

wawiwa

EVENT HANDLER (manejador de eventos)

Eventos

Un **evento** es una acción (o ocurrencia) que sucede en el sistema que estás programando.

Cuando nos referimos a la web, los eventos son disparados dentro de la ventana del navegador y tienen a ser adjuntos a un ítem específico que reside en él — esto puede ser un elemento único o un conjunto de elementos, el documento HTML es cargado en la pestaña actual o la ventana del navegador entera. Hay muchos diferentes tipos de eventos que pueden suceder. Por ejemplo:

- El usuario selecciona cierto elemento o suspende el cursor sobre cierto elemento.
- El usuario elige una tecla en el teclado.
- El usuario redimensiona o cierra la ventana del navegador.
- Una página web termina de cargar.
- Un formulario es enviado.
- Un video se reproduce, pausa o termina.
- Un error ocurre.

Eventos

Más ejemplos de eventos: <https://developer.mozilla.org/en-US/docs/Web/Events>

Evento	Descripción
onclick	Click en elemento
onmouseover	Cuando el cursor se suspende sobre un elemento
onmouseout	El usuario mueve el mouse fuera de un elemento HTML
onblur	Eventos relacionados con elementos ganando y perdiendo focus
onfocus	El evento onfocus ocurre cuando un elemento obtiene focus
onchange	Un elemento HTML cambia
onkeydown	El usuario pulsa una tecla
onkeyup	El evento onkeyup ocurre cuando el usuario libera una tecla (del teclado)
onload	El navegador termina de cargar la página

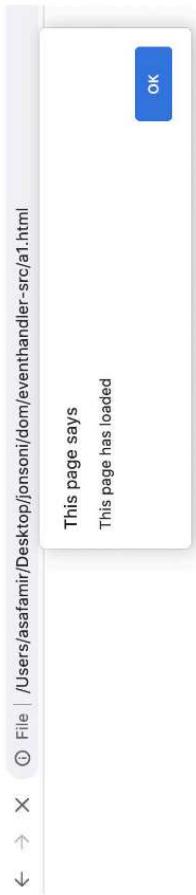
Otros Eventos

Otros eventos pueden ser:

- 1.) Eventos de media - audio/video.
- 2.) Eventos de progreso - monitorear el navegador en curso.
- 3.) Transición de eventos de CSS - inicio/ejecución/finalización de transición

Manejador de Eventos Onload

```
<html>
  <body onload="alert('This page has loaded');">
    here is my text
  </body>
</html>
```



Algunas maneras para programar Eventos

```
<html>
  <body>
    
  </body>
<script>
  var img = document.querySelector("img");

  img.onclick = function() {
    img.src = "images/green.png";
  };

  img.onmouseover = function() {
    img.src = "images/red.png";
  };
</script>
</html>
```



Algunas maneras para programar Eventos



```
<script>
function myFunction () {
    alert ("Hello myFunction");
}

document.addEventListener ('cli
</script>
```

onBlur y onFocus

```
<?> C ⌘ File | /Users/asafamir/Desktop/jsononi/dom/eventhandler-src/a4.html  
onfocus/onblur  
  
<html>  
  <body>  
    <button id = "btn1" onFocus=<?> C ⌘ File | /Users/asafamir/Desktop/jsononi/dom/eventhandler-src/a4.html  
      onBlur="myOnBlur (t)" onfocus/onblur  
      onfocus/onblur  
    </button>  
  </body>  
  <script>  
    function myOnFocus (obj) {  
      obj.style.background = "orange";  
    }  
    function myOnBlur (obj) {  
      obj.style.background = "yellow";  
    }  
  </script>  
</html>
```

The diagram illustrates the event flow. It starts with the `onFocus` and `onBlur` attributes on the button element. Arrows point from these attributes to the `onfocus` and `onblur` functions defined in the `<script>` block. From each function, an arrow points down to its corresponding implementation in the code.

