Javascript juego MOKEPON

Es un juego de combate de mascotas, hay mascotas de tierra, fuego y agua. Los jugadores deben elegir sus mascotas.

Una vez hecha la estructura html, necesitamos que interactúen los botones para que los jugadores puedan llevar a cabo su elección.

**Herramientas de JS**

**Necesito que JS “escuche el elemento al hacer click”**

1. **Debo crear una variable, asignarle un nombre *“botonMascotaJugador”***

* **En JS no se admiten** **“guiones medios -”**, tampoco **empezar con mayúscula**
* La forma de separar las palabras es por **letra mayúscula o guion bajo**

alert("Hola mundo JS!!")

let botonMascotaJugador = document.getElementById('boton-mascotas')

Se usa para poder manipular el DOM y seleccionar al botón que quiero que JS escuche: document.getElementById(‘id que quieras usar del html’)

1. **Una vez seleccionado el botón, debo hacer que se “escuche su evento” con el método ADDEVENTLISTENER.**
2. **Creamos una función para añadir al evento**, que nos permita seleccionar la mascota.

function seleccionarMascotaJugador() {

    alert("Seleccionaste tu mascota")

}

let botonMascotaJugador = document.getElementById('boton-mascotas')

botonMascotaJugador.addEventListener('click' , seleccionarMascotaJugador)

****

**Otra opción para hacer lo mismo es: window.addEventListener, poniendo la etiqueta <script> al principio del html** y no al final del código como en el anterior.

function iniciarJuego() {

    let botonMascotaJugador = document.getElementById('boton-mascotas')

    botonMascotaJugador.addEventListener('click' , seleccionarMascotaJugador)

}

function seleccionarMascotaJugador() {

    alert("Seleccionaste tu mascota")

}

window.addEventListener('load', iniciarJuego )

1. Hacer el **addEventListener** llamándolo **desde window**
2. En los parámetros poner **“load”**, para que escuche cuando ya **cargue el html**, así logramos que **no haya error en JS.**
3. Agregamos la función **“iniciarJuego”** y dentro de ella ponemos el código para que se escuche el evento del botón.

**Seleccionar mascota**

function seleccionarMascotaJugador() {

    let inputHipodoge = document.getElementById('Hipodoge')

    let inputCapipepo = document.getElementById('Capipepo')

    let inputRatigueya = document.getElementById('Ratigueya')

    let inputLangostelvis = document.getElementById('Langostelvis')

    let inputTucapalma = document.getElementById('Tucapalma')

    let inputPydos = document.getElementById('Pydos')

*if* (inputHipodoge.checked) {

        alert('Seleccionaste a Hipodoge')

    } *else* *if* (inputCapipepo.checked) {

        alert('Seleccionaste a Capipepo')

    } *else* *if* (inputRatigueya.checked) {

        alert('Seleccionaste a Ratigueya')

    } *else* *if* (inputLangostelvis.checked) {

        alert('Seleccionaste a Langostelvis')

    } *else* *if* (inputTucapalma.checked) {

        alert('Seleccionaste a Tucapalma')

    } *else* *if* (inputPydos.checked) {

        alert('Seleccionaste a Pydos')

    } *else* {

        alert('Debes seleccionar una mascota')

    }

}

1. **Establecemos variables** con el Id de cada personaje.
2. Luego agregamos **document.getElementById(‘ ’)** para que tome el Id del html
3. **Establecemos una condicional if** para la selección del primer personaje
4. **Establecemos else if** para todo lo que no cumpla con la primer condición.
5. **Else** para todo lo que no cumpla con las condiciones anteriores. En este caso, si no se selecciona una mascota no se puede proceder y se lo hacemos saber al jugador mediante un **alert(‘*Seleccionar mascota*’)**

**Método de manipulación del DOM**

Debemos manipular el DOM para que el jugador pueda efectivamente jugar y poder “atacar con su mascota elegida”.

* Para eso es necesario “**cambiar el html**” sus textos, entre otras cosas **con JavaScript, manipulando el DOM.**
* Debemos identificar lo que queremos cambiar del html con un id.
* Utilizamos la función **InnerHTML**

**Elección de mascotas por el enemigo**

Necesitamos manipular el DOM para que se elija una mascota de manera aleatoria para el enemigo.

