



16

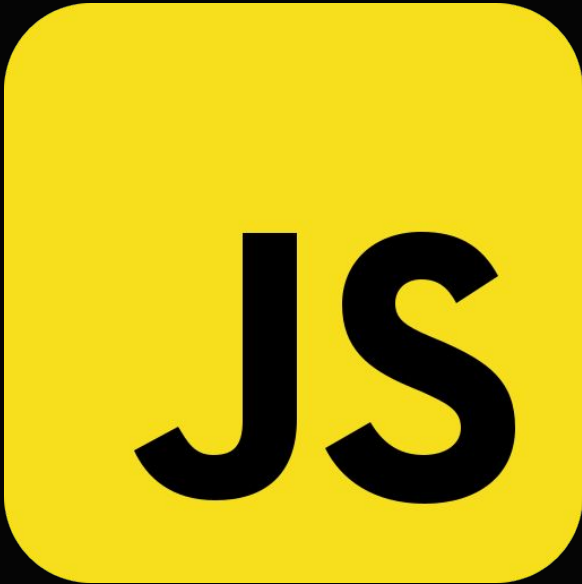
Formularios

Yael Ruiz Pucheta

Temas puntuales de la sección



- ¿Qué es un formulario?
- Etiqueta Form
- Estructura de un formulario
- Obtener el valor de un input
- Obtener Valores de todo el formulario a través del método submit



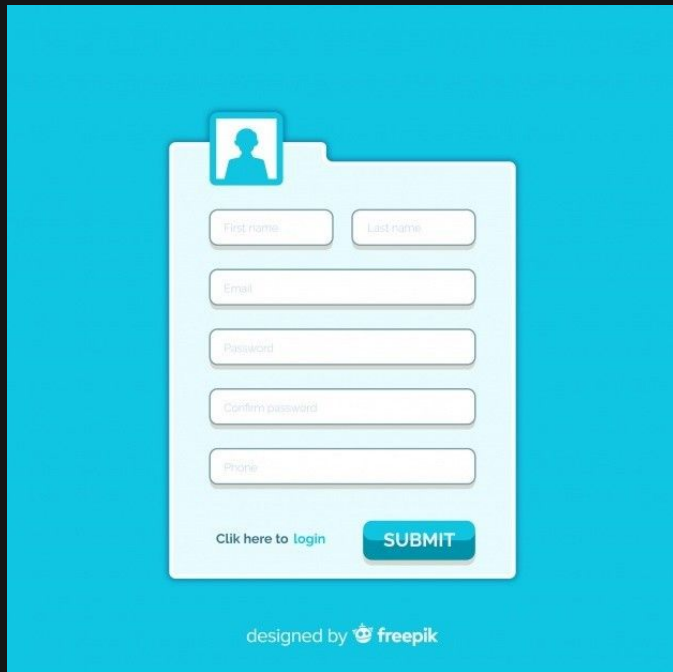
JS

Formulario

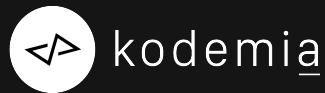
Formularios

Los formularios es aquel documento **físico o digital** utilizado para que el usuario nos ingrese **datos relevantes** para nuestra aplicación.

Este elemento (formulario) es uno de los principales puntos de interacción entre un usuario y un sitio web o aplicación



Formularios



```
<form>
```

```
[Conjunto de controles]
```

```
</form>
```

Dentro de nuestro Html la etiqueta **form** funcionará como un **contenedor o Wrap de inputs**, cada uno pensado para **recolectar información** ingresada por los usuarios, dicha información puede ser **texto, fechas, contraseñas, archivos**, entre más elementos.