onFocus - cuando lo presionamos

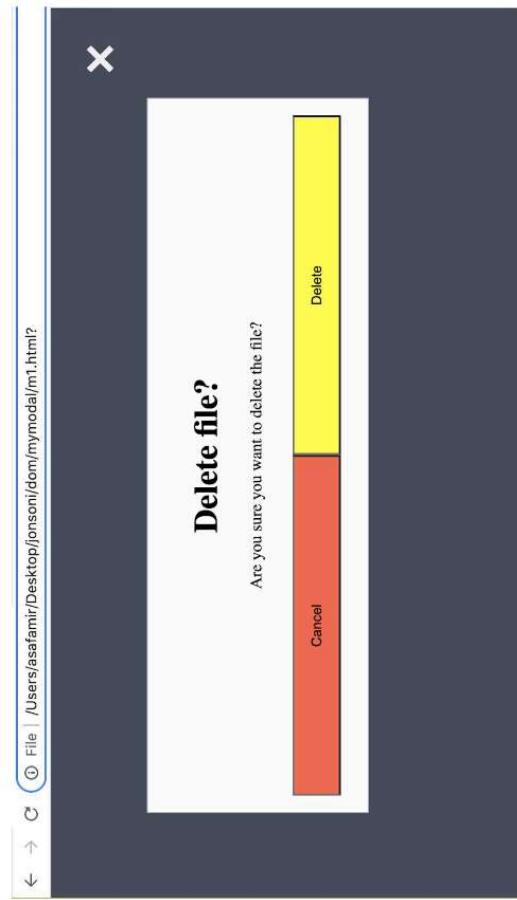
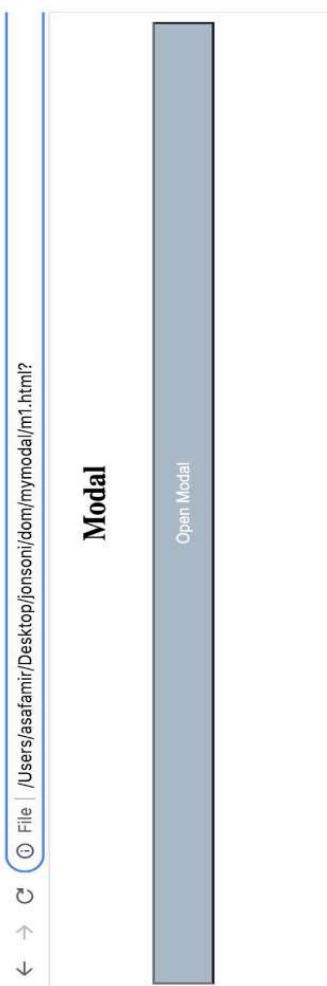
onBlur - cuando dejamos el text box, por ejemplo.

Creator botones dinámicamente

```
<html>
  <body>
    <button onclick="createButtons (5, 'blue')">
      Create
    </button>
    <section style="margin-top: 50px;"></section>
  </body>
<script>
  function createButtons (num, color) {
    for (let i = 0; i < num; i++) {
      const div = document.createElement ('div');
      div.style.backgroundColor = color;
      div.style.height = "30px";
      div.style.margin = "3px";
      div.style.width = "30px";
      div.style.cssFloat = "left";
      div.style.cursor = "pointer";
      div.draggable = "true";
      div.onclick = function () {
        alert ('you clicked on a box #' + i);
      };
    }
    document.getElementsByTagName ('section')[0].appendChild (div);
  }
</script>
</html>
```



Ejemplo de proyecto modal



Ejemplo de modal reiniciado - archivo HTML

```
<h2 style="text-align: center;">Modal</h2>
<button class="showModal">Open Modal</button>
<div id="myModal" class="modal">
  <span onclick="document.getElementById('myModal').style.display='none'">
    class="close" title="Close Modal">
      ×
    </span>
  </form>
  <form class="modal-content">
    <div class="container">
      <h1>Delete file?</h1>
      <p>Are you sure you want to delete this file?</p>
      <div class="clearfix">
        <button class="btncancel">Cancel</button>
        <button class="btndelete">Delete</button>
      </div>
    </div>
  </form>
</div>
```

Modal

Delete file?

