



Otros campos de formulario

HTML dispone de otros elementos para insertar campos de formulario. Estos son:

Optgroup

Representa un grupo de opciones, organizadas temáticamente, que forman parte de un elemento `<select>`. Por ejemplo:

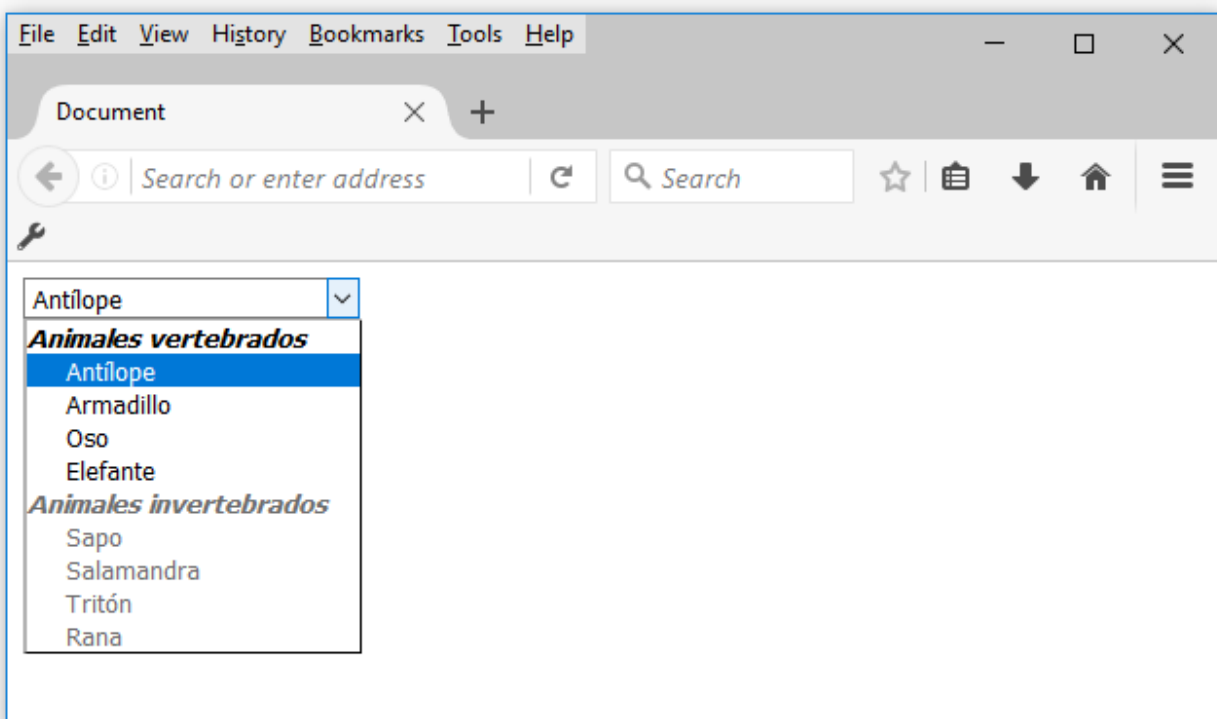
```
<select>
  <optgroup label="Animales vertebrados">
    <option value="antilope">Antílope</option>
    <option value="armadillo">Armadillo</option>
    <option value="oso">Oso</option>
    <option value="elefante">Elefante</option>
  </optgroup>
  <optgroup label="Animales invertebrados">
    <option value="sapo">Sapo</option>
    <option value="salamandra">Salamandra</option>
    <option value="tritón">Tritón</option>
    <option value="rana ">Rana </option>
  </optgroup>
</select>
```



HTML & CSS

desde cero hasta avanzado

Y este sería el resultado en la pantalla del navegador:



Este elemento admite dos atributos:

- **label**, que inserta una línea de texto que identifica al grupo de opciones.
- **disabled**, indica que el grupo y sus opciones se encuentran deshabilitados.