Estructura de un formulario

Estructura de un formulario

```
<form action="" method="post">  
  <input type="text">  
</form>
```

- **Form:** Etiqueta html para determinar qué es un formulario
- **Action:** Atributo que especifica la ruta o en otros casos el url para procesar el formulario.
- **Method:** Método o Verbo http se usará para enviar el conjunto de valores contenidos en los inputs

Estructura de un formulario



Si bien la estructura general de un formulario dicta a usar el atributo `action` y `method`, en la actualidad ya no son manipulados de esa forma delegando el tratamiento de los datos por medio de Javascript

Estructura de un formulario

```
<form action="" method="post">  
  <input type="submit" />  
  <button type="Submit">Enviar</button>  
</form>
```



Submit

Un elemento de tipo **input** y **button** que teniendo el valor "**submit**" en su atributo **type**, representa un botón que, cuando es presionado, **envía** el formulario (form) al que pertenece.

Estructura de un formulario

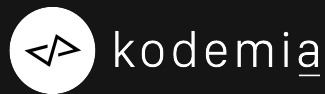
```
<form action="" method="post">
  <input type="reset" />
  <button type="reset">Limpiar</button>
</form>
```



Reset

Al igual que submit, puede ser cualquier elemento **input** o **button** con el atributo **type con el valor de reset**, su comportamiento es **restaurar** los elementos de un formulario a sus valores por defecto.

Estructura de un formulario



```
const formulario = document.querySelector('form');

formulario.addEventListener('reset', (e) => {
  // ... acciones
});

formulario.addEventListener('submit', (e) => {
  // ... acciones
});
```

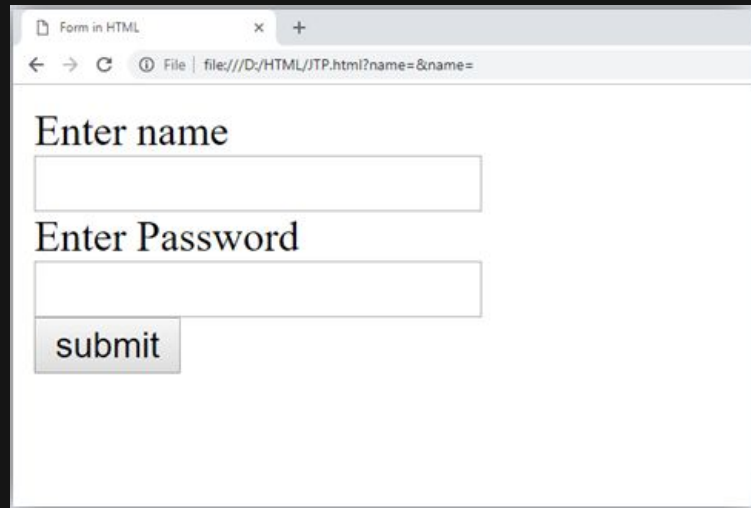
Si bien todo puede ser referenciado / manipulado a nivel del Html, para darle **mayores acciones o cambiar su comportamiento** es posible modificarlo mediante javascript, añadiendo **escuchadores** al evento.

Obtener el valor de un input

Obtener el valor de un input

El elemento input representa un campo de datos (Texto, archivos, booleanos, etc.

Este elemento es capaz de proveer muchos **tipos diferentes**, de acuerdo al valor que presente en el atributo **type**.



The screenshot shows a web browser window with the title "Form in HTML". The address bar displays the file path "file:///D:/HTML/JTP.html?name=&name=". The form contains two text input fields. The first field is labeled "Enter name" and the second field is labeled "Enter Password". Below the input fields is a button labeled "submit".

