Clase grabada

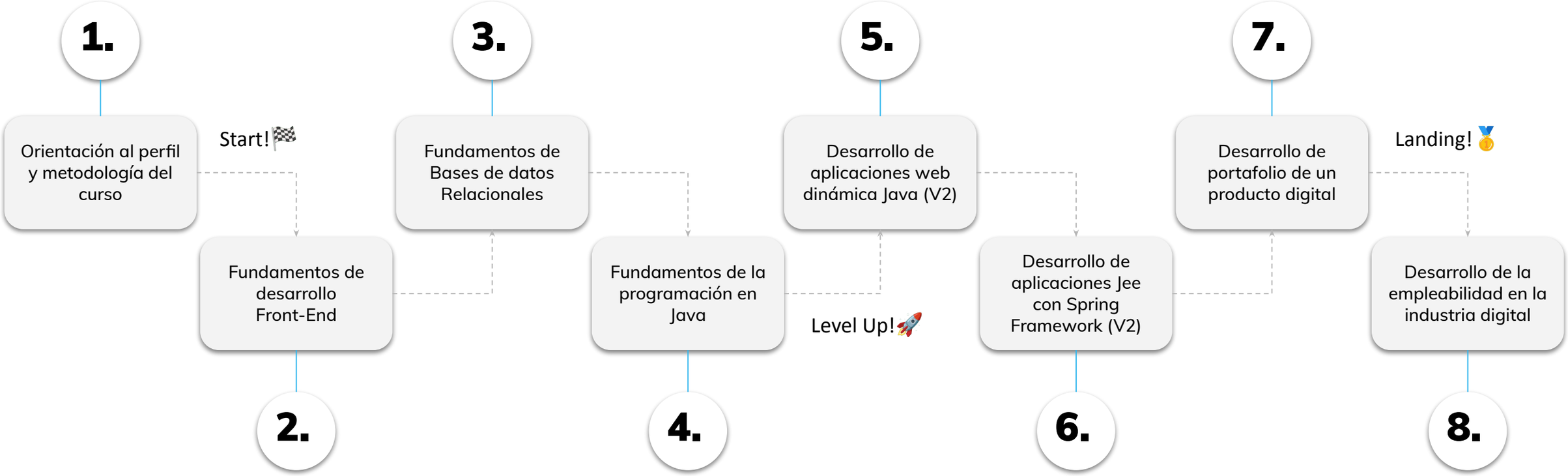
Bases del lenguaje Javascript

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

English Always | alkemy

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



English Always | alkemy

REPASO CLASE ANTERIOR

En la clase anterior trabajamos 📚:

- ✓ *Manipular el Document Object Model (DOM) de una página web utilizando JavaScript*
- ✓ *Seleccionar elementos HTML en una página utilizando selectores como getElementById y querySelector.*
- ✓ *Manipulación de contenido HTML mediante propiedades como innerHTML*

LEARNING PATHWAY

¿Sobre qué temas trabajaremos?

2. Start! 🏁

FUNDAMENTOS DE DESARROLLO FRONT-END → **Bases del lenguaje Javascript** → **Actividad N° 1: Evento Submit**

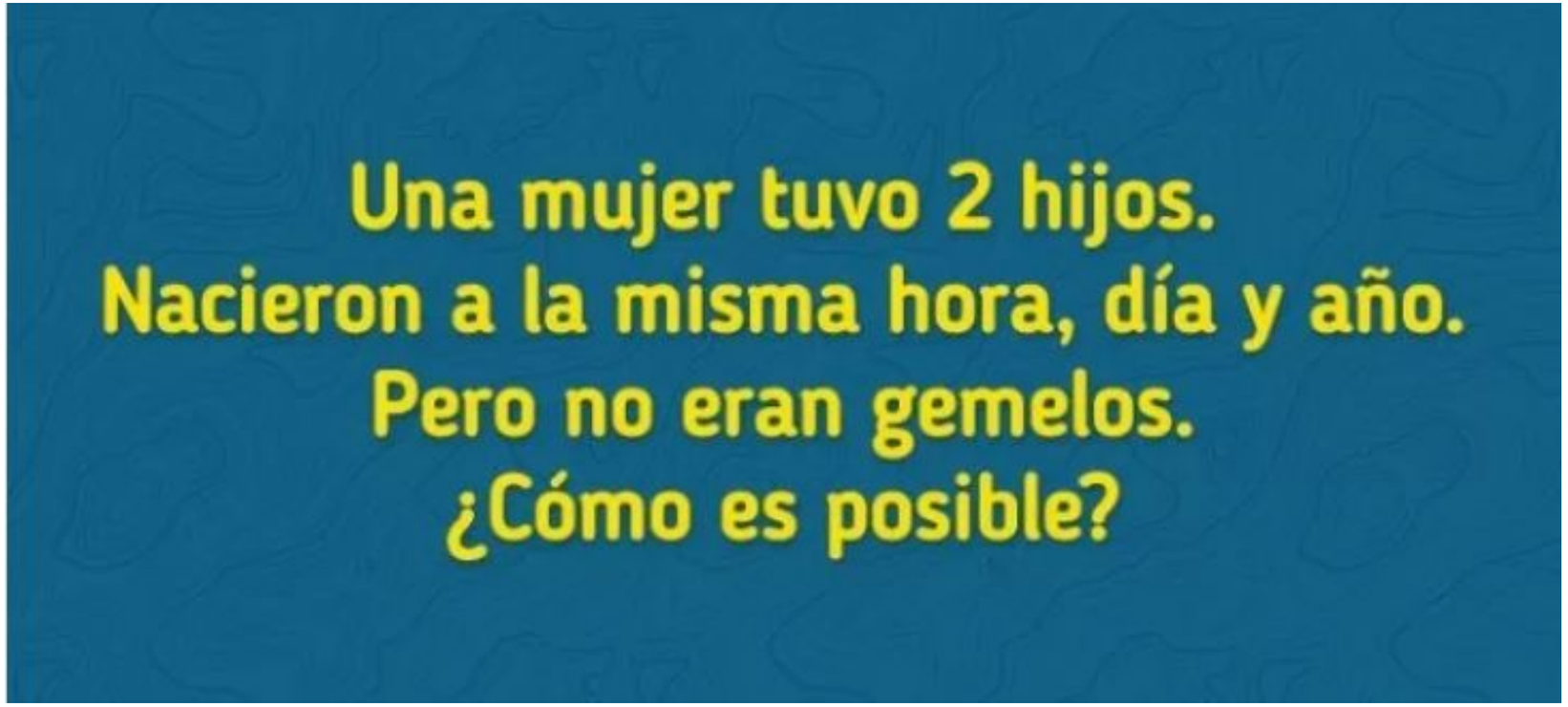
En esta lección, exploraremos eventos en la web. Aprenderás a crearlos en HTML y JavaScript, desencadenando acciones con onclick, onsubmit y onchange para interactuar con los usuarios y hacer tu sitio más dinámico.