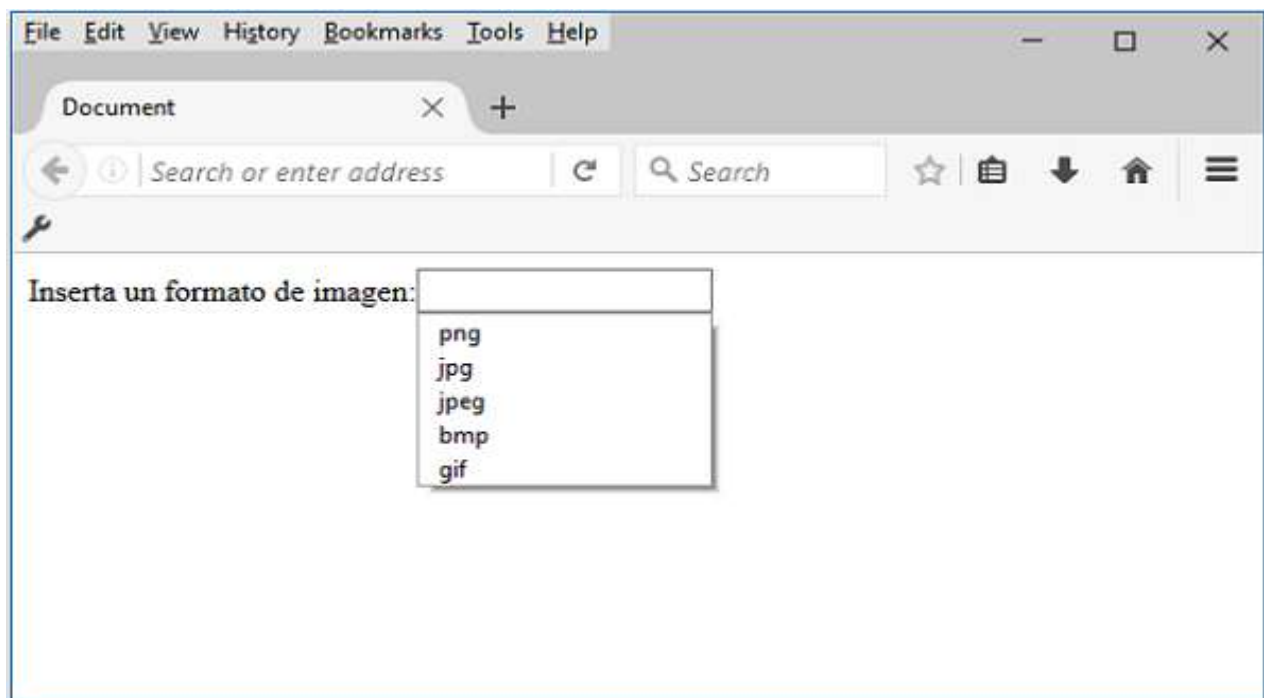
HTML & CSS

desde cero hasta avanzado

Datalist

Inserta una lista de opciones con el fin de proveer sugerencias a los usuarios. Estas sugerencias son definidas mediante el elemento `<option>`.

```
Inserta un formato de imagen:<input list="formatos">
<datalist id="formatos">
  <option value="png">
  <option value="jpg">
  <option value="jpeg">
  <option value="bmp">
  <option value="gif">
</datalist>
```





Este elemento no tiene otros atributos mas que aquellos que son comunes a todos los elementos.

Output

Se utiliza para mostrar los resultados de un cálculo llevado a cabo en el documento HTML. Este cálculo se realiza mediante un lenguaje de programación del lado cliente como JavaScript.

```
<form oninput="resultado.value=parseInt(a.value)+parseInt(b.value)">
  <input type="range" name="rango" value="50" /> +
  <input type="number" name="numero" value="10" /> =
  <output name="resultado">60</output>
</form>
```

