

PWD- Programación Web Dinámica



**Validar
formularios**



Universidad Nacional del Comahue
Facultad de Informática
Departamento de Programación

Validar Formularios

¿Dónde es mas conveniente chequear si los datos de un formulario se completaron y si además se realizo correctamente?

- En el cliente??...
- En el servidor??...
- Que puedo usar??

Validación con JavaScript

- La validación de los datos de un formulario mediante scripts JavaScript no sustituye a la validación que debe realizarse, por motivos de seguridad, en la aplicación del servidor que recibe la información.
- Una validación local con JavaScript mejora la **experiencia de usuario**, al no ser necesario enviar los datos al servidor y esperar su respuesta para obtener sólo un mensaje informando de la incorrección de la información suministrada.

Validando Formularios

- Algunas de las validaciones típicas:
 - Comprobar que se han suministrado todos los campos obligatorios.
 - Comprobar que el formato de un campo es el esperado (fechas, teléfonos, etc.).
 - Comprobar la validez (sintáctica) de las direcciones de correo y URLs.
 - Comprobar que no se sobrepasa la longitud, número de líneas o tamaño de la entrada.

Validando Formularios

- La validación de campos de formulario se basa en capturar el momento en que el usuario realiza el envío de los datos del formulario.
- Cuando se pulsa el botón de ENVIAR, que es de tipo “submit”.
- Cuando *submitimos* el formulario.
- ¿?

Evento onSubmit

- Cuando se produce el evento “onSubmit” ?
 - Cuando el usuario pulsa sobre el botón de enviar se genera el evento ***submit***.
- Como se captura?
 - A partir del atributo ***onSubmit*** en la etiqueta del formulario.

Validando Formularios

- Ejemplo:

```
<form name="form1" method="post" action="accion/destino.php" onSubmit="return  
validar();">
```

```
  Ingrese dato: <input name="dato_form" type="text" id=" dato_form " /><br/>
```

```
  <input type="submit" name="Submit" value="Aceptar" />
```

```
</form>
```

- Agregamos el atributo onSubmit al form y llamamos a la función javascript que valide los datos del formulario.
- Si la función devuelve false los datos no se envían al servidor.

Validar Formularios

```
<script type="text/javascript">
```

```
function validar(){  
    resp = true;  
    var myDato = document.getElementById('dato_form').value;  
    if(myDato == ""){  
        alert("Los datos son requeridos");  
        resp false;  
    }  
    return resp;  
}  
</script>
```


Validar Formularios

- Si el código JavaScript del atributo ***onSubmit*** devuelve **false**, la validación es incorrecta y el navegador no continúa con el envío del formulario.
- Si devuelve **true** la validación es correcta y el formulario se envía normalmente.

Validar Formularios / HTML 5

- **HTML5** añadió nuevos **tipos semánticos** para el elemento `<input>` y validación de restricciones.
- Se pueden usar restricciones básicas y comunes, sin la necesidad de JavaScript.
- Para restricciones más complejas se puede usar la API de *Validación de Restricciones*.

Validar Formularios / HTML5

- Restricciones intrínsecas `<input>`
- Asignar el valor semántico más apropiado para el atributo `type`:
 - `<input type="URL">`
 - `<input type="email">`

HTML5 / Atributos para validación

Atributo	Tipos de input que soportan el atributo	Valores posibles	Descripción	Incumplimiento asociado
pattern	text, search, url, tel, email, password	Una expresión regular de JavaScript (compilada con las banderas <code>global</code> , <code>ignoreCase</code> , y <code>multiline</code> de ECMAScript 5 deshabilitadas)	El valor debe coincidir con el patrón.	Incumplimiento de restricción por incompatibilidad de patrones (Pattern mismatch)
min	range, number	Un número válido	El valor debe ser mayor o igual al de este atributo.	Incumplimiento de restricción por flujo insuficiente (Underflow)
	date, month, week	Una fecha válida		
	datetime, datetime-local, time	Una fecha y hora válidos		
max	range, number	Un número válido	El valor debe ser menor o igual al de este atributo	Incumplimiento de restricción por desborde (Overflow)
	date, month, week	Una fecha válida		
	datetime, datetime-local, time	Una fecha y hora válidos		
required	text, search, url, tel, email, password, date, datetime, datetime-local, month, week, time, number, checkbox, radio, file; también en los elementos <code><select></code> y <code><textarea></code>	<i>ninguno</i> , pues éste es un atributo de tipo Boolean: su presencia representa <i>true</i> , y su ausencia representa <i>false</i>	Debe tener un valor (si se establece).	Incumplimiento de restricción por ausencia (Missing)
step	date	Un número entero de días	A menos que se establezca el atributo con la literal <code>any</code> , el valor debe ser <code>min</code> + un enter múltiplo del valor de este atributo.	Incumplimiento de restricción por inconsistencia de paso (Step mismatch)
	month	Un número entero de meses		
	week	Un número entero de semanas		
	datetime, datetime-local, time	Un número entero de segundos		
	range, number	Un entero		
maxlength	text, search, url, tel, email, password; también en el elemento <code><textarea></code>	Una longitud en enteros	El número de caracteres (puntos de código) no debe exceder el valor del atributo.	Incumplimiento de restricción por ser muy largo (Too long)

HTML5 / Validaciones complejas

- Necesidad de validar regla del negocio propias del dominio.
- Validar mas de un de un elemento de un formulario o realizar cálculos complejos.
- Implementar restricciones complejas usando la API de Restricciones de HTML5.

Alguna consulta ?