Are you sure you want to delete the file?

Cancel **Delete**

Ejemplo de modal reiniciado - Agrega el CSS siguiente

```
* { box-sizing: border-box; }

button {
    background-color: #aabbcc; text-align: center;
    color: white;
    padding: 14px 20px;
    margin: 10px 0;
    cursor: pointer;
    width: 100%;
}

.button {
    position: absolute;
    top: 0;
    width: 100%;
    height: 100%;
    overflow: auto;
    background-color: #474e5d;
    padding-top: 50px;
}

.button .cancel {
    background-color: tomato;
    color: black;
}

.button .cancel::before {
    background-color: #fefefe;
    margin: 5% auto 15% auto;
    border: 1px solid #aabbcc;
    width: 80%;
}

.button .cancel::before {
    float: left;
    width: 50%;
}

.button .cancel::before {
    background-color: yellow;
    color: black;
}
```

Ejemplo de modal reiniciado - Agrega el CSS siguiente

```
hr {  
    border: 1px solid #f1f1f1;  
    margin-bottom: 25px;  
}  
  
.close {  
    position: absolute;  
    right: 35px;  
    top: 15px;  
    font-size: 50px;  
    font-weight: bold;  
    color: #f1f1f1;  
}  
  
.close:hover, .close:focus {  
    color: #f44336;  
    cursor: pointer;  
}  
  
/* Clear floats */  
.clearfix::after {  
    content: " ";  
    clear: both;  
    display: table;  
}
```

Ejemplo de Modal fase 1

1. Declara una var modal con una referencia a id=mymodal .
2. Escribe una función showModal () que el estilo del id=mymodal display sea block .
3. Escribe una función hideModal () que el estilo del id=mymodal display sea none .

Ejemplo de Modal fase 1

1. Declara una var modal con una referencia a id=mymodal .
2. Escribe una función showModal () que el estilo del id=mymodal display sea block .
3. Escribe una función hideModal () que el estilo del id=mymodal display sea none .

```
var modal = document.getElementById('mymodal');

function showModal() {
    document.getElementById('mymodal').style.display = 'block';
}

function hideModal() {
    document.getElementById('mymodal').style.display = 'none';
}
```

Ejemplo de Modal fase 2

1. Agrega event listeners (oyentes de eventos) de click a un botón con class=showModal que llame a la **función showModal**.
2. Agrega event listeners de click a un botón con class btnCancel que llame a la **función hideModal**.
3. Agrega event listeners (oyentes de eventos) de click a un botón con class=btndellete que llame a la **función hideModal**.

Ejemplo de Modal fase 2 - solución

1. Agrega event listeners (oyentes de eventos) de click a un botón con class=showModal que llame a la función **showModal**.
2. Agrega event listeners de click a un botón con class=btnCancel que llame a la función **hideModal**.
3. Agrega event listeners (oyentes de eventos) de click a un botón con class=btndelete que llame a la función **hideModal**.

```
const modal = document.getElementById('myModal');

document.querySelector('.showModal').addEventListener("click", showModal)
document.querySelector('.cancel').addEventListener("click", hideModal)

document.querySelector('.btndelete').addEventListener("click", hideModal)

function showModal () { document.getElementById('myModal').style.display = 'block'; }

function hideModal () { document.getElementById('myModal').style.display = 'none'; }
```

Ejemplo de Modal fase 3

Agrega un event listeners de teclado anónimos a "document" que escuchen la tecla "Escape" presionada. Cuando el usuario haga clic en "Escape", el modal se cerrará.

Ejemplo de Modal fase 3 - solución

Agrega un event listeners de teclado anónimos a "document" que escuchen la tecla "Escape" presionada. Cuando el usuario haga clic en "Escape", el modal se cerrará.

```
document.addEventListener('keydown', function(e) {  
  if (e.key === 'Escape') {  
    hideModal();  
  }  
});
```

Proyecto de ejemplo de Search search

Countries

Search for Country..

