

AUT2_07_Formularios y varios

Rubén González Rodríguez y Rubén Ruiz Pérez 2ºDAW

Índice

Contenidos teóricos.....	3
Etiqueta <input>.....	3
Atributos de <input>.....	3
Presentación.....	5
Ejercicio práctico.....	7
Solución propuesta del ejercicio.....	8

Contenidos teóricos

Etiqueta <input>

La etiqueta <input> es un elemento HTML que permite recibir datos de diversos tipos del usuario. Se sitúa normalmente dentro de un formulario y es una etiqueta de autocierre, es decir, no necesita una etiqueta </input> después.

Atributos de <input>

Esta etiqueta cuenta con los siguientes atributos:

- **type:** es el atributo más importante, ya que define el tipo de entrada a mostrar. Su valor por defecto es *text*, y presenta los siguientes posibles valores:
 - button: se muestra como un botón sin funcionalidad específica.
 - checkbox: casillas de selección múltiple, se usa junto al atributo **value** para determinar el valor que se envía.
 - color: permite especificar un color. Envía el código del color seleccionado en formato hexadecimal.
 - date: permite introducir una fecha (año, mes y día).
 - datetime-local: similar a *date*, pero permite además introducir hora, minutos y segundos.
 - email: usado para introducir direcciones de correo electrónico. El valor de esta entrada se valida antes de enviarse el formulario y, en caso de tener un formato incorrecto, se manda una cadena vacía.
 - file: permite al usuario seleccionar un archivo. Se puede especificar los formatos de archivos válidos con el atributo **accept** y si se puede mandar más de un archivo con el atributo **multiple**.
 - hidden: el input no se muestra por pantalla pero su valor es enviado al servidor.
 - image: funciona como botón de envío de formulario, pero con una imagen.
 - number: permite introducir números de punto flotante. Se puede indicar un valor mínimo, uno máximo y el incremento que se aplica cada vez que se aumenta o decrece el valor usando los controles.
 - password: permite introducir texto ocultando su valor.
 - radio: entrada que permite seleccionar un único valor de entre todas las entradas de su tipo que tengan el mismo valor del atributo **name**.
 - reset: botón que devuelve el contenido del formulario a sus valores por defecto.
 - submit: botón que envía el formulario.
 - text: permite enviar una cadena de texto de una sola línea.
 - time: permite introducir una hora con minutos y segundos.
- **accept:** se emplea junto al tipo “file”. Permite introducir los tipos de archivos aceptados por el servidor. Se puede introducir las extensiones separadas por comas “.jpg,.png,.pdf” o los siguientes tipos:
 - “audio/*” para archivos de audio.
 - “image/*” para archivos de imagen.
 - “video/*” para archivos de vídeo.
- **autofocus:** atributo booleano que indica que el control tiene el cursor cuando la página cargue. Solo un control del formulario puede tener este atributo.
- **checked:** se emplea junto al tipo “radio” o “checkbox”. Es un atributo booleano que determina si el control está seleccionado por defecto.

- **disabled:** atributo booleano que determina que no se puede interactuar con el control, por lo que los eventos “click” no se invocarán ni su valor será enviado con el formulario.
- **form:** referencia al formulario del que forma parte un control, por lo que permite poner `<input>` en cualquier lugar del documento. Su valor debe coincidir con el atributo **id** del formulario y, si no se especifica, el input debe ser descendiente del formulario.
- **max:** permite establecer un valor máximo para campos numéricos o de fecha-hora. Su valor debe ser mayor que el atributo **min**.
- **maxlength:** permite establecer la longitud máxima de caracteres permitidos en entradas de texto (*text, email, password...*).
- **min:** permite establecer un valor mínimo para campos numéricos o de fecha-hora. Su valor debe ser menor que el atributo **max**.
- **minlength:** permite establecer la longitud mínima de caracteres permitidos en entradas de texto (*text, email, password...*).
- **multiple:** atributo booleano que indica que el usuario puede introducir más de un valor. Se emplea únicamente con los tipos “file” e “email”.
- **placeholder:** atributo que muestra una pista al usuario del valor que debe introducir.
- **readonly:** indica que el usuario no puede modificar el valor del control.
- **required:** atributo que especifica que un campo debe rellenarse antes de enviar el formulario.
- **size:** especifica el tamaño del control en píxeles.
- **value:** especifica el valor inicial del `<input>`. Es opcional, salvo para el tipo “radio” y “checkbox”.

Presentación

1. Qué es la etiqueta `<input>` y dónde se emplea. Atributo `type`, por qué es el más importante y algunos tipos y su explicación.


Formularios III

10. Formularios III: Objeto input

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

Atributos

☐ type

- **button** 
- **checkbox** ☐ Checkbox no marcada
☒ Checkbox marcada
- **color** 
- **date**
- **email**
- **file** No se ha seleccionado ningún archivo.