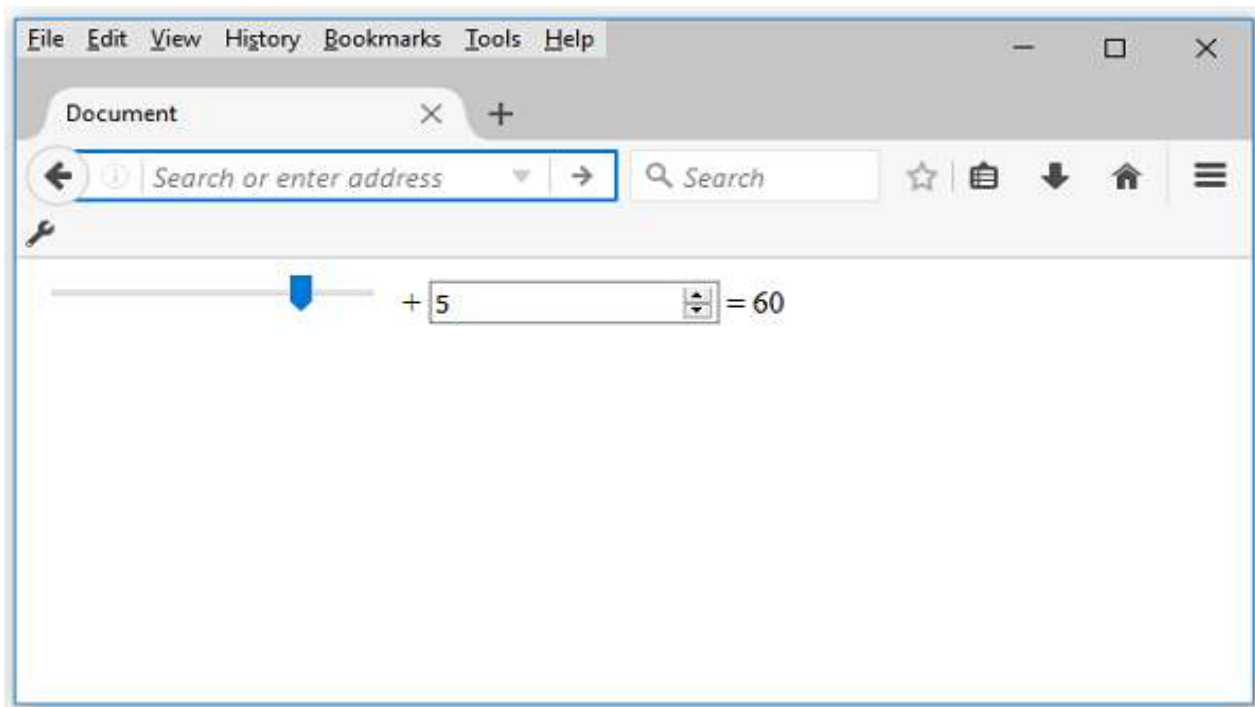
En el código de arriba, se muestra un formulario que cuenta con un campo de rango numérico, conjuntamente con un campo de entrada de datos dentro del cual podemos ingresar valores numéricos de forma manual. Cada vez que cambian los valores de cada campo el resultado se muestra en el elemento `<output>`.

Y así es como se muestra en el navegador:



HTML & CSS

desde cero hasta avanzado



Los atributos que admite este elemento son:

- **For:** define una lista de identificadores que van separados por espacios, cada uno de los cuales debería coincidir con el atributo *id* de uno de los campos utilizados en el cálculo.
- **Form:** define el formulario al cual está asociado un elemento.
- **Name:** el nombre que será usado en el envío del formulario.



Fieldset y Legend

El elemento `<fieldset>` agrupa un conjunto de campos de formulario bajo un mismo nombre. Por defecto, el navegador muestra un marco alrededor de los campos agrupados.

Por su parte, el elemento `<legend>` crea un título para los campos agrupados.

El elemento `<fieldset>` admite los atributos *name*, *disabled* y *form*.

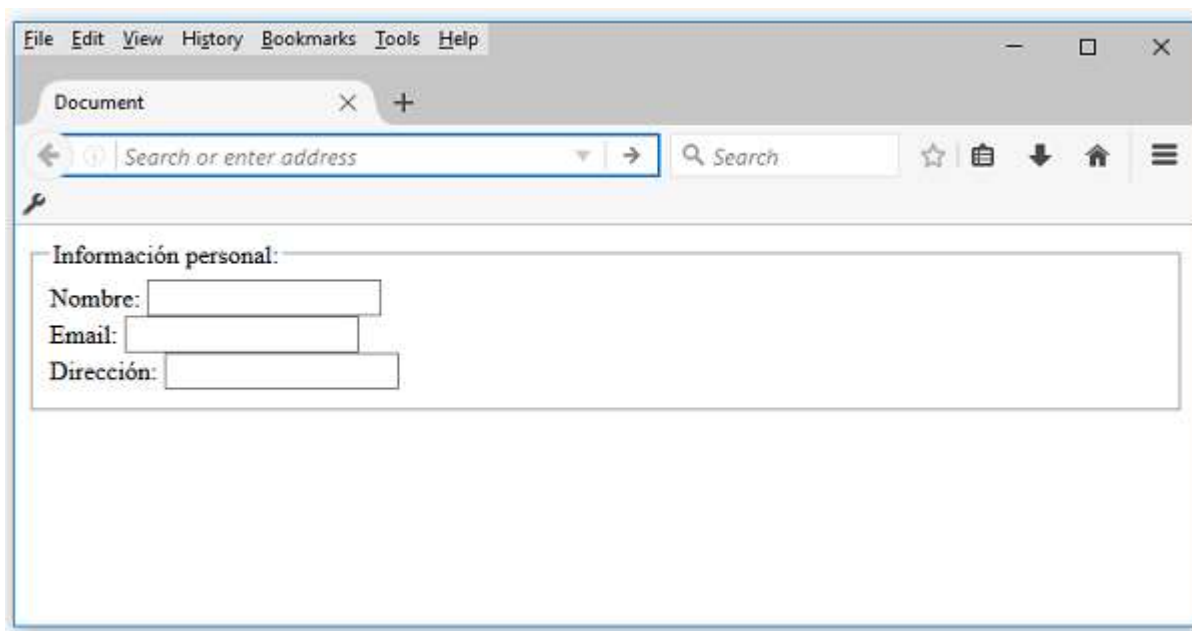
El elemento `<legend>` no tiene atributos específicos.

```
<form>
  <fieldset>
    <legend>Información personal:</legend>
    Nombre: <input type="text"><br>
    Email: <input type="email"><br>
    Dirección: <input type="text">
  </fieldset>
</form>
```