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?

- *Comprender cómo crear eventos interactivos en páginas web utilizando HTML y JavaScript.*
- *Aprender cómo los eventos permiten a los usuarios interactuar con tu sitio web.*
- *Conocer el uso de atributos como onclick, onsubmit, y onchange para desencadenar acciones específicas.*

Rompehielo🥶

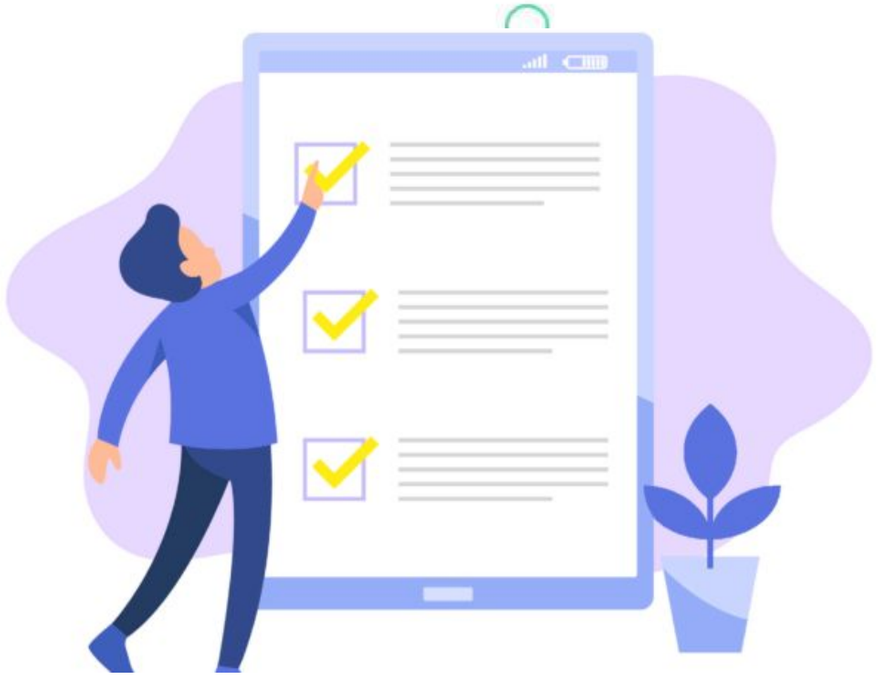

Una mujer tuvo 2 hijos.
Nacieron a la misma hora, día y año.
Pero no eran gemelos.
¿Cómo es posible?

Eventos básicos

Eventos básicos

Los eventos son acciones o sucesos que ocurren en el navegador, como hacer clic en un elemento, ingresar texto en un campo de entrada o cambiar el valor de una casilla de verificación. JavaScript nos permite manejar estos eventos y ejecutar código en respuesta a ellos.