Obtener el valor de un input

```
// html
<form >
  <input id="mensaje" type="text" >
</form>

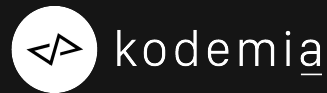
//js

const mensaje = document.querySelector('#mensaje')

console.log(mensaje.value);
```

Para obtener el valor de un input en javascript, es necesario tener la referencia al elemento y posteriormente acceder a la propiedad value

Obtener el valor de un input



```
// html
<form >
    <input id="mensaje" type="text" >
</form>

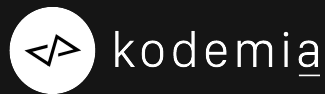
//js

const mensaje = document.querySelector('#mensaje')

console.log(mensaje.value);
```

Obtener valores de un formulario mediante Submit

Obtener valores de un formulario



Debemos obtener la referencia al formulario, posteriormente añadir un listener / escuchador al evento submit

```
// html

<form id="formulario">
  <input name="asunto" type="text" >
  <input name="mensaje" type="text" >
  <button type="submit">Enviar</button>
</form>

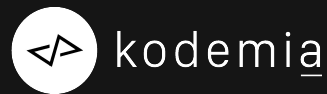
//js

const formulario = document.getElementById('formulario');

formulario.addEventListener('submit', (e) => {

});
```

Obtener valores de un formulario



```
// html

<form id="formulario">
  <input name="asunto" type="text" >
  <input name="mensaje" type="text" >
  <button type="submit">Enviar</button>
</form>

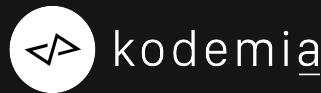
//js

const formulario = document.getElementById('formulario');

formulario.addEventListener('submit', (e) => {

});
```

Obtener valores de un formulario

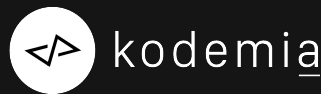


Dentro del listener del método submit, se debe llamar al evento `preventDefault`, contenido en el argumento event (e).

Este método es invocado para cancelar su comportamiento original y así evitar que se recargue la página.

```
const formulario = document.getElementById('formulario');  
  
formulario.addEventListener('submit', (e) => {  
  e.preventDefault();  
});
```

Obtener valores de un formulario

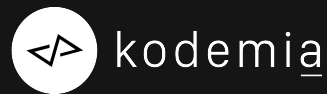


Debemos obtener la referencia al formulario a través de la propiedad `target` del evento.

Target en la mayoría de ocasiones es una referencia al objeto en el cual se lanzó el evento.

```
const formulario = document.getElementById('formulario');  
  
formulario.addEventListener('submit', (e) => {  
  e.preventDefault();  
  const formulario = e.target;  
});
```

Obtener valores de un formulario



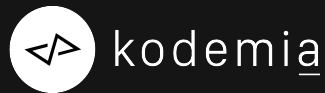
El target en términos simples es un equivalente a una búsqueda del elemento html ya sea por medio de `QuerySelector` `getElementById`, `getElementsByClassName`, etc.

```
const formulario = e.target;
```

=

```
const formulario = document.getElementById('formulario');
```

Obtener valores de un formulario



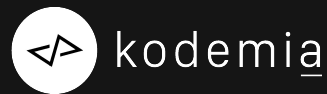
El siguiente paso es obtener todos los inputs dentro del formulario, podemos utilizar `querySelectorAll`, sin embargo este método nos regresa un **nodelist** por lo cual debemos convertirlo a un Arreglo o Array para poder tratarlos los datos con un `map`, `filter`, `foreach`, etc.

```
const formulario = document.getElementById('formulario');

formulario.addEventListener('submit', (e) => {
  e.preventDefault();
  const formulario = e.target;
  const inputs = formulario.querySelectorAll('input');
  const inputsList = Array.from(inputs);

});
```

Obtener valores de un formulario



Después de la conversión a un arreglo, mediante el método `reduce` es posible recorrer cada elemento del arreglo y encapsular todos los valores en un **objeto literal** colocando como **Key el nombre del input** y su valor como **el value del input**

```
const formulario = document.getElementById('formulario');

formulario.addEventListener('submit', (e) => {
  e.preventDefault();
  const formulario = e.target;
  const inputs = formulario.querySelectorAll('input');
  const inputsList = Array.from(inputs);
  const values = inputsList.reduce((acc, input) => {
    acc[input.name] = input.value;
    return acc;
  }, {});
  console.log(values);
});
```

