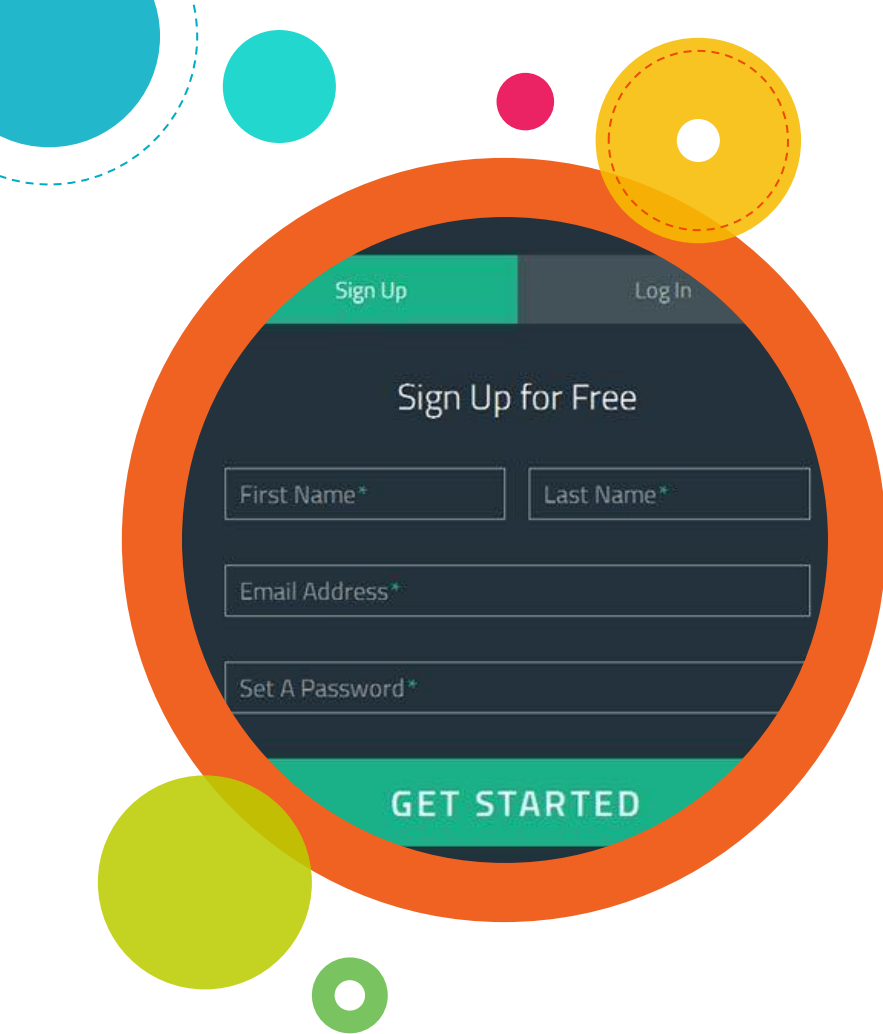


# HTML + CSS

# ¿Qué vimos la clase anterior?



Rutas, enlaces, imágenes, videos,  
fondos con css y selectores nuevos

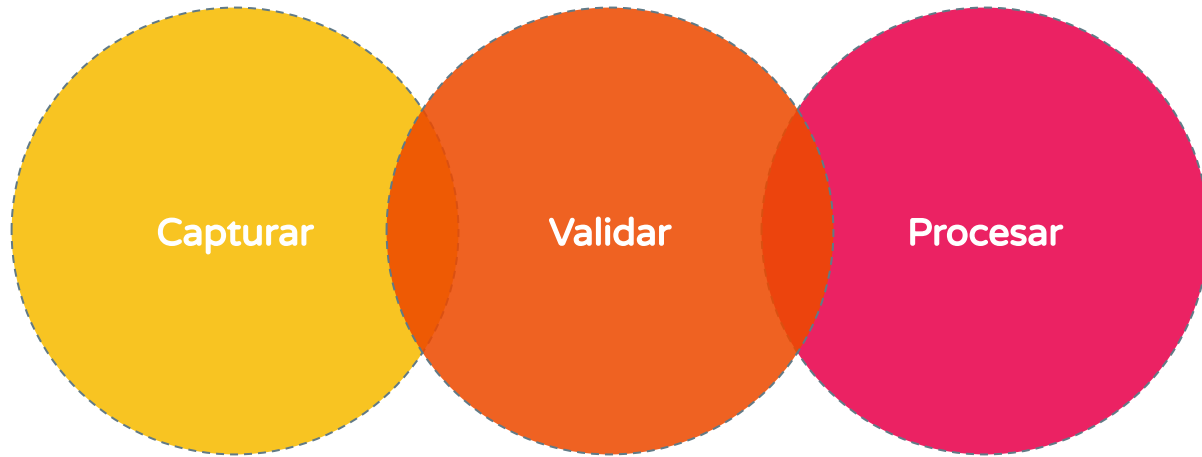


# Formularios HTML

Sistema para **capturar datos**.

Necesita de lenguajes adicionales para su completo funcionamiento.

# Funcionamiento de un Formulario en HTML





Definamos!