1. Establecemos la function **Math.random** para poder elegir números al azar. (debajo de todo)
2. Después de la function “**seleccionarMascotaJugador”** declaramos **la function “ataqueEnemigo”**

* Dentro establecemos las variables “let” porque necesitamos llamar la función de aleatoriedad y innerHTML.

1. **Establecemos las condicionales**

* Comenzamos con **If**
* Seguimos con **else if**
* Finalizamos con **else** para todo lo demás

Lo hacemos estableciendo las condiciones para que se muestren los resultados. ***Ej: si elijo 1 == Hipodoge.***

1. **Dentro de las condicionales llamamos a la función innerHTML** para poder cambiar los nombres de las mascotas de ataque. Esto lo hacemos para manipular el DOM.

function seleccionarMascotaEnemigo() {

    let ataqueAleatorio = aleatorio(1,6)

    let spanMascotaEnemigo = document.getElementById('mascota-enemigo')

*if* (ataqueAleatorio == 1) {

        spanMascotaEnemigo.innerHTML = 'Hipodoge'

    } *else* *if* (ataqueAleatorio == 2) {

        spanMascotaEnemigo.innerHTML = 'Capipepo'

    } *else* *if* (ataqueAleatorio == 3) {

        spanMascotaEnemigo.innerHTML = 'Ratigueya'

    } *else* *if* (ataqueAleatorio == 4) {

        spanMascotaEnemigo.innerHTML = 'Langostelvis'

    } *else* *if* (ataqueAleatorio == 5) {

        spanMascotaEnemigo.innerHTML = 'Tucapalma'

    } *else* {

        spanMascotaEnemigo.innerHTML = 'Pydos'

    }

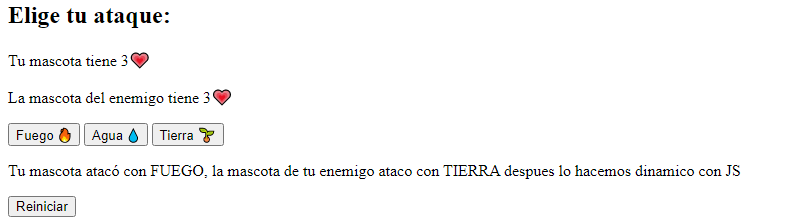
}

function aleatorio(*min*, *max*) {

*return* Math.floor(Math.random() \* (*max* - *min* + 1) + *min*)

}

**Selección del ataque**



**Tenemos 3 botones fuego, agua y tierra.**

1. **Debemos crear un evento - *addEventListener -*** para cada botón. Cuando le den click debe realizar alguna función.
2. Definimos las funciones necesarias para llevar a cabo la elección del ataque.
3. Debemos establecer variables globales por encima de nuestro código para poder establecer los ataques. **La llamaremos *let ataqueJugador***
4. Volvemos al html para poder afectar a cada uno de los botones de ataque con el addEventListener. Para esto debemos:

* **Crear variables para llamar los botones del html.**
* **Ponemos el addEventListener dentro de la función iniciarJuego**
* **El addEventListener lo ponemos dentro de las variables que llame dentro de la fx**

**Variable Global**

**Es aquella que está disponible a lo largo del script actual, es decir, dentro de todo el código JavaScript que se está escribiendo.**

**Una variable que se declara fuera de cualquier función se transforma en global, aun definiéndola con una palabra reservada (var, let, const) o no.**

* Se definen por encima de todo nuestro código de JS

let ataqueJugador   *//---> var global, la podemos llamar en cualquier fx*

*//tenemos que mandar a llamar un evento para cuando hagan click en el boton agua, tierra o fuego suceda algo.*

function iniciarJuego() {

    let botonMascotaJugador = document.getElementById('boton-mascotas')

    botonMascotaJugador.addEventListener('click' , seleccionarMascotaJugador)

    let botonFuego = document.getElementById('boton-fuego')-->creamos variables

    botonFuego.addEventListener('click', ataqueFuego)-->creamos el evento y establecemos una fx para llevar adelante el ataque.

    let botonAgua = document.getElementById('boton-agua')

    botonAgua.addEventListener('click', ataqueAgua)

    let botonTierra = document.getElementById('boton-tierra')

    botonTierra.addEventListener('click', ataqueTierra)

}

1. Creamos las funciones que establecimos en el AddEventListener

*//function ataque jugador*

function ataqueFuego(){

    ataqueJugador = 'FUEGO'

    alert(ataqueJugador)