HTML & CSS

desde cero hasta avanzado



Meter

Se utiliza para representar una medida dentro de un rango conocido.

Este elemento es muy útil en situaciones en las cuáles debemos mostrar información que se puede representar mediante un medidor, por ejemplo, para indicar que una contraseña tiene un nivel de seguridad bajo, medio o alto, o también para mostrar el espacio disponible en un disco duro, etc.

Los atributos que soporta este elemento son:

Value: representa el valor actual del medidor.



HTML & CSS

desde cero hasta avanzado

Min y Max: representan los límites inferior y superior del rango.

```
Valor actual y valor mínimo y máximo: <br><br>
<meter min="2" max="20" value="4">valor 4</meter> <br>
<meter min="2" max="20" value="10">valor 10</meter> <br>
<meter min="2" max="20" value="19">valor 19</meter> <br><br>
```

Low y High: definen valores considerados como bajos y altos, respectivamente.

```
Valor optimum <br><br>
<meter min="10" max="30" value="15" optimum="20">valor 15</meter> <br>
<meter min="10" max="30" value="25" optimum="20">valor 25</meter> <br>
<meter min="10" max="30" value="20" optimum="20">valor 20</meter>
<br><br>
```

Optimum: indica cuál es el valor óptimo del medidor.

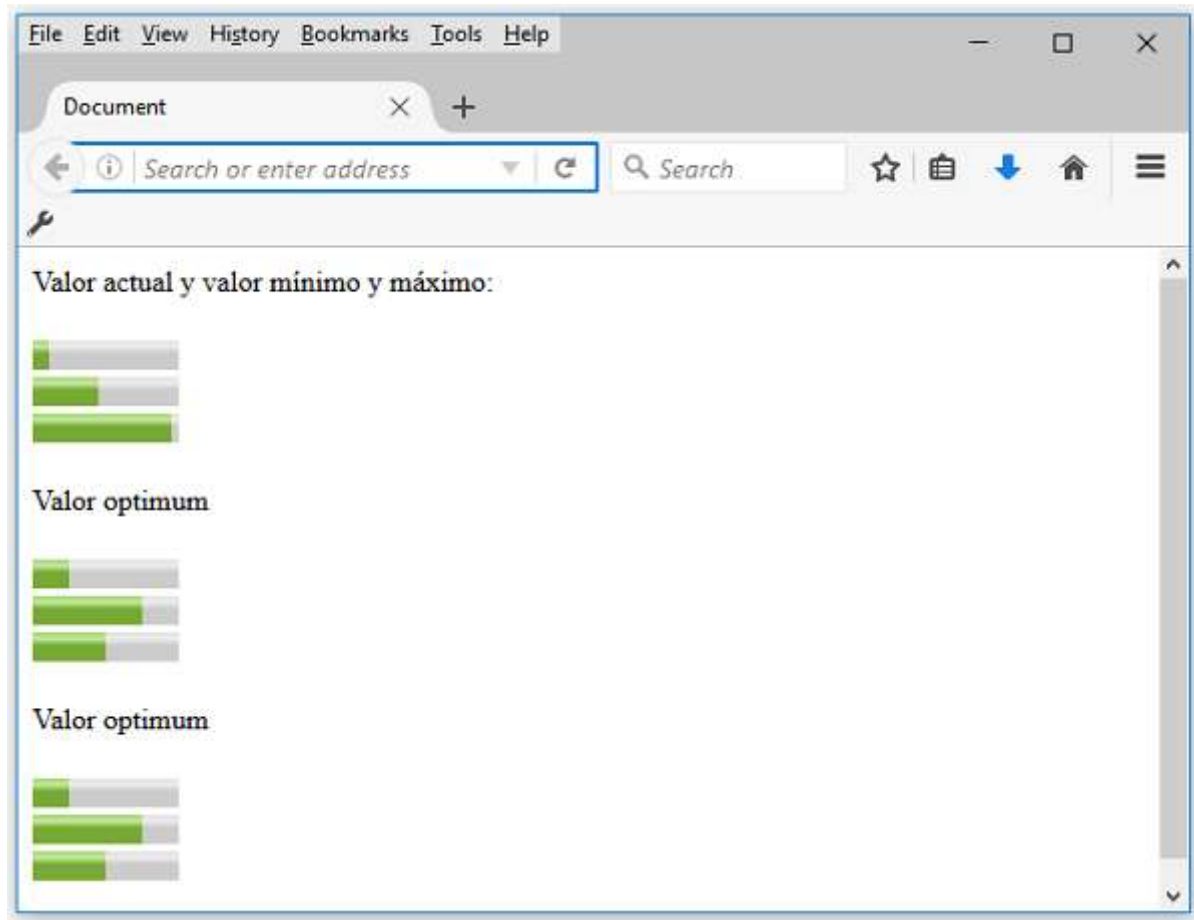
```
Valor optimum <br><br>
<meter min="10" max="30" value="15" optimum="20">valor 15</meter> <br>
<meter min="10" max="30" value="25" optimum="20">valor 25</meter> <br>
<meter min="10" max="30" value="20" optimum="20">valor 20</meter>
<br><br>
```