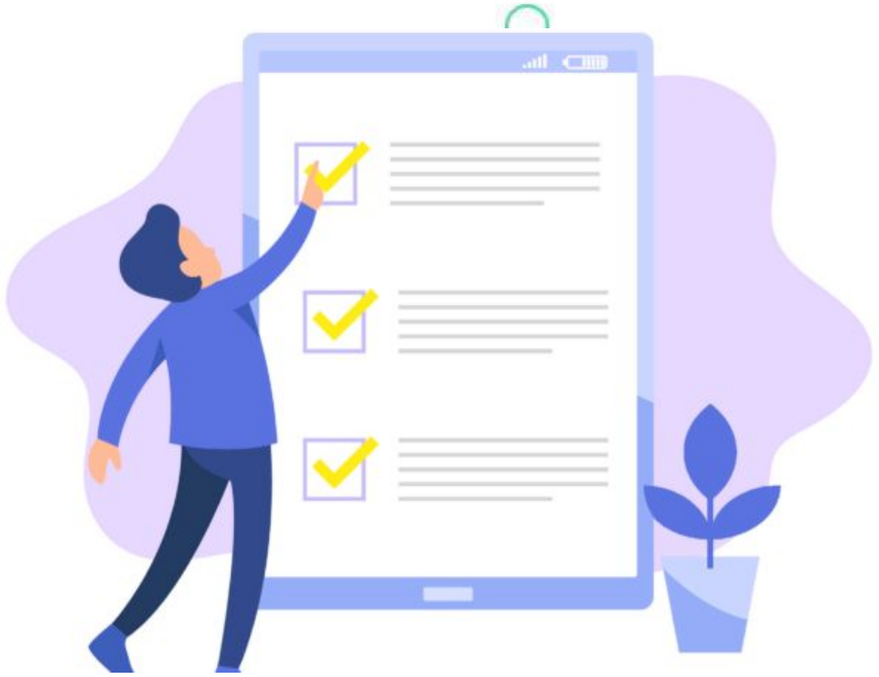
JavaScript permite asignar una función a cada uno de los eventos. Reciben el nombre de event handlers o manejadores de eventos. Así, ante cada evento, JavaScript asigna y ejecuta la función asociada al mismo. Hay que entender que los eventos suceden constantemente en el navegador.



Eventos básicos

Definir eventos en JS

El método addEventListener() permite definir qué evento escuchar sobre cualquier elemento seleccionado.
El primer parámetro corresponde al nombre del evento y el segundo a la función de respuesta.



Eventos básicos

El método addEventListener() es utilizado para asignar un evento a un elemento específico en JavaScript. Permite definir qué evento escuchar y qué función se debe ejecutar como respuesta a ese evento.

```
<button id="mi-boton">Haz clic aquí</button>


const boton = document.getElementById('mi-boton');
boton.addEventListener('click', function() {
  console.log('¡Hola, mundo!');
});
```

Eventos básicos

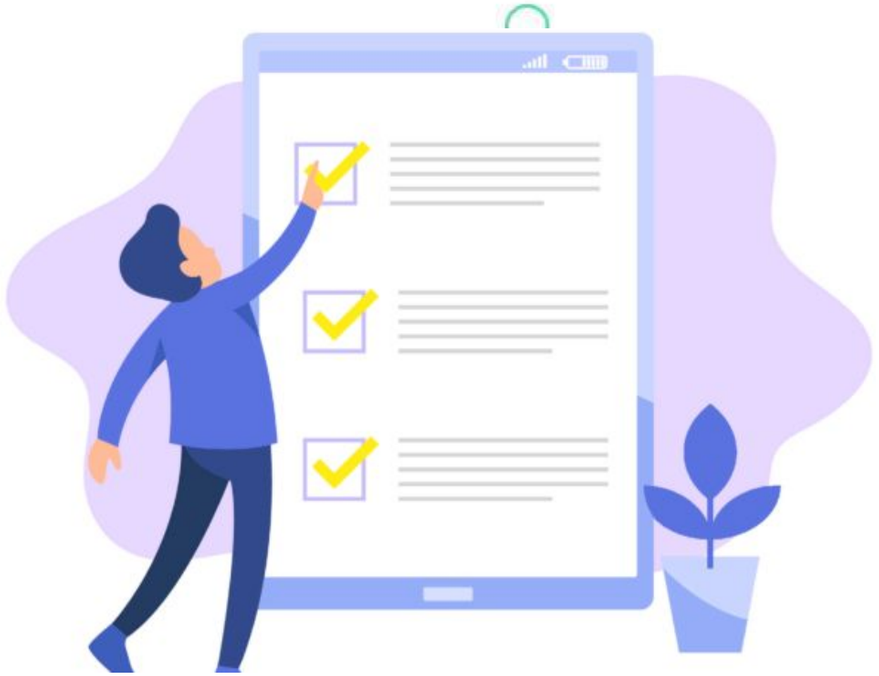
La ventaja de utilizar addEventListener() es que puedes asignar múltiples eventos a un mismo elemento y manejarlos de forma separada.
también puedes utilizar la sintaxis de funciones flecha para definir la función de respuesta de manera más concisa.

```
<button id="mi-boton">Haz clic aquí</button>


boton.addEventListener('click', () => {
  console.log('¡Hola, mundo!');
});
```

Eventos básicos

Definir eventos en JS

Otra forma de definir eventos en JavaScript es utilizando las propiedades de los nodos para asignar las respuestas a los eventos. Estas propiedades se identifican con el nombre del evento precedido por el prefijo "on".



Eventos básicos

En este ejemplo, hemos seleccionado un elemento de botón con el ID "mi-boton" utilizando getElementById(). Luego, hemos asignado la función anónima como respuesta al evento onclick utilizando la propiedad onclick del botón. Cuando se haga clic en el botón, se imprimirá el mensaje "¡Hola, mundo!" en la consola.

```
<button id="mi-boton">Haz clic aquí</button>


const boton = document.getElementById('mi-boton');
boton.onclick = function() {
  console.log('¡Hola, mundo!');
};
```

Eventos básicos

También puedes utilizar funciones flecha para definir la respuesta de manera más concisa.
Esta forma de asignar eventos utilizando las propiedades del nodo puede ser útil en ciertos casos.
Sin embargo, ten en cuenta que al utilizar esta sintaxis, si asignas un nuevo manejador de eventos a la misma propiedad, el manejador anterior se reemplazará.

