

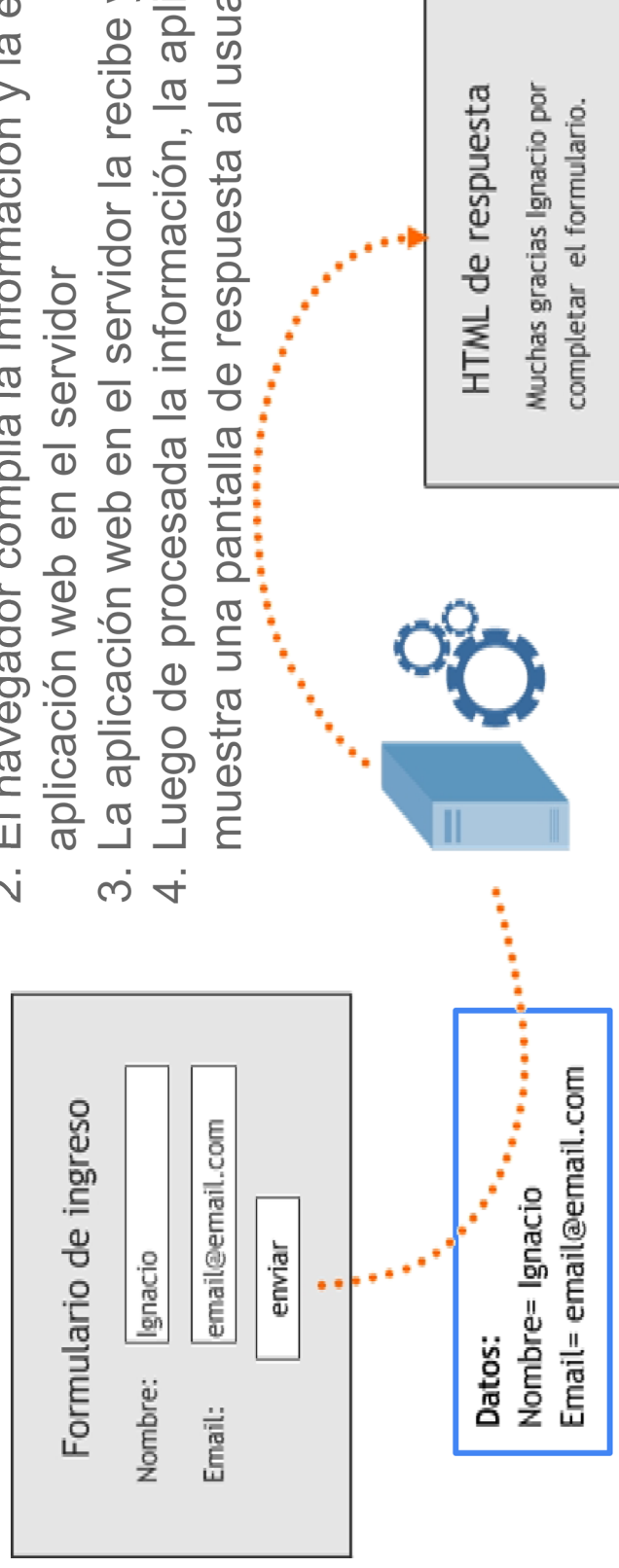
Formularios

Formularios

¿Cómo trabaja un formulario?

Dos partes para que trabaje un formulario. La que ve el usuario con los campos y botones, y la que procesa la información que ingresa el usuario

1. El usuario completa el formulario y presiona enviar.
2. El navegador compila la información y la envía a una aplicación web en el servidor
3. La aplicación web en el servidor la recibe y procesa.
4. Luego de procesada la información, la aplicación le muestra una pantalla de respuesta al usuario.



Formularios

Declaración de un Formulario | `<form>`

Más que una etiqueta de marcado es una declaración de que una acción se ejecutará, y que permitirá enviar los datos seleccionados e ingresados a través de los campos del formulario.

Ejemplo:

```
<form method="POST" action="respuesta.php">
```

Formularios

Atributos de la etiqueta **<form>**

Al declarar un formulario la etiqueta de apertura incluye dos atributos sin los cuales el formulario no funciona:

- **ACTION**
Señala la ruta donde se procesan los datos del formulario.
- **METHOD**
Especifica cómo se va a enviar la información.
Puede ser GET o POST y determina la forma de envío de los datos.

Formularios

Fieldset y Legend

Fieldset es usado para indicar un grupo lógico de los campos.

Toda etiqueta **Fieldset** incluye una etiqueta **Legend** que le da un nombre a los campos agrupados.

Label

Relaciona un texto a un elemento `<input>`.

```
<label for="Inombre">Nombre:</label>
```

```
<input type="text" id="Inombre" name="fnombre" />
```

Formularios

Campos | `<input type="text">`

La etiqueta `<input>` indica un elemento simple de formulario.

Lleva dos atributos:

1. type (tipo de elemento)
2. name (para asignarle un nombre)

Ejemplo:

```
<input type="text" name="nombre">
```

Formularios

Áreas de Texto | `<textarea>`

Las Áreas de Texto son campos de entrada donde el usuario podrá escribir comentarios, consultas, etc.

Se diferencia de los campos de texto por contener muchas líneas.

Ejemplo:

```
<textarea name="comentarios" id="comentarios" cols="45"
rows="5"></textarea>
```

Formularios

Casillas de Verificación | `<input type="checkbox">`

Permiten elegir entre varios elementos de una lista. Seleccionando uno o varios de los elementos de una lista.

El atributo name siempre es el mismo y lo que varía es el atributo value.