Germany
Poland
America
Romania
Italy

Ejemplo de search reiniciado - archivo HTML

```
<h1 style="color:coral; text-align: center;">
    Countries
</h1>

<input type="text" id="userInput" onkeyup="search ()"
placeholder="Search for Country...." title="Type Country">

<ul id="countryUL">
    <li><a href="#">Germany</a></li>
    <li><a href="#">Poland</a></li>
    <li><a href="#">America</a></li>
    <li><a href="#">Romania</a></li>
    <li><a href="#">Italy</a></li>
</ul>
```

Countries

Search for Country...

- Germany
- Poland
- America
- Romania
- Italy

Ejemplo de search reiniciado - Agrega el siguiente CSS

```
#countryUL li a {  
    * { box-sizing: border-border: 1px solid rgb(40, 5, 236);  
        margin-top: -1px;  
        background-color: #c1e90d;  
    }  
  
    #userInput {  
        background-position-padding: 12px;  
        background-repeat: text-decoration: none;  
        width: 100%;  
        font-size: 18px;  
        font-size: 16px;  
        color: black;  
        padding: 12px 20px display: block;  
        border: 1px solid rgb(46, 8, 218);  
        margin-bottom: 20px;  
    }  
  
    #countryUL li a:hover:not (.header) {  
        background-color: #ee0b0b;  
    }  
  
    #countryUL {  
        list-style-type: none;  
        padding: 0;  
        margin: 0;  
    }  
}
```

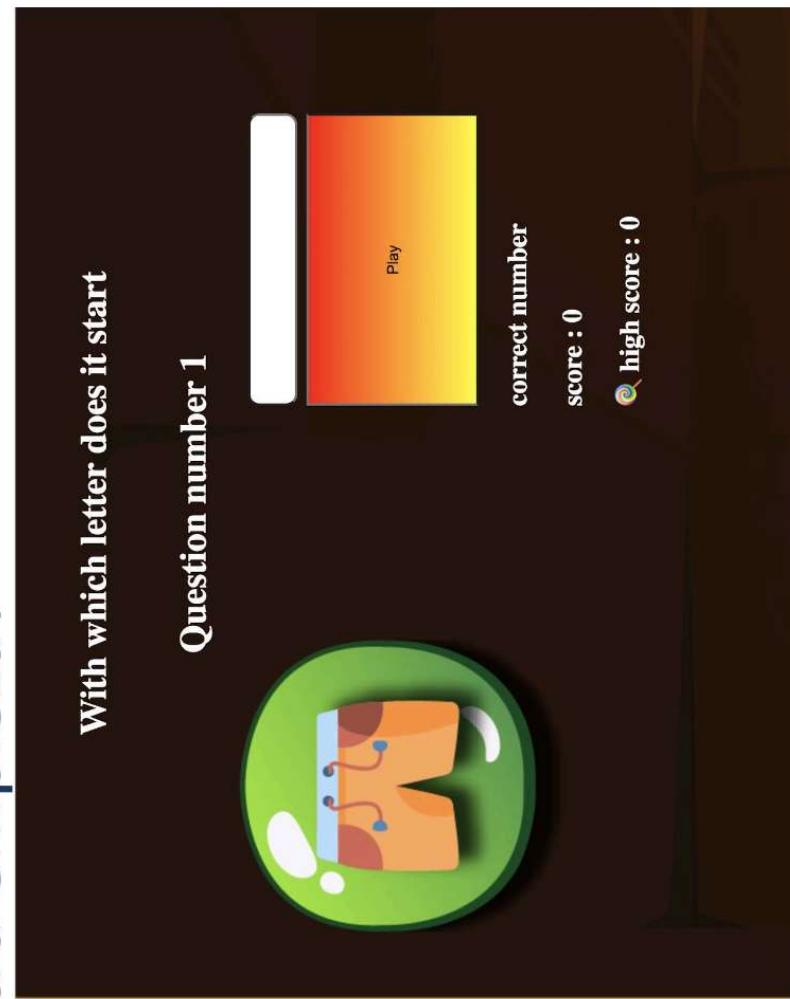
Ejemplo de Search js

```
function search () {
    const filter =
        document.getElementById("userInput") .value .toUpperCase () ;
    const li =
        document.getElementById("countryUL") .getElementsByName ("li") ;

    for (let i = 0; i < li.length; i++) {
        const a = li[i] .getElementsByName ("a") [0];
        const value = a.textContent || a.innerText;
        if (value .toUpperCase () .indexOf (filter) > -1) {
            li[i] .style .display = "";
        }
        else {
            li[i] .style .display = "none";
        }
    }
}
```