Y así es como se muestran los bloques de códigos anteriores en el navegador:



HTML & CSS

desde cero hasta avanzado



Progress

Se utiliza para indicar de manera gráfica el progreso de una tarea o de un proceso. Este campo de formulario se representa por una barra de progreso que se mueve de izquierda a derecha, de acuerdo con el progreso de la operación.

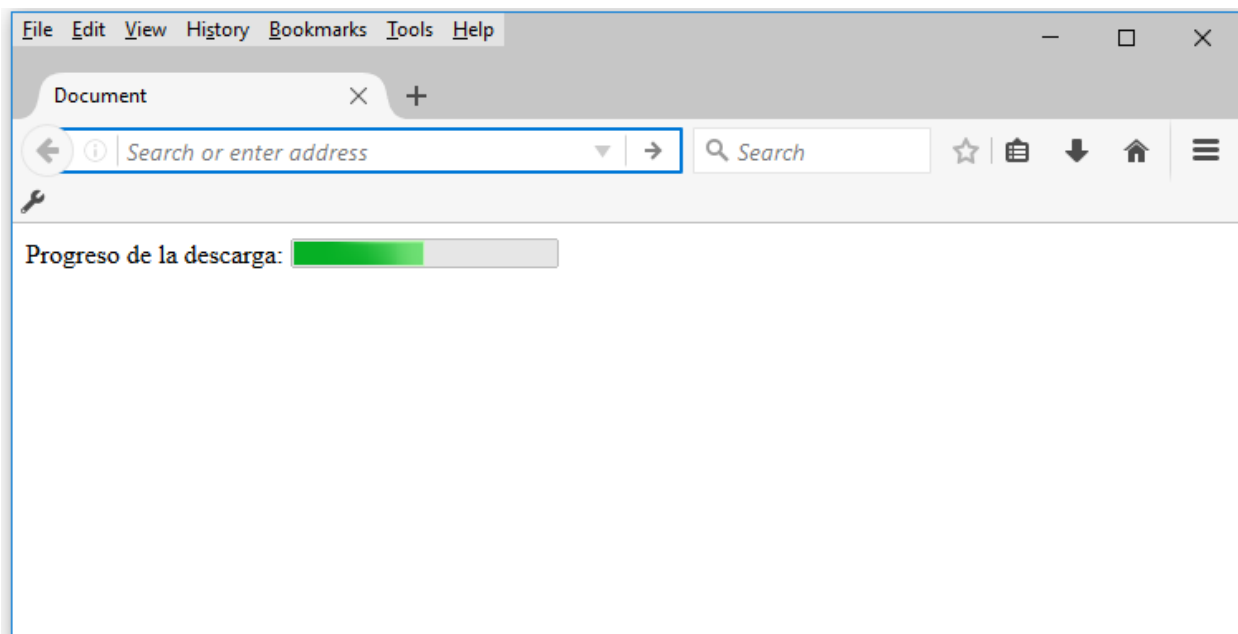


HTML & CSS

desde cero hasta avanzado

Mediante el atributo *max* podemos definir el valor que debe tener la barra de progreso para estar completa, mientras que el atributo *value* permite definir el valor actual de la barra de progreso.

```
Progreso de la descarga: <progress value="50" max="100"></progress>
```



Por defecto, las barras de progreso se muestran sin interactividad, pero podemos hacer que los valores aumente de manera dinámica mediante JavaScript.