

# Formularios con JS

The right way

# ¿Para que se usan los formularios?

Los **formularios HTML** están preparados para obtener información del usuario y enviarla (en la mayoría de los casos) a un servidor que la procese.

## I am a form



Username

Password

Email

Submit Reset

Cuando apretamos un botón o un input del tipo 'submit', el navegador automáticamente intenta **enviar el formulario**.

**Por esto la pagina se recarga!**

**Como hago si quisiera procesar el  
formulario con JS antes de  
enviarlo?**

**O incluso no quisiera enviarlo al  
servidor?**

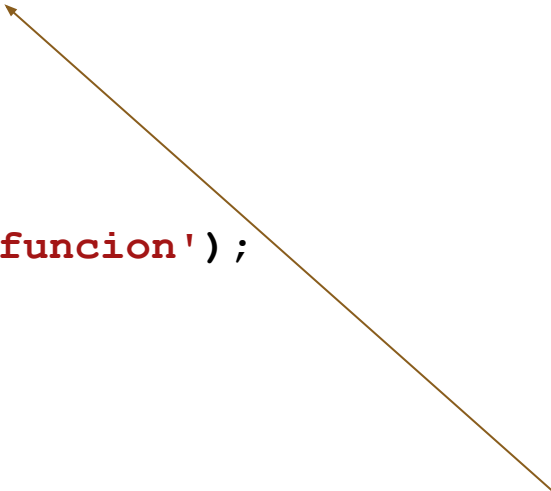
# Formulario con JS

```
"use strict"

// escuchamos el SUBMIT del form

let form = document.querySelector('#form');
form.addEventListener('submit', agregar);

function agregar(e) {
    e.preventDefault();
    console.log('entré a la funcion');
}
```



En lugar de escuchar el evento “click” del botón, escuchamos el evento “submit” del formulario.

Con esto logramos aprovechar las validaciones HTML de los formularios.

# FormData

## Usando objetos **FormData**

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

```
<form id="form">  
  <input name="email">  
  <input name="nombre">  
</form>
```

```
let form = document.querySelector('#form');  
let formData = new FormData(form);
```

```
let nombre = formData.get('nombre');  
let email = formData.get('email');
```

# Validaciones HTML

---

Html tiene sus mecanismos para validar de **formulario en el lado del cliente** y ayuda a garantizar que los datos que se envían coinciden con los requisitos establecidos en los diversos controles de formulario.

minlength

maxlength

min

max

step

required

disabled

readonly

[https://developer.mozilla.org/es/docs/Learn/Forms/Form\\_validation](https://developer.mozilla.org/es/docs/Learn/Forms/Form_validation)

# AHORA LES TOCA PRACTICAR :D