Obtener valores de un formulario



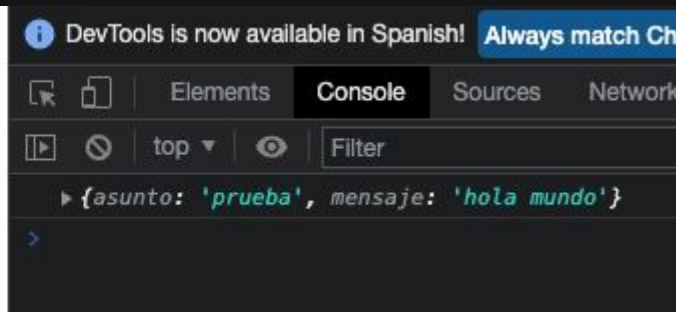
kodemia

```
const formulario = document.getElementById('formulario');

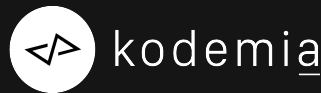
formulario.addEventListener('submit', (e) => {
  e.preventDefault();
  const formulario = e.target;
  const inputs = formulario.querySelectorAll('input');
  const inputsList = Array.from(inputs);
  const values = inputsList.reduce((acc, input) => {
    acc[input.name] = input.value;
    return acc;
  }, {});
  console.log(values);
});
```


Obtener valores de un formulario

Resultado a obtener



Obtener valores de un formulario

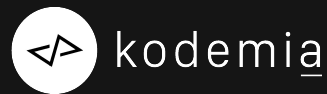


Sin bien todo en programación no tiene una solución definitiva, es posible optimizar el proceso para obtener los valores utilizando prototipos y métodos de Javascript

```
const formulario = document.getElementById('formulario');

formulario.addEventListener('submit', (e) => {
  e.preventDefault();
  const data = Object.fromEntries(new FormData(e.target));
  console.log(data);
});
```

Obtener valores de un formulario



```
const formulario = document.getElementById('formulario');

formulario.addEventListener('submit', (e) => {
  e.preventDefault();
  const data = Object.fromEntries(new FormData(e.target));
  console.log(data);
});
```

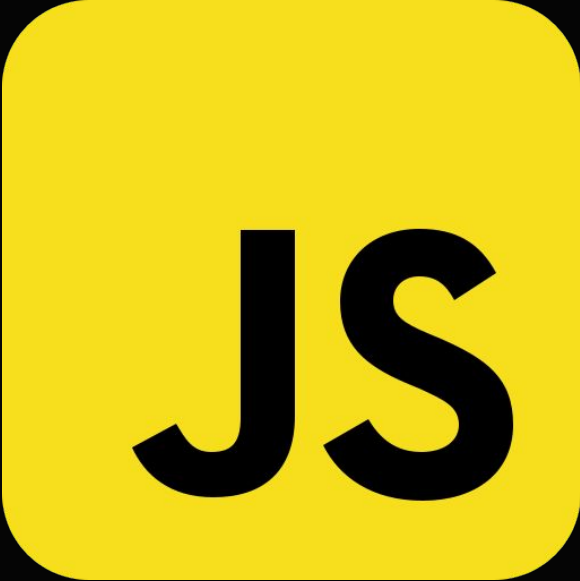
FormData: Prototipo que proporciona una manera sencilla de construir un conjunto de parejas clave/valor que representan los campos de un formulario y sus valores.

Object.fromEntries: Método transforma una lista de pares clave-valor en un objeto.

Temas puntuales de la sección



- ¿Qué es un formulario?
- Etiqueta Form
- Estructura de un formulario
- Obtener el valor de un input
- Obtener Valores de todo el formulario a través del método submit



JS



16

Formularios

Yael Ruiz Pucheta