Un formulario está compuesto por “**3 partes**” para su completo funcionamiento. Éstas son:

**Capturar**

HTML  
(Cliente)

**Validar**

Javascript  
(Cliente)

**Procesar**

PHP / CGI  
(Servidor)

# Tags para formularios

## == Contenedor ==

```
<form action="script.php" method="post">
```

...

```
</form>
```

**Definición:** tag **MUY** importante, sin éste el formulario no funciona.  
**TODOS** los controles de un formulario van anidados dentro.

# Tags para formularios

## == Contenedor ==

```
<fieldset>  
  <legend> Datos personales </legend>  
  ...  
</fieldset>
```

### Definición:

**<fieldset>** : agrupa campos relacionados.

**<legend>** : asigna un "título" a este grupo de campos .

# Tags para formularios

## == Rótulos ==

`<label>Nombre y apellido:</label>`

**Definición:** tag que **acompaña** a un campo, el texto se muestra en el browser e indica la información que es requerida por el mismo. Cada control deberá llevar un **rótulo**.



# Tags para formularios

## == Campos unilínea ==

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

```
<input type="email" name="correo">
```

```
<input type="tel" name="telefono">
```

```
<input type="number" name="cantidad">
```

**Definición:** tag **MULTIFUNCIÓN**, cambiando el valor de su atributo **type** podemos obtener distintos tipos de campos. El atributo **name** lo identifica y diferencia de los demás campos además de servir para procesar la información del mismo.

# Tags para formularios

## == Campos unilínea ==

```
<input type="url" name="sitioWeb">
```

```
<input type="password" name="contraseña">
```

```
<input type="file" name="tu_foto">
```

```
<input type="submit" value="ENVIAR">
```

**Definición:** tag **MULTIFUNCIÓN**, cambiando el valor de su atributo **type** podemos obtener distintos tipos de campos. El atributo **name** lo identifica y diferencia de los demás campos además de servir para procesar la información del mismo.

# Tags para formularios

## == Opciones ==

```
<input type="radio" name="estado_civil">
```

```
<input type="radio" name="estado_civil">
```

```
<input type="checkbox" name="hobbies">
```

```
<input type="checkbox" name="hobbies">
```

**Definición:** **radio-button** genera un botón de única opción, **checkbox** genera una casilla de verificación (permite elegir varios).

# Tags para formularios

## == Campo multilinea ==

```
<textarea name="mensaje"> </textarea>
```

**Definición:** campo para escribir varias líneas de texto. Usado generalmente para la opción “tu mensaje” en un formulario.

# Tags para formularios

## == Desplegable ==

```
<select name="pais">  
  <option value="Ar">Argentina</option>  
  <option value="Br">Brasil</option>  
  <option value="Co">Colombia</option>  
</select>
```

**Definición:** combobox ó dropdown de única selección. Agrupa muchas opciones en un solo control.

# Tags para formularios

## == Desplegable ==

```
<select name="pais">  
  <optgroup label="Sur América">  
    <option value="Ar">Argentina</option>  
    <option value="Br">Brasil</option>  
    <option value="Co">Colombia</option>  
  </optgroup>  
</select>
```

**Definición:** usado para agrupar opciones relacionadas dentro de un dropdown.

# Tags para formularios

## == Botones ==

```
<button type="submit">Enviar</button>
```

```
<button type="reset">Borrar</button>
```

```
<button type="button">Otra acción</button>
```

**Definición:** botones de acción para enviar, borrar o generar otro tipo de acción, **permite anidamiento** de otros tags.

# Atributo == required ==

```
<input type="email" required>
```

**Definición:** hace que un **campo sea obligatorio** y arroja una alerta si no se ha ingresado datos en el mismo. Dependiendo del tipo de campo la **alerta puede variar**.

Cuando validemos con JS o PHP, este atributo no será necesario.



# Atributo == placeholder ==

```
<input type="email" placeholder="user@email.com">
```

**Definición:** permite insertar un texto de ayuda para el usuario que describe el valor esperado en este campo.



Practiquemos

Realizar el punto **#1** y **#2** de la guía de ejercicios.

# Más de CSS



Nuevas propiedades y pseudo selectores

# Propiedades de CSS

## == Redimensión ==

```
resize: none;
```

Otros valores:

🎯 [both | horizontal | vertical]



**Definición:** permite definir si queremos redimensionar un elemento.  
Su **uso más común** suele darse para el tag `<textarea>`.

# Pseudo Selectores CSS

## == Sintaxis ==

```
selector:pseudo {  
    propiedad: valor;  
}
```



**Definición:** un pseudo selector nos **permite controlar eventos especiales** de un elemento. Suelen ser aplicados sobre un selector existente.

# Pseudo selectores CSS

## == :hover ==

```
a {  
  text-decoration: none;  
}
```

```
a: hover {  
  text-decoration: underline;  
}
```

**Definición:** controla el estado “**rollover**” de cualquier elemento. Lo que definamos con este selector, solo será visible al **posar el cursor sobre** el elemento.

# Pseudo selectores CSS

## == :focus ==

```
input {  
  background-color: #ff0000;  
}
```

```
input:focus {  
  background-color: #000000;  
}
```

**Definición:** controla el estado “**focal**” de los campos de un formulario. Lo que definamos con este selector, solo será visible al **ubicar el cursor dentro** del campo.



Practiquemos

Realizar el punto **#3** de la guía de ejercicios.



Gracias!



¿Consultas?