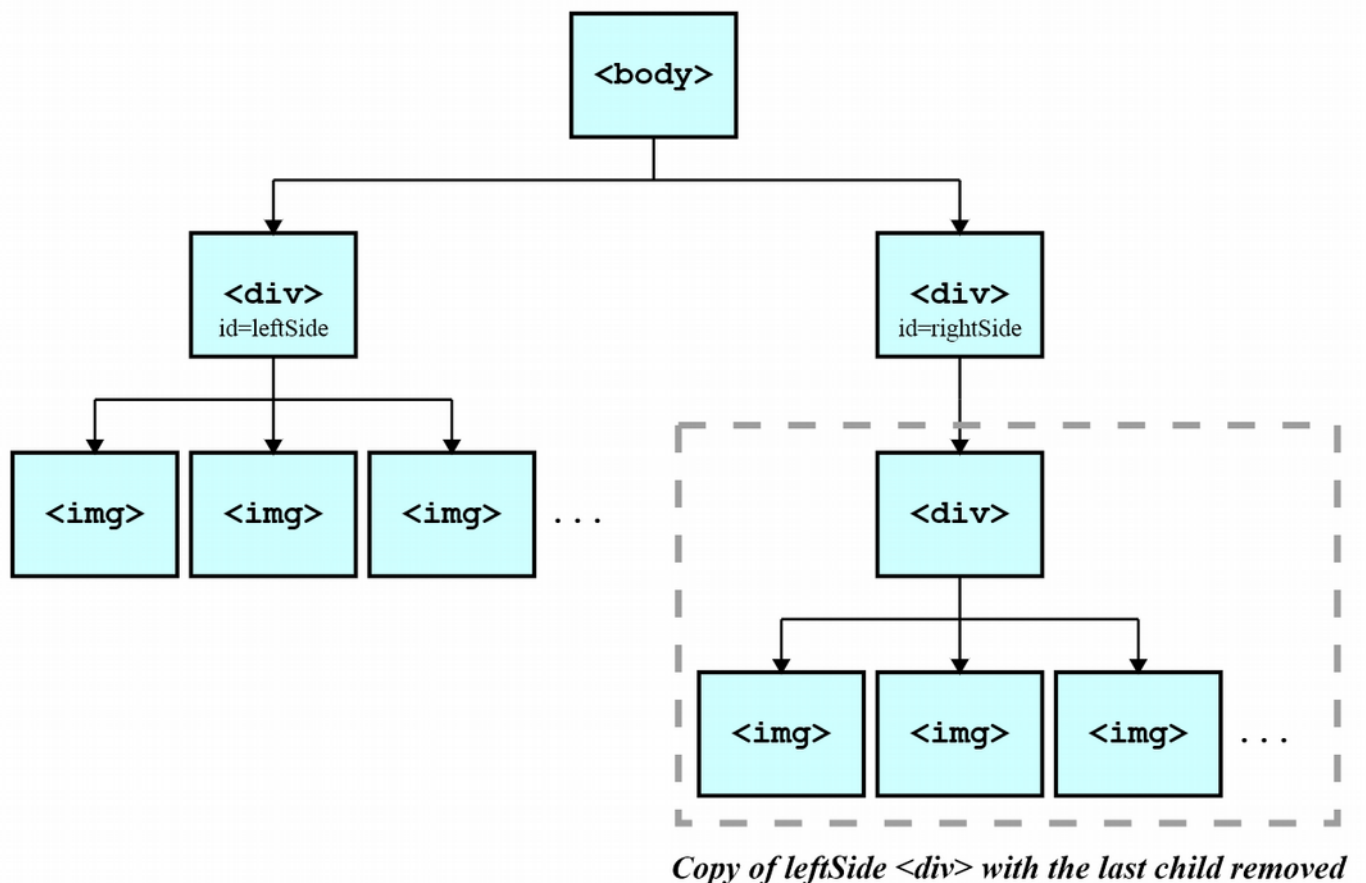
Part 3 – lado derecho

En esta parte hay que ampliar el código JavaScript de la parte 2:

- Crea una nueva variable `theRightSide` que apunte al `div` de la derecha:

```
var theRightSide = document.getElementById("rightSide");
```
- Usa `cloneNode(true)` para copiar el `div` con `id leftSide`:

```
leftSideImages = theLeftSide.cloneNode(true);
```
- Borra el último hijo de `leftSideImages`
- Añade `leftSideImages` al `div` con `id rightSide`
- El DOM será similar al siguiente:



Part 4 – finalización del juego

En esta parte hay que ampliar el código JavaScript de las partes 2 y 3 para gestionar los eventos y la lógica del juego.

Variables

- Crea una variable llamada *theBody* que apunte al elemento body:

```
var theBody = document.getElementsByTagName("body")[0];
```

Añade un gestor de eventos para la cara extra

- Tras clonar el nodo hay que añadir una función para la gestión del evento *onclick* para el último hijo de la parte izquierda, que es la cara en la que se supone que el jugador hará click.
- El código para implementar la función para este elemento será como la siguiente:

```
theLeftSide.lastChild.onclick= function nextLevel(event){  
    event.stopPropagation();  
    numberOfFaces += 5;  
    generateFaces();  
};
```

- El código

```
theLeftSide.lastChild.onclick= function nextLevel(event){ ...}
```

Significa que estamos creando una función que se ejecutará cuando el usuario haga click en el último nodo hijo en *theLeftSide*. El nombre de la función (`nextLevel()`) no es relevante y se puede llamar de cualquier manera.

- La línea `event.stopPropagation();` se necesita para asegurar que el evento no se propagará a otros elementos en la página web, como otras caras.
- La función `generateFaces();` genera un nuevo conjunto de caras. Dado que el valor de `numberOfFaces` se ha incrementado en 5 habrá 5 caras más en ambos lados.

Añade un gestor de eventos para el body

- Hay que crear otra función para gestionar el evento que se produce cuando el usuario pulsa en cualquier otro lugar de la página que no sea la cara extra:

```
theBody.onclick = function gameOver() {  
    alert("Game Over!");  
    theBody.onclick = null;  
    theLeftSide.lastChild.onclick = null;  
};
```

- La línea: `theBody.onclick = function gameOver() {` crea una función que se ejecuta cuando se pulsa en *theBody*.

- La línea: `theBody.onclick = null;`

indica que no sucederá nada si el usuario pulsa a partir de este momento en cualquier lugar de la página.

- La línea: `theLeftSide.lastChild.onclick = null;` indica que no sucederá nada si el usuario pulsa a partir de este momento en la cara extra.

Borrar los nodos hijo

- Recuerda que cada vez que el usuario hace click en la cara correcta todas las demás caras se borran y se genera un nuevo conjunto de caras. Esto significa que todos los hijos de los div *leftSide* y *rightSide* tienen que borrarse. Para borrar todos los nodos hijo puedes usar un bucle while, tal y como vimos en clase.

Entrega

Hay que enviar un único fichero HTML que incluya el código JavaScript dentro del fichero, es decir, no hay ficheros externos, ni de CSS ni de JavaScript.

Adjunta también pantallazos del funcionamiento del programa.