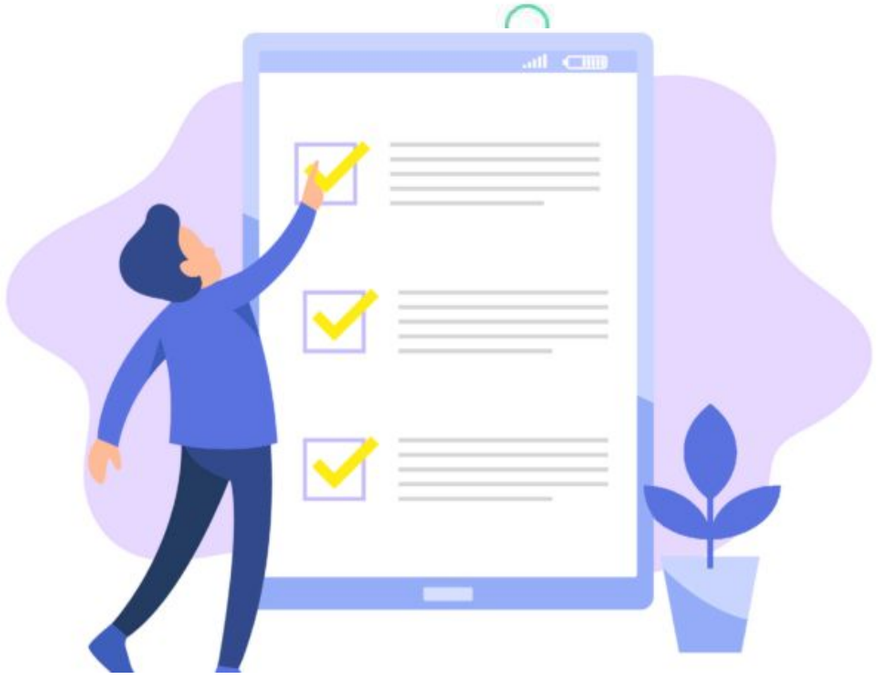
```
<button id="mi-boton">Haz clic aquí</button>


const boton = document.getElementById('mi-boton');

boton.onclick = () => {
  console.log('¡Hola, mundo!');
};
```

Eventos básicos

Evento Clic y Doble Clic

El evento clic (click) se activa cuando se hace clic en un elemento, ya sea mediante un clic del mouse o una pulsación táctil en dispositivos móviles.

El evento doble clic (dblclick) se activa cuando se realiza un doble clic en un elemento, es decir, dos clics rápidos consecutivos.



Eventos básicos

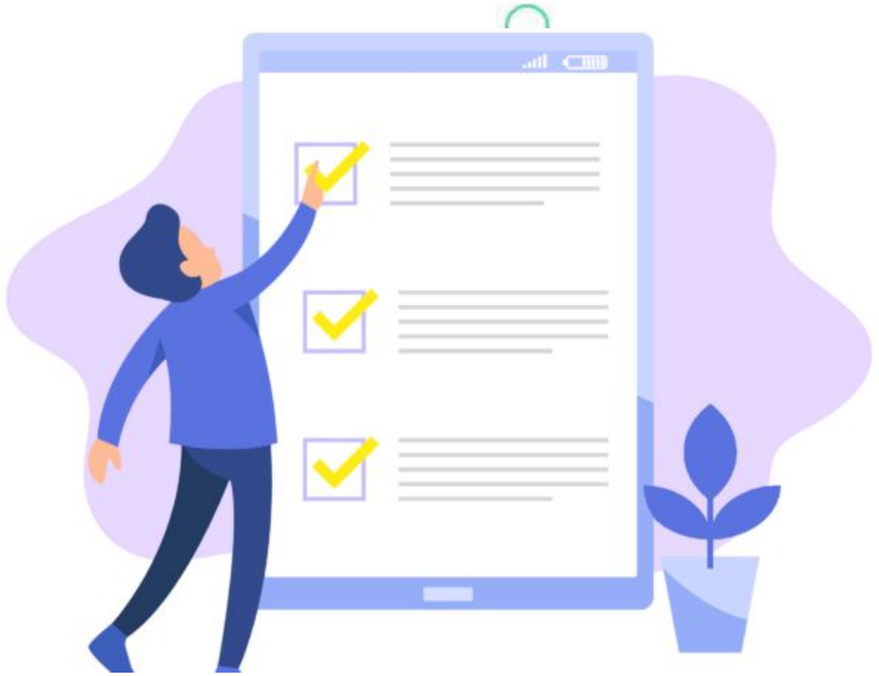
Cuando se hace clic en el botón, se activa el evento click y se muestra un mensaje en la consola. Cuando se realiza un doble clic en el botón, se activa el evento dblclick y se muestra otro mensaje en la consola.

```
<button id="myButton">Haz clic o doble clic aquí</button>

const button = document.getElementById('myButton');
button.addEventListener('click', () => {
  console.log('Has hecho clic en el botón');
});
button.addEventListener('dblclick', () => {
  console.log('Has hecho doble clic en el botón');
});
```

Eventos básicos

Eventos del mouse

Se producen por la interacción del usuario con el mouse. Entre ellos se destacarán los que se encuentran a continuación.