}

function ataqueAgua(){

    ataqueJugador = 'AGUA'

    alert(ataqueJugador)

}

function ataqueTierra(){

    ataqueJugador = 'TIERRA'

    alert(ataqueJugador)

}

**Ataque Oponente**

//ataque oponente

function ataqueAleatorioEnemigo() {

    let ataqueAleatorio = aleatorio(1,3) llamo a la fx aleatorio.

    if (ataqueAleatorio == 1){

        ataqueOponente = 'Fuego'

    } else if (ataqueAleatorio == 2){

        ataqueOponente = 'Agua'

    } else {

        ataqueOponente = 'Tierra'

    }

}

Debemos hacer lo mismo que con la selección del ataque del jugador.

1. Establecemos una var global AtaqueOponente.
2. Luego creamos una función que haga que el ataque del enemigo sea aleatoria. Para esto llamaos a la fx aleatorio
3. Establecemos las condicionales
4. Por último, llamamos a la función ***ataqueAleatorioEnemigo* debajo del ataque del jugador.**

Esto lo hacemos para que se genere aleatoriamente inmediatamente al momento de que el jugador seleccione su ataque.

//function ataque jugador

function ataqueFuego(){

    ataqueJugador = 'FUEGO'

    ataqueAleatorioEnemigo()

}

function ataqueAgua(){

    ataqueJugador = 'AGUA'

    ataqueAleatorioEnemigo()

}

function ataqueTierra(){

    ataqueJugador = 'TIERRA'

    ataqueAleatorioEnemigo()

}

//ataque oponente

function ataqueAleatorioEnemigo() {

    let ataqueAleatorio = aleatorio(1,3)

    if (ataqueAleatorio == 1){

        ataqueOponente = 'Fuego'

    } else if (ataqueAleatorio == 2){

        ataqueOponente = 'Agua'

    } else {

        ataqueOponente = 'Tierra'

    }

}

//aleatoriedad

function aleatorio(min, max) {

    return Math.floor(Math.random() \* (max - min + 1) + min) //con esto damos aleatoriedad a la elección del personaje.

}

**Imprimiendo ataques del enemigo**

1. Crear una fx para mostrar las elecciones de ataques de los jugadores.
2. Establecemos dentro una var para agregar desde JS un párrafo en el HTML.
3. Aplicamos el InnerHtml para que se modifique en el DOM. Lo usamos para poder meter la información de los ataques seleccionados.

Hasta este momento aun no se ve reflejado en el juego. Solamente puedo ver desde console los resultados.

function crearMensaje() {  //llamamos a la fx para que nos arroje el mensaje de la eleccion de los ataques.

    let parrafo = document.createElement('p') //creamos una var y le asignamos crear un p en el html

    parrafo.innerHTML = 'Tu mascota atacó con' + ataqueJugador + 'La mascota del enemigo atacó con' + ataqueOponente + 'pediente de programar'

}

1. Creamos una variable y usamos **Document.createElement()**

En un documento HTML, el método **Document.createElement()** **crea un elemento HTML** especificado por su tagName.

1. Luego utilizamos la fx **appendChild**

**GANADORES – PERDEDORES - EMPATE**

//ataque oponente

function ataqueAleatorioEnemigo() {

    let ataqueAleatorio = aleatorio(1,3)

    if (ataqueAleatorio == 1){

        ataqueOponente = 'Fuego'

    } else if (ataqueAleatorio == 2){

        ataqueOponente = 'Agua'

    } else {

        ataqueOponente = 'Tierra'

    }

    combate() //sacamos la fx crear mensaje y ahora llamamos a la fx combate

}

function combate() {

    if(ataqueJugador == ataqueOponente) {

        crearMensaje("EMPATE")

    } else if(ataqueJugador == 'Fuego' && ataqueOponente == 'Tierra') {

        crearMensaje("GANASTE")

    } else if( ataqueJugador == 'Agua' && ataqueOponente == 'Fuego') {

        crearMensaje("GANASTE")

    } else if( ataqueJugador == 'Tierra' && ataqueOponente == 'Agua') {

        crearMensaje("GANASTE")

    } else {

        crearMensaje("PERDISTE")

    }

Para definir ganadores y perdedores debemos crear la **función combate** donde establecemos las condicionales de nuestro juego para ganar/perder/empatar.

* Dentro de ella llamamos a la **fx crearMensaje,** la que hará que el resultado aparezca en el html.