Mini proyecto 1

¿Con qué letra empieza?



Antes de empezar: Números aleatorios

```
function getRandomInt(max) {  
    return Math.floor(Math.random() * max);  
  
    console.log(getRandomInt(3));  
    // resultado esperado: 0, 1 o 2  
  
    console.log(getRandomInt(1));  
    // resultado esperado: 0  
  
    console.log(Math.random());  
    // resultado esperado: un número entre 0 y <1
```

Mini proyecto 1 reiniciar - archivo html

```
<html>
<body class="backGroundBody">
<h1>With which letter does it start</h1>
<h1 class="countQuestion">Question number 1</h1>
<div class="imgGuess">

</div>
<div class="input">
<input type="text" id="inputText" value="" />
<button class="btnPlay" type="button" value="Play">>Play</button>
</div>
<div class="output">
<h2 class="message" id="correctMessage">correct answer</h2>
<h2 class="score" id="score">score : 0 </h2>
<h2 class="highscore" id="highScore">>��分</h2>
</div>
<div class="playAgain">
<button type="button" value="Game over-lets play again">>Game over-lets play again</button>
</div>
</body>
</html>
```

With which letter does it start

Question number 1

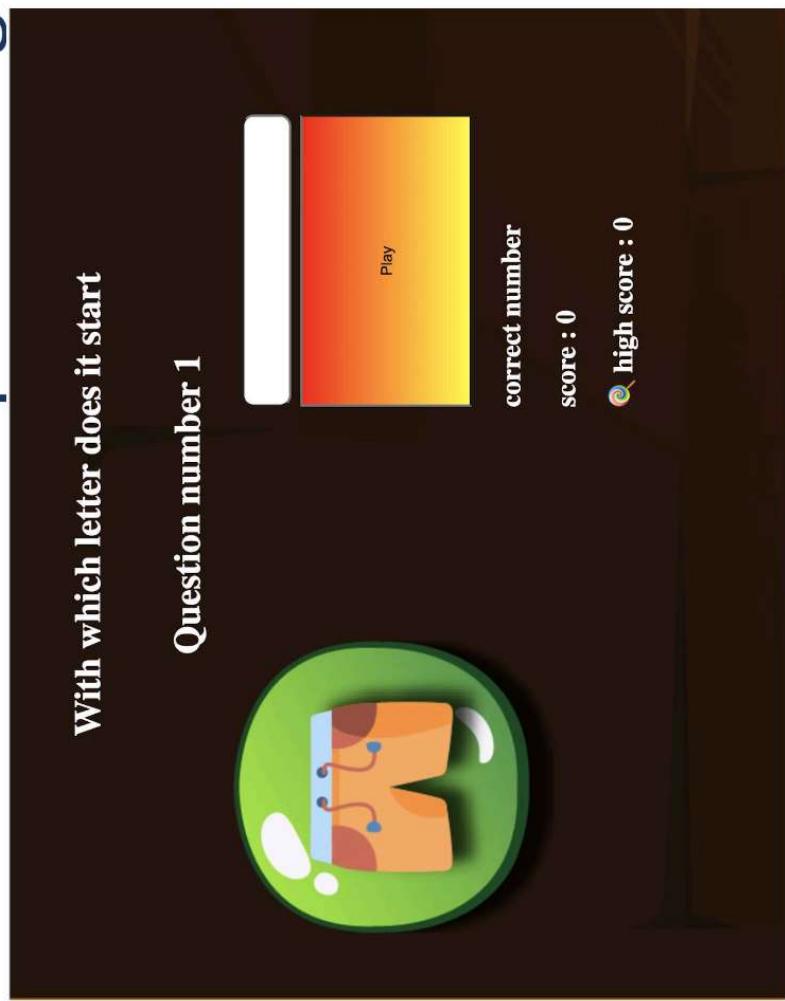
Mini proyecto reiniciado - Agrega el siguiente CSS

```
.imgGuess img {  
    width: 50%;  
}  
  
.btn {  
    width: 100%;  
    height: 150px;  
    background-image: linear-gradient(red, yellow);  
}  
  
h1 {  
    text-align: center;  
    color: white;  
    margin-top: 50px;  
}  
  
input[type=text] {  
    width: 100%;  
    padding: 12px 20px;  
    margin: 8px 0;  
    box-sizing: border-box;  
    border-radius: 10px;  
    margin-left: 10%;  
}  
  
.playAgain {  
    float: left;  
    width: 100%;  
    display: block;  
    margin-left: 45%;  
}  
  
.input {  
    float: left;  
    display: block;  
    width: 30%;  
}  
  
.output {  
    float: left;  
    display: block;  
    color: white;  
    width: 30%;  
}  
  
.imgGuess {  
    float: left;  
    display: block;  
    color: white;  
    width: 60%;  
}
```