Eventos básicos

mousemove: El movimiento del mouse sobre el elemento activa el evento.

click: Se activa después de mousedown o mouseup sobre un elemento válido.

mousedown/mouseup: Se oprime/suelta el botón del ratón sobre un elemento.

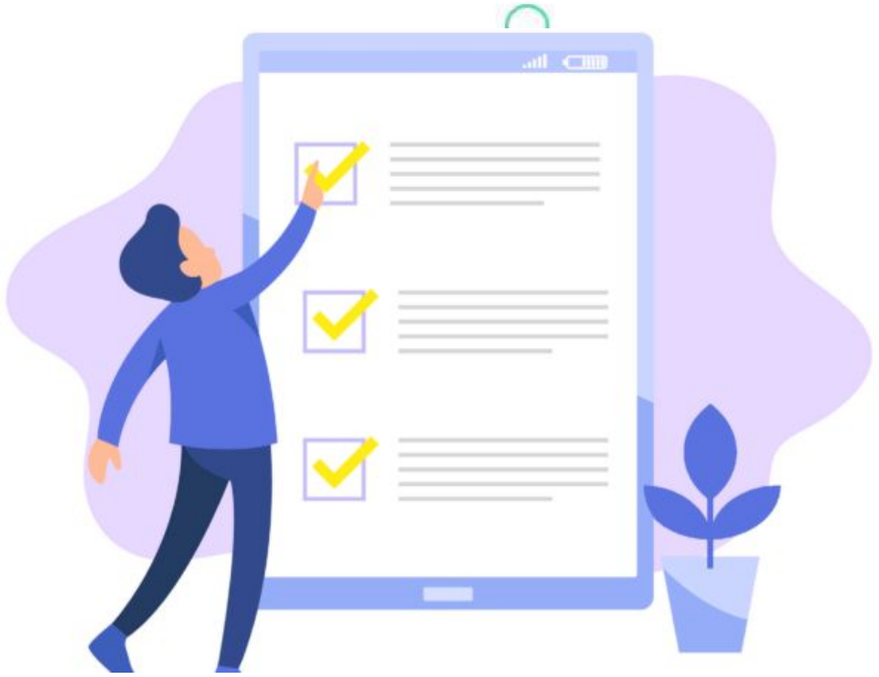
mouseover/mouseout: El puntero del mouse se mueve sobre/sale del elemento.

```
//CODIGO HTML DE REFERENCIA
<button id="btnMain">CLICK</button>
//CODIGO JS
let boton = document.getElementById("btnMain")
boton.onclick = () => {console.log("Click")}
boton.onmousemove = () => {console.log("Move")}
```

Eventos básicos

Eventos del teclado

Se producen por la interacción del usuario con el teclado. Entre ellos se destacarán los que se encuentran a continuación.



Eventos básicos

keydown: Cuando se presiona.

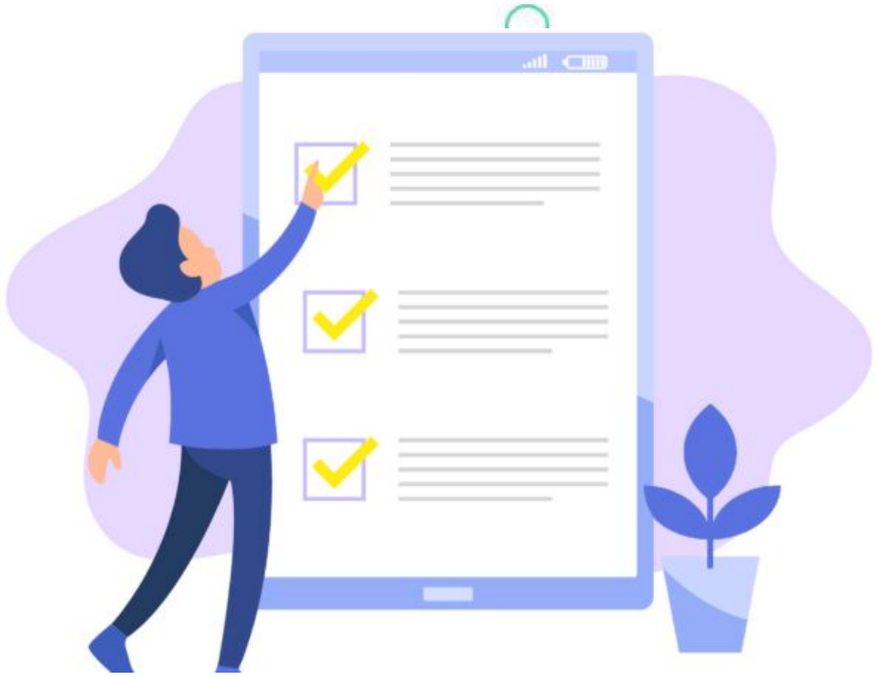
keyup: Cuando se suelta una tecla.

```
//CODIGO HTML DE REFERENCIA
<input id = "nombre" type="text">
<input id = "edad"   type="number">
//CODIGO JS
let input1 = document.getElementById("nombre")
let input2 = document.getElementById("edad")
input1.onkeyup = () => {console.log("keyUp")}
input2.onkeydown = () => {console.log("keyDown")}
```

Eventos básicos

Evento change

El evento change se activa cuando se detecta un cambio en el valor de un elemento, pero no se dispara inmediatamente mientras se está escribiendo en un campo de texto. En cambio, se activa cuando el elemento pierde el foco, es decir, cuando el usuario realiza alguna acción para mover el foco a otro elemento, como hacer clic fuera del campo de texto o presionar la tecla "Tab" para cambiar al siguiente campo.