// ----------- mensaje para jugadores sobre el ataque-------------------

function crearMensaje(resultado) {  //llamamos a la fx para que nos arroje el mensaje de la eleccion de los ataques dentro de la fx ataqueAleatorioEnemigo.

    let sectionMensajes = document.getElementById('mensajes') // para tomar la section mensajes y poder poner el texto\*

    let parrafo = document.createElement('p') //creamos una var y le asignamos crear un p en el html

    parrafo.innerHTML = 'Tu mascota atacó con ' + ataqueJugador + '. La mascota del enemigo atacó con ' + ataqueOponente + ' -' + resultado //mensaje

    sectionMensajes.appendChild(parrafo) //\*ponemos el texto en la section parrafo.

}

**Contador de vidas**

* Debemos programar el contador de vidas dentro de la **function combate** que es donde se disputan la vidas de las mascotas.
* Creamos 2 variables, a las que vinculamos con **document.getElementById**
  + - spanVidasJugador
    - spanVidasEnemigo
* Luego debemos imprimir el mensaje de cuántas vidas vamos perdiendo. Pero para eso es necesario **contabilizar las vidas desde JS,** ya que solo las tenemos establecidas en html.
  + - Creamos 2 variables nuevamente pero estas van a ser globales (como en ataqueJugador/oponente) y establecemos con cuántas vidas comenzamos

let ataqueJugador   //---> var global, la podemos llamar en cualquier fx

let ataqueOponente

let vidasJugador = 3

let vidasEnemigo = 3

function combate() {

    let spanVidasJugador = document.getElementById('vidas-jugador')

    let spanVidasEnemigo = document.getElementById('vidas-enemigo')

    if(ataqueJugador == ataqueOponente) {

        crearMensaje("EMPATE")

    } else if(ataqueJugador == 'Fuego' && ataqueOponente == 'Tierra') {

        crearMensaje("GANASTE")

        vidasEnemigo-- // para restar vidas

        spanVidasEnemigo.innerHTML = vidasEnemigo //para mostrar el resultado

    } else if( ataqueJugador == 'Agua' && ataqueOponente == 'Fuego') {

        crearMensaje("GANASTE")

        vidasEnemigo--

        spanVidasEnemigo.innerHTML = vidasEnemigo

    } else if( ataqueJugador == 'Tierra' && ataqueOponente == 'Agua') {

        crearMensaje("GANASTE")

        vidasEnemigo--

        spanVidasEnemigo.innerHTML = vidasEnemigo

    } else {

        crearMensaje("PERDISTE")

        vidasJugador--

        spanVidasJugador.innerHTML = vidasJugador

    }

    revisarVidas() // llamamos la fx dentro de la fx combate.

}

//contador de vidas Luego esta fx la llamamos dentro de la fx combate.

function revisarVidas() {

    if (vidasEnemigo == 0) {

        alert("Ganaste!") // yo puse un alert para avisar que ganamos.

    } else if (vidasJugador == 0) {

        alert("Perdiste!")

    }

}

Dentro de la función combate debajo de cada mensaje “perdiste / ganaste ” debemos restar las vidas, ya sea al enemigo o al jugador.

¿Quién gana / pierde?

Necesitamos hacer una función que contabilice las vidas, y cuando detecte el primero que llegue a cero establezca un ganador y perdedor.

function revisarVidas() {

    if (vidasEnemigo == 0) {

        crearMensajeFinal("FELICITACIONES GANASTE🥳🥳")

    } else if (vidasJugador == 0) {

        crearMensajeFinal("QEPD PERDISTE😢😢")

    }

}

function crearMensajeFinal(resultadoFinal) {

    let sectionMensajes = document.getElementById('mensajes')

    let parrafo = document.createElement('p')

    parrafo.innerHTML = resultadoFinal

    sectionMensajes.appendChild(parrafo)

}

**BOTON DE REINICIO DEL JUEGO**

* Necesito hacer que el juego deshabilite los botones de ataque cuando la vida de un jugador llegue a cero.
* Habilitar el botón reiniciar juego.

1. Para “escuchar al botón” cuando hagan click debo **usar addEventListener**

Esto lo debemos ubicar dentro de la **fx iniciarJuego**

* Establecemos una var botonReiniciar, el cual lo vinculamos con el botón del html. Para poder seleccionar el botón.

