

2023

UT5 – Practica 1

DESARROLLO WEB EN ENTORNO CLIENTE
DANIEL RODRIGUEZ MUÑOZ

Crea un formulario HTML con elementos input que aún no hayamos utilizado (por ejemplo: date, range, password, tel, url, etc).

A continuación, mediante código javascript, visualiza el contenido de los datos rellenos del formulario en un elemento <p> o <pre> o <div>

Utiliza las estructuras de nivel 0 o nivel 2 del DOM vistas en la teoría para referenciar los elementos.

Este es el HTML: Se compone de dos partes, un div izquierdo, que es el que contiene los diferentes inputs usados (password, email, date, range, tel y url) dentro de un form, así como el botón de submit que llamará a la función encargada de coger el contenido de cada input y mostrarlo en el div derecho, que es la segunda parte del HTML.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <script src="script.js"></script>
  <link rel="stylesheet" href="stylesheet.css"/>
</head>
<body>
<div id="container-left">
  <h3>Inputs</h3>
  <form name="form">
    <div class="subgroup">
      <label for="password">Password:</label>
      <input type="password" name="password" id="password" placeholder="password" pattern="^[0-9a-zA-Z]{5,50}$"
        title="La contraseña debe tener entre 5 y 50 caracteres. Solo números y letras." required/>
    </div>
    <br/><br/>
    <div class="subgroup">
      <label for="email">Email:</label>
      <input type="email" name="email" id="email" placeholder="example@example.com" required/>
    </div>
    <br/><br/>
    <div class="subgroup">
      <label for="date">Date:</label>
      <input type="date" name="date" id="date" required/>
    </div>
    <br/><br/>
  </form>
</div>
```

```

<div class="subgroup">
  <label for="range">Range:</label>
  <small>0 </small><input type="range" name="range" id="range" min="0" max="100" required/><small> 100</small>
</div>
<br/><br/>
<div class="subgroup">
  <label for="tel">Phone:</label>
  <input type="tel" name="tel" id="tel" placeholder="123456789" pattern="[0-9]{9}" required/>
</div>
<br/><br/>
<div class="subgroup">
  <label for="url">Url:</label>
  <input type="url" name="url" id="url" placeholder="https://github.com" required/>
</div>
</form>
<div class="subgroup">
  <input type="submit" id="button" value="Mostrar resultados" onclick="showResults()">
</div>
</div>

```

Y aquí está el segundo div, el que mostrará los datos:

```

<div id="container-right">
  <h3>Results</h3>
  <div class="subgroup-right"><label>Password:</label><span id="password-result" class="monospace-result"></span></div>
  <br/><br/>
  <div class="subgroup-right"><label>Email:</label><span id="email-result" class="monospace-result"></span></div>
  <br/><br/>
  <div class="subgroup-right"><label>Date:</label><span id="date-result" class="monospace-result"></span></div>
  <br/><br/>
  <div class="subgroup-right"><label>Range:</label><span id="range-result" class="monospace-result"></span></div>
  <br/><br/>
  <div class="subgroup-right"><label>Phone:</label><span id="phone-result" class="monospace-result"></span></div>
  <br/><br/>
  <div class="subgroup-right"><label>Url:</label><span id="url-result" class="monospace-result"></span></div>
</div>
</body>
</html>

```

En cuanto al javascript, abajo del todo añadimos el listener que ejecutará la función start() cuando cargue la página web:

```
54     window.addEventListener( type: "load", start, options: false);  
55 }
```

Después, declaramos los diferentes elementos html que usaremos. En este caso cada uno de los inputs, el form y cada uno de los sections que mostrarán la información:

```
8  {  
9      let inputPassword;  
10     let resultPassword;  
11     let inputEmail;  
12     let resultEmail;  
13     let inputDate;  
14     let resultDate;  
15     let inputRange;  
16     let resultRange;  
17     let inputPhone;  
18     let resultPhone;  
19     let inputUrl;  
20     let resultUrl;  
21     let form;
```

En la función start() inicializaremos las variables declaradas anteriormente, vinculándolas a los respectivos elementos del HTML:

```
23     function start() :void {
24         inputPassword = document.getElementById( elementId: "password");
25         resultPassword = document.getElementById( elementId: "password-result");
26         inputEmail = document.getElementById( elementId: "email");
27         resultEmail = document.getElementById( elementId: "email-result");
28         inputDate = document.getElementById( elementId: "date");
29         resultDate = document.getElementById( elementId: "date-result");
30         inputRange = document.getElementById( elementId: "range");
31         resultRange = document.getElementById( elementId: "range-result");
32         inputPhone = document.getElementById( elementId: "tel");
33         resultPhone = document.getElementById( elementId: "phone-result");
34         inputUrl = document.getElementById( elementId: "url");
35         resultUrl = document.getElementById( elementId: "url-result");
36
37         form = document.forms["form"];
38     }
```

Por último, la función showResults(), que será a la que llame el botón cuando es presionado, gracias al onclick que tiene. Esta función chequea la validez del formulario mediante reportValidity() y, en caso de que no sea válido, muestra al usuario cual es el error en el formulario. Si el formulario es válido, entra dentro del if y setea el texto interno de cada section al valor que tienen sus correspondientes inputs:

```
40     function showResults() :void {
41         if (form.reportValidity()) {
42             resultPassword.innerText = inputPassword.value;
43             resultEmail.innerText = inputEmail.value;
44             resultDate.innerText = inputDate.value;
45             resultRange.innerText = inputRange.value;
46             resultPhone.innerText = inputPhone.value;
47             resultUrl.innerText = inputUrl.value;
48         }
49     }
```


Por último, y aunque esto no tiene nada que ver con el comportamiento de la página, aquí dejo el código del css que hice:

```
h3 {
  text-align: center;
  font-size: 25px;
}

label {
  margin-right: 10px;
}

input {
  flex: 1;
  min-width: 1px;
}

span {
  flex: 1;
  min-width: 1px;
  text-align: right;
}

.subgroup {
  display: flex;
  align-items: center;
  justify-content: center;
  width: 100%;
}

.subgroup-right {
  display: flex;
  align-items: center;
  width: 100%;
  border-bottom: dashed 2px dimgrey;
}

.monospace-result {
  font-family: monospace;
  color: dimgrey;
  font-weight: bolder;
}

#container-left {
  top: 5%;
  left: 5%;
  right: 52.5%;
  bottom: 5%;
  position: absolute;
  border: solid 2px black;
  border-radius: 10px;
  padding: 0 5% 5% 5%;
  background-color: #ededed;
}
```

```

#container-right {
  top: 5%;
  left: 52.5%;
  right: 5%;
  bottom: 5%;
  position: absolute;
  border: solid 2px black;
  border-radius: 10px;
  padding: 0 5% 5% 5%;
  background-color: #ededed;
}

input[type="submit"] {
  position: absolute;
  flex: 1;
  width: 76.5%;
  min-width: 1px;
  text-align: center;
  bottom: 0;
  padding: 1%;
  margin: 5%;
  background-color: dodgerblue;
  color: #dbe9fb;
  font-weight: bolder;
  border-color: #dbe9fb;
  border-radius: 10px;
}

```

```

input[type="submit"]:hover {
  background-color: #dbe9fb;
  color: dodgerblue;
  font-weight: bolder;
  border-color: dodgerblue;
  border-radius: 10px;
}

```

El programa en ejecución:

Inputs

Password:

Email:

Date: ☐

Range: 0 100

Phone:

Url:

Results

Password: _____

Email: _____

Date: _____

Range: _____

Phone: _____

Url: _____

Si das al botón, pero hay algún campo incorrecto:

Inputs

Password:

Email:

Date: ☐

Range: 0 100

Phone:

Url:

Results

Password: _____

Email: _____

Date: _____

Range: _____

Phone: _____

Url: _____

Si pulsas el botón después de meter todos los campos correctamente:

Inputs

Password:

Email:

Date: ☐

Range: 0 100

Phone:

Url:

Results

Password: _____

Email: _____

Date: _____

Range: _____

Phone: _____

Url: _____