Eventos básicos

Este evento es útil cuando se necesita detectar cambios en el valor de un elemento, como un input de texto o un select, después de que el usuario ha finalizado su interacción con el elemento.

```
<input type="text" id="inputText">

const input = document.getElementById('inputText');

input.addEventListener('change', (event) => {
  const newValue = event.target.value;
  console.log('Nuevo valor:', newValue);
  // Realizar acciones adicionales con el nuevo valor
});
```

Eventos básicos

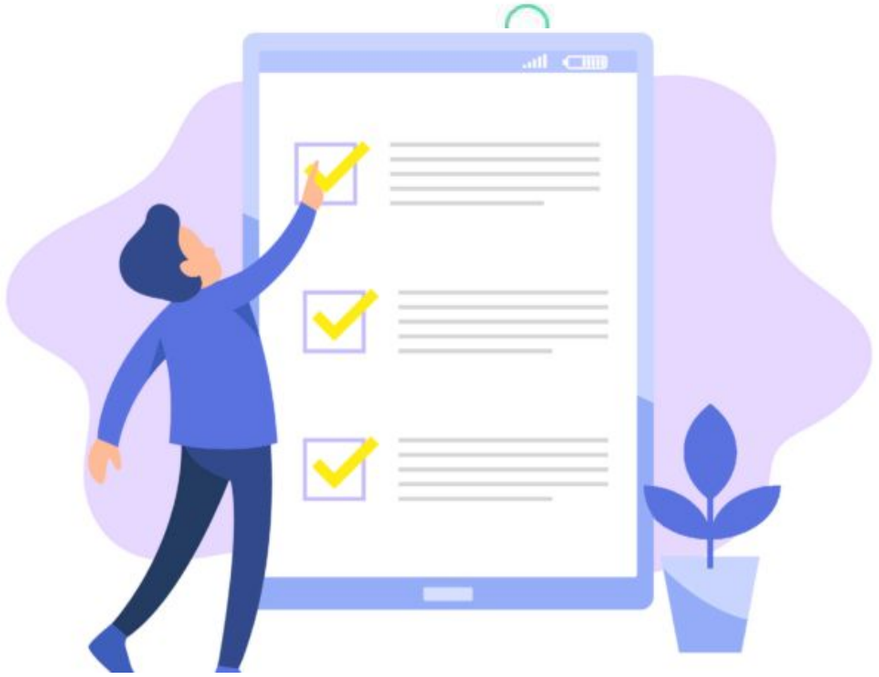
El evento change solo se activa cuando hay un cambio real en la selección y se confirma, es decir, cuando se elige una opción diferente. Si el usuario selecciona la misma opción nuevamente, no se activará el evento change.

```
<select id="selectOptions">
  <option value="option1">Opción 1</option>
  <option value="option2">Opción 2</option>
  <option value="option3">Opción 3</option>
</select>

const select = document.getElementById('selectOptions');
select.addEventListener('change', (event) => {
  const selectedValue = event.target.value;
  console.log('Valor seleccionado:', selectedValue);
// Realizar acciones adicionales con el valor seleccionado
});
```

Eventos básicos

Evento Input

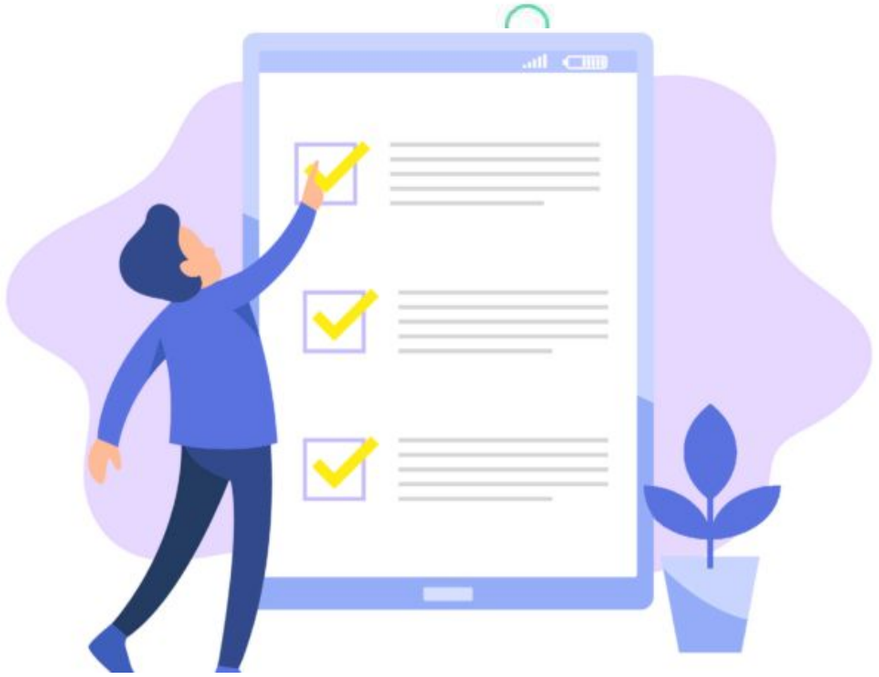
El evento input se activa cada vez que se realiza un cambio en el valor de un elemento de entrada, como un campo de texto o un área de texto. A diferencia del evento change, el evento input se dispara inmediatamente mientras el usuario está interactuando con el elemento, incluso mientras está escribiendo o eliminando texto.