2. Más tipos de `<input>` y cómo se emplean.

Formularios III

- **hidden**
- **image** 
- **number**
- **password**
- **radio** ☒ Opción 1
☐ Opción 2
- **reset**
- **submit**
- **text**
- **time**

3. Otros atributos de <input> y sus explicaciones.

Otros atributos:

Formularios III

- ❑ **accept:** junto a tipo *file*, controla qué tipos de archivo acepta.
 - ❑ "audio/*", "image/*", "video/*", ".jpg,.png,.gif"
- ❑ **autofocus:** indica que el cursor debe situarse en este elemento cuando la página cargue.
- ❑ **checked:** junto a tipo *radio* o *checkbox*, determina si el control está seleccionado.
- ❑ **disabled:** no permite interactuar con el control y su valor no se envía con el formulario.
- ❑ **max/min:** junto a campos numéricos o de fecha/hora, establece un valor máximo/mínimo.
- ❑ **maxlength/minlength:** junto con entradas de texto, establece una longitud máxima/mínima de caracteres.
- ❑ **multiple:** junto a tipo *email* y *file*, permite introducir más de un valor.

4. Más atributos de <input> y cómo acceder a los atributos de los inputs en Javascript.

Formularios III

- ❑ **placeholder:** muestra una pista al usuario del valor que debe introducir.
- ❑ **readonly:** indica que el usuario no puede modificar el valor del control.
- ❑ **required:** especifica que el control debe rellenarse antes de enviar el formulario.
- ❑ **size:** indica el tamaño en píxeles del control.
- ❑ **value:** indica el valor del control. Es obligatorio para los tipos *radio* y *checkbox*.

Cómo acceder a los atributos del input

Por ejemplo, acceder a los valores de los inputs de un formulario:

```
let valor = nodoInput.value; //tomamos el valor directamente de campos de texto, numericos...
let valorRadio = arrayNodos //para los input de tipo radio o checkbox, creamos un array a
.filter(input => input.checked)[0].value; //partir del nombre y nos quedamos con el que esta seleccionado
```

Para modificar o añadir un atributo:

```
nodoInput.size = 30; //asignamos directamente los atributos a los nodos
arrayNodos.forEach(input => input.required = true);
```

5. Enunciado del ejercicio práctico.

Ejercicio: Tomando el siguiente HTML, crear el método `mostrarResultado()`, que toma los valores del formulario y muestra en `#resultado` el nombre con el color y fuente elegidos. ¡Como extra puedes añadir más input y personalizar más el resultado!

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
  </style>
</head>
<body>
  <div>
    <form action="">
      <label for="nombre">Nombre: </label>
      <input type="text" name="nombre" id="nptNombre">
      <br>
      <label for="color">Color de nombre: </label>
      <input type="color" name="color" id="nptColor">
      <br>
      <label for="fuente">Fuente de letra: </label>
      <input type="radio" name="fuente" value="Arial" checked="">Arial
      <input type="radio" name="fuente" value="Verdana">Verdana
      <br>
      <input type="button" onclick="mostrarResultado()" value="Ejecutar">
    </form>
  </div>
  <div>
    <p>Resultado: <span id="resultado"></span></p>
  </div>
</body>
</html>
```

Formularios III

Ejercicio práctico

Tomando el siguiente HTML, crear el método `mostrarResultado()`, que toma los valores del formulario y muestra en `#resultado` el nombre con el color y fuente elegidos. ¡Como extra puedes añadir más input y personalizar más el resultado!

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
  </style>
</head>
<body>
  <div>
    <form action="">
      <label for="nombre">Nombre: </label>
      <input type="text" name="nombre" id="nptNombre">
      <br>
      <label for="color">Color de nombre: </label>
      <input type="color" name="color" id="nptColor">
      <br>
      <label for="fuente">Fuente de letra: </label>
      <input type="radio" name="fuente" value="Arial" checked>Arial
      <input type="radio" name="fuente" value="Verdana">Verdana
      <br>
      <input type="button" onclick="mostrarResultado()" value="Ejecutar">
    </form>
  </div>
  <div>
    <p>Resultado: <span id="resultado"></span></p>
  </div>
  <script>
    /* mostrarResultado() -> toma los valores del formulario y muestra en el span#resultado el
    nombre introducido con el color y fuente elegidos */
  </script>
</body>
</html>
```

Solución propuesta del ejercicio

```
<script>
  const DOM = {
    input: {
      nombre: document.getElementById("nptNombre"),
      color: document.getElementById("nptColor"),
      fuente: document.getElementsByName("fuente")
    },
    resultado: document.getElementById("resultado")
  }
  const mostrarResultado = () => {
    let nombre = DOM.input.nombre.value;
    let color = DOM.input.color.value;
    let fuente = Array.from(DOM.input.fuente).filter(input => input.checked)[0].value;
    DOM.resultado.innerHTML = nombre;
    DOM.resultado.style.color = color;
    DOM.resultado.style.fontFamily = fuente;
  }
</script>
```