Música Cine Leer

Botones de Opción | `<input type="radio">`

Indican una lista de elementos de los cuales sólo puede seleccionarse uno.

Masculino Femenino

Formularios

Listas/Menú | `<select>`

Los combos o selecciones permiten al usuario elegir uno o varios elementos de un menú o lista.

Ejemplo:

```
<select name="pais" id="pais">  
  <option selected="selected">seleccione país</option>  
  <option value="argentina">argentina</option>  
  <option value="brasil">brasil</option>  
  <option value="colombia">colombia</option>  
</select>
```

Formularios

Botones | `<input type="submit">` | `<input type="reset">`

El botón de envío (submit) le indica al navegador que envíe al servidor los datos del formulario.

Ejemplo:

```
<input type="submit" name="button" id="button" value="Enviar" />
```

Validación de formularios

Agregar un atributo al campo para obligar al usuario a rellenar determinados datos. **Required**

Dreamweaver maneja validar formularios con el componente llamado **Spry**

Personalizar campos de Formulario

Selectores de Atributos:

`input[type=text] { }` es un selector de atributos, y sólo seleccionará esos elementos, con el atributo correspondiente. Según las especificaciones de CSS, los navegadores pueden o no usar información sobre los atributos predeterminados.

Selector **:nth-child()**

El selector **`nth-child (n)`** coincide con cada elemento que es el enésimo hijo, independientemente del tipo, de su elemento principal.

También se puede hacer para: el primero, último, pares o impares.

- **`first-child`**, el primero
- **`last-child`**, el último
- **`nth-child(even)`** , los pares
- **`nth-child(odd)`**, los impares

Slider

```
function carousel() {  
  var i;  
  var x = document.getElementsByClassName("mySlides");  
  for (i = 0; i < x.length; i++) {  
    x[i].style.display = "none";  
  }  
  slideIndex++;  
  if (slideIndex > x.length) {slideIndex = 1}  
  x[slideIndex-1].style.display = "block";  
  setTimeout(carousel, 2000);  
  // Cambia la imagen cada 2 segundos  
}
```

CSS MediaQueries

- **Desktop 1100px**
- **Tablets 768px**
`@media (max-width: 768px) { }`
- **Smartphone 480px**
`@media (max-width: 480px) { }`

Posicionamiento

Las propiedades de posicionamiento permiten colocar un elemento sin importar el flujo del HTML.

- **Static:** Predeterminado

Está siempre posicionado de acuerdo con el flujo normal de la página.

- **Fixed:** Se coloca en relación con la ventana del navegador.
- **Relative:** Situado con relación a su posición normal
- **Absolute:** Posicionado en relación con el primer elemento

padre que tiene una posición que no sea estática. Si no se encuentra dicho elemento, el bloque de contención es

<html>

Propiedades CSS3

- **Mantener Padding(relleno) en el ancho de Caja:**
`box-sizing: border-box;`
- **Bordes redondeados:**
`border-radius: 6px;`
- **Sombra de Cajas:**
`box-shadow: 2px 2px 2px #333;`

Navegación en una misma pantalla

Se definen anclas con ID, y se vinculan con un # adelante

```
id="parte1"
```

```
<a href ="parte1 " >ver sección</a>
```

Transición animada

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></scrip
t>
<script>
$(document).ready(function(){
  $("a").on('click', function(event) {
    if (this.hash !== "") {
      event.preventDefault();
      var hash = this.hash;
      $('html, body').animate({
        scrollTop: $(hash).offset().top
      }, 800, function(){
        window.location.hash = hash;
      });
    } // End if
  });
});</script>
```

Parallax

- **Parallax es una tendencia de sitio web donde el contenido de fondo se mueve a una velocidad diferente que el contenido de primer plano mientras se desplaza**
- **Lo principal es congelar la imagen de fondo:**
background-attachment: fixed;

Reemplazar menu responsive

```
<script>
  var navigation = responsiveNav("#nav", {
    animate: true,      // Boolean: Use CSS3 transitions, true or false
    transition: 400,    // Integer: Speed of the transition, in milliseconds
    label: "Menu",     // String: Label for the navigation toggle
    insert: "after",    // String: Insert the toggle before or after the navigation
    customToggle: "",  // Selector: Specify the ID of a custom toggle
    openPos: "relative", // String: Position of the opened nav, relative or
    static
    jsClass: "js",     // String: 'JS enabled' class which is added to <html>
    init: function(), // Function: Init callback
    open: function(), // Function: Open callback
    close: function() // Function: Close callback
  });
</script>
```

iFrame

inline frame o marco incorporado: es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

Por ejemplo videos de YouTube

```
<iframe width="560" height="315"  
src="https://www.youtube.com/embed/CReHroIfwYE"  
frameborder="0" allowfullscreen></iframe>
```

Posicionamiento

Las propiedades de posicionamiento permiten colocar un elemento sin importar el flujo del HTML.

- **Static:** Predeterminado

Está siempre posicionado de acuerdo con el flujo normal de la página.

- **Fixed:** Se coloca en relación con la ventana del navegador.
- **Relative:** Situado con relación a su posición normal
- **Absolute:** Posicionado en relación con el primer elemento padre que tiene una posición que no sea estática. Si no se encuentra dicho elemento, el bloque de contención es `<html>`

Slider

```
function carousel() {  
  var i;  
  var x = document.getElementsByClassName("mySlides");  
  for (i = 0; i < x.length; i++) {  
    x[i].style.display = "none";  
  }  
  slideIndex++;  
  if (slideIndex > x.length) {slideIndex = 1}  
  x[slideIndex-1].style.display = "block";  
  setTimeout(carousel, 2000);  
  // Cambia la imagen cada 2 segundos  
}
```