Eventos básicos

Evento Input

El evento input se dispara en cada cambio, por lo que puede haber una gran cantidad de eventos generados mientras el usuario está interactuando con el campo de texto. Si necesitas realizar acciones que no requieren una respuesta inmediata, como realizar una búsqueda o una solicitud a un servidor, es posible que desees combinar el evento input con un temporizador o utilizar otro enfoque para evitar realizar múltiples solicitudes innecesarias.



Eventos básicos

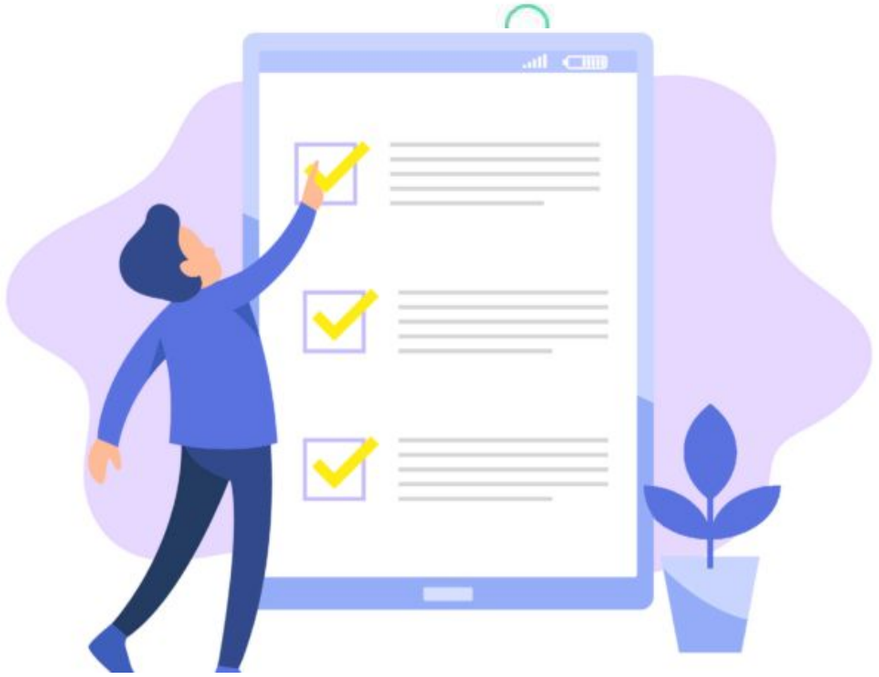
El evento input es útil cuando se necesita detectar cambios en tiempo real a medida que el usuario escribe o realiza acciones de edición en un campo de texto.

```
<input type="text" id="inputText">


  const input = document.getElementById('inputText');


  input.addEventListener('input', (event) => {
    const currentValue = event.target.value;
    console.log('Valor actual:', currentValue);
    // Realizar acciones adicionales con el valor actual
  });
```

Eventos básicos

Evento Submit

El evento submit se activa cuando se envía un formulario, ya sea haciendo clic en un botón de envío o presionando la tecla Enter dentro de un campo de entrada. Este evento se utiliza comúnmente para capturar y procesar los datos ingresados por el usuario antes de enviarlos al servidor.



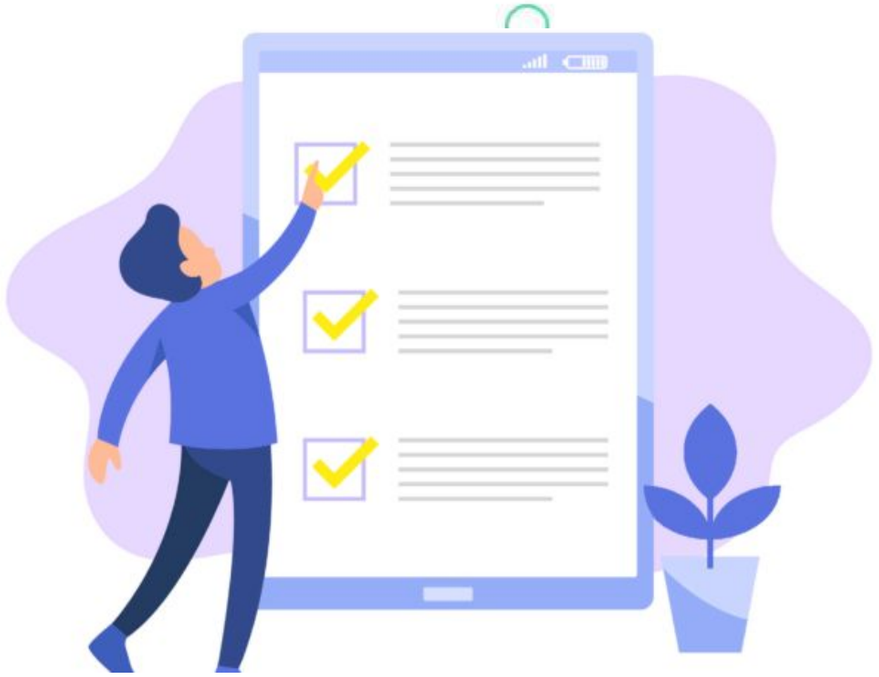
Eventos básicos

```
<form id="myForm">
  <label for="name">Nombre:</label>
  <input type="text" id="name" required>
  <button type="submit">Enviar</button>
</form>
  const form = document.getElementById('myForm');
  form.addEventListener('submit', (event) => {
    event.preventDefault();
 // Evitar el envío del formulario por defecto
// Obtener los valores de los campos de entrada
const name = document.getElementById('name').value;
// Realizar acciones adicionales con los datos ingresados
 console.log('Nombre:', name);
// Hacer algo más con los datos (enviar a un servidor, procesarlos, etc.)
});
```