Mini proyecto reiniciado - Agrega el siguiente CSS

```
.playAgain button{  
    margin: 0;  
    position: absolute;  
    left: 50%;  
    background-image: linear-gradient (rgb(9, 255, 0), yellow);  
    -ms-transform: translate (-50%, -50%);  
    transform: translate (-50%, -50%);  
    width: 30%;  
    height: 60px;  
    margin-top: 10px;  
    margin-bottom: 10px;  
}  
  
.hidden {  
    display: none;  
}
```

Mini proyecto fase 2 - Después de agregar el CSS



Mini proyecto instrucción 1

Abre un archivo html llamado mini.html y agregale el html y css.

Abre una carpeta llamada imágenes y pon tres imágenes en el.

Configura 5 variables globales en el script.

```
var arr = ["images/hat.png", "images/boat.png", "images/short.png"] ;  
var arrWord = ["hat", "boat", "shirt"] ;  
var counter = 0 ;  
var i = selectRandomImage() ;  
var score = 0 ;  
var highScore = 0 ;  
document.body.classList.add('backGroundBody') ;  
let btnPlayAgain = document.querySelector('.playAgain') ;
```

Mini proyecto instrucción 2

Escribe una función llamada `selectRandomImage`. La función debe hacer lo siguiente:

1. Cambia el texto de `countQuestion` a "Question number" + counter.
2. Obtiene un número aleatorio entre 0 y 3 a una variable llamada `index`.
3. Muestra la imagen del arreglo `arrWord` en un elemento el cual su id es `imgToGuess`.
4. La función retorna el index a la variable llamada `i`.
5. Actualiza la página varias veces y verifica si la imagen cambia.

Mini proyecto instrucción 2 - Solución

```
var arr = ["images/hat.png", "images/boat.png", "images/short.png"];
var arrWord = ["hat", "boat", "shirt"];
var counter = 0;
var i = selectRandomImage();
var score = 0;
var highScore = 0;
document.body.classList.add('backGroundBody');
let btnPlayAgain = document.querySelector('.playAgain');

function selectRandomImage () {
    document.querySelector('.countQuestion').textContent = "Question number " + counter;
    let index = Math.floor(Math.random() * 3);
    console.log(index);
    document.getElementById("imgToGuess").src = arr[index];
    return index;
}
```

Mini proyecto

instrucción 3

Escribe una función llamada **play()** La función hará lo siguiente:

1. Incrementa el counter un 1
 2. Si el counter <= 3
 - a. Declara let word = arrWord[i] // arrWord y i son variables globales
 - b. Declara let letter = document.getElementById("input") .value; // valor del input del usuario
 - c. If !letter entonces document.querySelector(".message") .textContent = "No Letter"
 - i. Y disminuye al usuario 2 puntos
 - d. Else If letter == word[0] document.querySelector(".message") .textContent = "correct Answer"
 - i. Incrementa al usuario 5 puntos
 - e. Else If letter != word[0] document.querySelector(".message") .textContent = "not correct Answer"
 - i. Disminuye al usuario 3 puntos
 - f. Reinicia el valor del document.getElementById("input") .value a vacío ("")
 - g. Muestra el score correcto en document.querySelector(".score") .textContent
 - h. If counter < 3 entonces i = selectRandomImage();
 3. If counter > 2 {} // lo escribiremos en las siguientes fases
 4. Agrega un event listener de click al botón play () abajo de la función play
- ```
document.querySelector(".play") .addEventListener('click', play);
```

## Mini proyecto instrucción 3 - solución

```
function play() {
 counter++;
 if (counter <= 3) {
 const word = arrWord[i];
 const messageElement = document.querySelector(".message");
 const letter = document.getElementById("input").value;
 if (!letter) {
 messageElement.textContent = "No Letter 🎈";
 score -= 2;
 } else if (letter === word[0]) {
 messageElement.textContent = "Correct Answer 🎉";
 score += 5;
 } else if (letter != word[0]) {
 messageElement.textContent = "Wrong Answer 🤦";
 score -= 2;
 }
 document.getElementById("input").value = "";
 document.querySelector(".score").textContent = "score : " + score;
 }
}

if (counter < 3) {
 i = selectRandomImage();
}

document.querySelector(".play").addEventListener('click', play);
```

Agrega la  
función play  
y ve la  
solución

# Mini proyecto instrucción 4

```
Agrega 2 funciones: showbtnPlayAgain, hiddenbtnPlayAgain arriba de
selectRandomImage ()

// botones hidden y show button, game over, play again
const showbtnPlayAgain = function () {
 btnPlayAgain.classList.remove ('hidden');
};

const hiddenbtnPlayAgain = function () {
 btnPlayAgain.classList.add ('hidden');
};
```

# Mini proyecto instrucción 5

Agrega la función gameover: **function gameOver()**

1. If score > highScore entonces guarda highScore y muestralos en document.querySelector(".highScore").textContent with "��分: highScore" + score.
2. Remueve la class backGroundBody con document.documentElement.classList.remove('backGround')  

```
document.documentElement.classList.remove('backGround');
if(counter>2){
 rain();
}
over en la función play
```



# Mini proyecto instrucción 5 solución

```
function gameOver() {
 if (score > highScore) {
 highScore = score;
 document.querySelector(".highscore").textContent = "🔍 high score " + score;
 }
}
```

```
document.body.classList.remove('backGroundBody');
showbtnPlayAgain();
}
}
```

```
Llama la función if(counter>2){ ay co if(counter>2){ .f counter > 2
gameOver()
}
```

# Mini proyecto instrucción 6

Agrega a playAgain un selector EventListener de click que llame a la función reset

```
document.querySelector(".playAgain").addEventListener('click', reset);
```

Agrega la función reset: **function** reset ()

1. document.body.classList.add('backGroundBody');
2. Counter = 0
3. i = selectRandomImage(); // obtiene una imagen aleatoria nueva
4. Reinicia el score a 0
5. document.getElementById("input").value = "";
6. hiddenbtnPlayAgain();

# Mini proyecto instrucción 6

```
//Reinicia el juego y juega de nuevo reset
document.querySelector("#playAgain").addEventListener('click', reset);

function reset () {
 document.body.classList.add('backGroundBody');

 counter = 0;

 i = selectRandomImage(); // get new random picture
 score = 0;

 document.querySelector('.score').textContent = 0;
 document.getElementById("input").value = "";

 hiddenbtnPlayAgain();
}

}
```

# Mini proyecto, corre el proyecto :)