1. Ahora hay que crear el escuchador de eventos.  
    **(evento)**

Llamamos al botonReiniciar y le ponemos el método addEventListener**(‘click’**, fx que hará que se reinicie el juego)

function iniciarJuego() {

    let botonMascotaJugador = document.getElementById('boton-mascotas')

    botonMascotaJugador.addEventListener('click' , seleccionarMascotaJugador)

    //-----------------botones de ataques-------------------------------------------

    let botonFuego = document.getElementById('boton-fuego')//establecemos variables

    botonFuego.addEventListener('click', ataqueFuego)//las escuchamos con el addEventListener

    let botonAgua = document.getElementById('boton-agua')

    botonAgua.addEventListener('click', ataqueAgua)

    let botonTierra = document.getElementById('boton-tierra')

    botonTierra.addEventListener('click', ataqueTierra)

    //---------------boton reiniciar ------------------------------------

    let botonReiniciar = document.getElementById('boton-reiniciar')

    botonReiniciar.addEventListener('click', reiniciarJuego)

}

1. Crear la función que llamamos en el escuchador**. Fx reiniciarJuego**

* Dentro de ella vamos a llamar al **método location.reload,** hace referencia a la ubicación que nos encontramos ej: <https://platzi.com/clases/3208-programacion-basica/52006-reiniciando-el-juego/>
* **El metodo location. reload() carga de nuevo la URL actual**, como lo hace el botón de Refresh de los navegadores.

//------------fx boton reiniciar juego-----------------------------

// todo lo programado dentro de las llaves se va a ejecutar cuando haga click en el boton.

function reiniciarJuego() {

    location.reload()

// (location)es un método ---- reload es una fx, la cual permite recargar url.

}

**Finalizar el juego cuando las vidas lleguen a 0**

Cuando las vidas llegan a cero el juego debe terminar, para eso debemos **deshabilitar los botones de ataque del html con el atributo disabled.**

* Se puede poner el atributo en el html, dentro del boton. Pero quedaría estático. Por eso lo hacemos desde JS.
* Se deben deshabilitar los botones una vez que aparezca el mensaje de “ganaste/perdiste”
* Por eso lo hacemos dentro de la **fx crearMensajeFinal**.

function crearMensajeFinal(resultadoFinal) {

    let sectionMensajes = document.getElementById('mensajes')

    let parrafo = document.createElement('p')

    parrafo.innerHTML = resultadoFinal

    sectionMensajes.appendChild(parrafo)

    let botonFuego = document.getElementById('boton-fuego')//establecemos variables

    botonFuego.disabled = true //deshabilitamos los botones cuando termine el juego, es decir, cuando aparece el mensaje final.

    let botonAgua = document.getElementById('boton-agua')

    botonAgua.disabled = true

    let botonTierra = document.getElementById('boton-tierra')

    botonTierra.disabled = true

}

**CSS + JS MOKEPON**

En nuestro juego dimos estilos a la primer sección.

La segunda que es la de seleccionar ataque, debemos ordenar todos nuestros elementos de acuerdo a nuestro diseño con contenedores padres e hijos.

<section id="seleccionar-ataque">

        <h2 class="ataque-tittle">Elige tu ataque:</h2>

        <div class="botones-ataque">

            <button id="boton-fuego"> Fuego 🔥</button>

            <button id="boton-agua"> Agua 💧</button>

            <button id="boton-tierra"> Tierra 🌱</button>

        </div>

        <div class="section-mensajes-reiniciar">

            <section id="mensajes">

            </section>

            <section id="reiniciar">

                <button id="boton-reiniciar">Reiniciar</button>

            </section>

        </div>

        <div class="vidas-mascota">

            <p>Tu mascota <span id="mascota-jugador"></span> tiene <span id="vidas-jugador">3</span>💗</p>

            <p>La mascota del enemigo <span id="mascota-enemigo"></span> tiene  <span id="vidas-enemigo">3</span>💗</p>

        </div>

 </section>

* Necesitamos poner display flex para el primer contenedor.
* Para que se vea reflejado debo cambiar en el JS que en esta section habíamos puesto display block

//---------------------seleccion de mascotas-------------------

function seleccionarMascotaJugador() {

    let sectionSeleccionarAtaque = document.getElementById('seleccionar-ataque')

    sectionSeleccionarAtaque.style.display = 'block' // para mostrar la section seleccion ataque que habiamos ocultado.