Eventos básicos

Evento Submit

La llamada a event.preventDefault() evita que el formulario se envíe de forma predeterminada, lo que permite realizar acciones personalizadas antes de enviar los datos. puedes usar esta función para realizar validaciones adicionales, mostrar mensajes de error o realizar cualquier otra acción necesaria antes de enviar los datos.



LIVE CODING

Ejemplo en vivo

Eventos de Clic y Doble Clic en JavaScript

Eres un instructor de programación y estás realizando una sesión en vivo para enseñar a los estudiantes cómo trabajar con eventos de clic y doble clic en JavaScript. El objetivo es que los estudiantes comprendan cómo responder a estas interacciones del usuario y cómo implementar acciones específicas en respuesta a eventos de clic y doble clic.

Consigna:

Realizar un live coding para demostrar cómo trabajar con eventos de clic y doble clic en JavaScript, permitiendo a los estudiantes interactuar con elementos HTML y ejecutar acciones según la interacción del usuario.

Tiempo: 10 minutos

LIVE CODING

Ejemplo en vivo

- **Agregar Evento de Clic**

- Define una función llamada manejarClic en JavaScript que se ejecutará cuando el elemento sea clickeado.
- Asocia esta función al evento de clic del elemento HTML. Esto se puede hacer utilizando el método addEventListener().
- Dentro de la función manejarClic, muestra un mensaje en la consola como "Clic detectado".

- **Agregar Evento de Doble Clic**

- Define una función llamada manejarDobleClic en JavaScript que se ejecutará cuando el elemento sea doble clickeado.
- Asocia esta función al evento de doble clic del elemento HTML utilizando el método addEventListener().
- Dentro de la función manejarDobleClic, muestra un mensaje en la consola como "Doble clic detectado".

Evaluación Integradora ✨

Trabajo Integrador del Módulo 💪🏽

Iremos completándolo progresivamente clase a clase

CLICK AQUÍ
para ver la consigna completa

Ejercicio

Evento submit

Evento Submit

Breve descipción

Contexto: 🙌

Seguimos avanzando en nuestra wallet. El objetivo es implementar un proceso de autenticación seguro y cómo gestionar el acceso a la wallet.

Consigna: ✍️

Crear un sistema de inicio de sesión para la wallet digital.

Tiempo🕐: 10 minutos

Evento Submit

Breve descipción

Paso a paso: ⚙️

1. Define una función llamada verificarCredenciales en JavaScript que se ejecutará cuando el formulario se envíe.
2. Dentro de esta función, compara el nombre de usuario y la contraseña ingresados con valores predeterminados (por ejemplo, "usuario123" y "claveSecreta").
3. Si las credenciales son correctas, permite el acceso a la wallet y muestra un mensaje de "Inicio de sesión exitoso". Si no son correctas, muestra un mensaje de "Credenciales incorrectas".
4. Asocia la función verificarCredenciales al evento de envío del formulario.
5. Intenta iniciar sesión con credenciales correctas e incorrectas para verificar si se muestra el mensaje adecuado.

Evento Submit

```
UTF-8">
wport" content="width=device-width, initial-scale=1.0">
igital</title>


esión</h1>
-form">
"username">Nombre de Usuario:</label>
="text" id="username" required><br><br>
"password">Contraseña:</label>
="password" id="password" required><br>aula
e="submit">Iniciar Sesión</button>

e"></div>
```

```
    // Paso 1: Validación de Credenciales
     function verificarCredenciales(event) {
         event.preventDefault();
  const usuarioIngresado = document.getElementById("username").value;
  const contrasenaIngresada = document.getElementById("password").valu

         // Valores predeterminados (pueden ser reemplazados)
         const usuarioValido = "usuario123";
         const contrasenaValida = "claveSecreta";

 if (usuarioIngresado === usuarioValido && contrasenaIngresada === con
   document.getElementById("message").textContent = "Inicio de sesión
         } else {
    document.getElementById("message").textContent = "Credenciales in
         }
      }
      // Paso 2: Asociar la Función al Formulario
document.getElementById("login-form").addEventListener("submit", verif
```

¿Alguna consulta?

RESUMEN

¿Qué logramos en esta clase?

- ✓ *Comprender cómo crear eventos interactivos en páginas web utilizando HTML y JavaScript.*
- ✓ *Aprender cómo los eventos permiten a los usuarios interactuar con tu sitio web.*
- ✓ *Conocer el uso de atributos como onclick, onsubmit, y onchange para desencadenar acciones específicas.*

#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 👇👇👇

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
   a. Material 1: Lección 5: Bases del lenguaje Javascript: 49-57
   b. Material 2: Working Time de la Lección N° 5
   c. Material 3: Ponte a Prueba de la Lección N° 5
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 👋🏻

English Always | alkemy

Momento:

Time-out!

⏰5 min.