



Linkia FP

| | | | | |
|-----|-----|-----|----------|--------------------------------|
| DAW | M06 | T05 | Material | Cookies, eventos y formularios |
|-----|-----|-----|----------|--------------------------------|

```
<form type="password" action="codigo.php" method="get" name="formularioEjemplo">
```

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

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

```
<input type="radio" name="sex" value="male">
<input type="radio" name="sex" value="female">
```

```
<input type="radio" name="edat" value=">18">
<input type="radio" name="edat" value="<18">
```

```
<input type="checkbox" name="vehicle" value="bike"> I have a bike <br />
<input type="checkbox" name="vehicle" value="car"> I have a car <br />
```

```
<select name="dias_pc" >
```

```
<option value="1">uno</option>
```

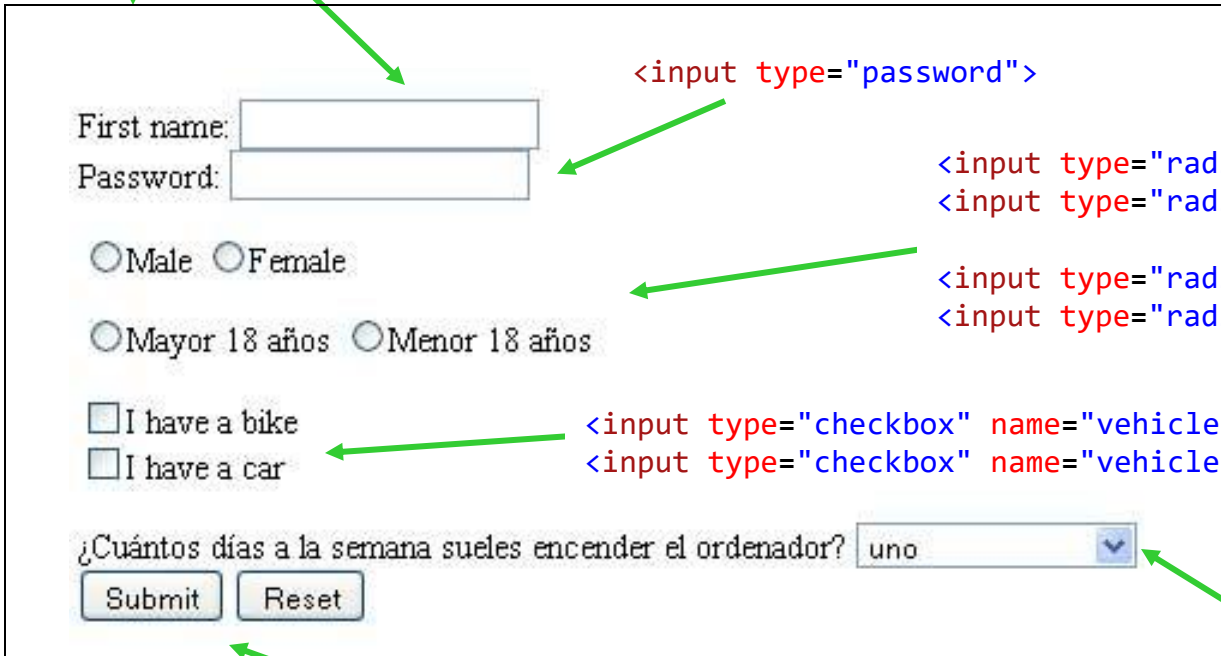
```
<option value="2">dos</option>
```

```
</select>
```

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

```
<input type="reset" value="Reset" />
```

```
</form>
```



HTML

```
<form name="miFormulario" >
```

```
<input type="text" name="texto" />
```

JS

```
var formulario = document.forms["miFormulario"];
```

```
var text = formulario["texto"].value;
```

- Para validar un formulario antes de ser enviado, se debe realizar **onsubmit**. Si se retorna un **false**, no se enviará.

```
<form action="codigo.php" onsubmit="return validaForm();">  
  Nombre <input type="text" name="name" onblur="validaNom()" >  
  <input type="submit" >  
</form>
```

```
function validaForm(){  
  var formulario = document.forms["miFormulario"];  
  var text = formulario["name"].value;  
  if(text=="pedro"){  
    return true;  
  }  
  return false;  
}
```

Este formulario solo se enviará cuando el nombre sea "pedro"

```
Hombre<input type="radio" name="sex" value="male">  
Mujer<input type="radio" name="sex" value="female">
```

```
var radios = formulario["sex"]; //array con los radio de name="sex"  
var radio="";  
for(var k =0; k< radios.length; k++){  
    if(radios[k].checked == true){  
        radio= radios[k].value;  
        break;  
    }  
}
```

```
<input type="checkbox" name="vehicle" value="bike"> I have a bike <br />  
<input type="checkbox" name="vehicle" value="car"> I have a car <br />
```

```
var checkboxes = formulario["vehicle"]; //array con los checkbox de  
name="vehicle"  
for(var k =0; k < checkboxes.length; k++){  
    if(checkboxes[k].checked== true){  
        alert("selected: "+checkboxes[k].value);  
    }  
}
```

```
<select name="listado" value="-1" multiple>
    <option value="1"> uno</option>
    <option value="2"> dos</option>
    <option value="3"> tres</option>
</select>
```

```
//comprovamos que se han seleccionado dos elementos
var selector = formulario["listado"];
var indexSeleccionado=selector.selectedIndex; //primer elemento seleccionado
var optionsArr = selector.options;//array con las opciones

if( indexSeleccionado ==-1){  /*no se ha seleccionado ningun option.*/  }

for(var k =0; k < optionsArr.length; k++){ //recorremos el array de opciones
    if(optionsArr[k].selected == true){
        alert("selected: "+optionsArr[k].text);
    }
}
```

Objeto RegExp.test(texto)

```
var texto = 'cristia@gmail.com';

//creamos un objeto de expresion regular
var patt1=new RegExp("[A-z]{1,5}$");
if ( patt1.test(texto) ){ //comprovamos si se hace el matching con test
    alert ('Nombre valido');
} else {      alert('Nombre  invalido');    }
```

texto.match(Expresion Regular)

```
var texto = 'cadena de texto a validar';

//creamos la expresion regular como una string
var RegExPattern = /^[^0-9][A-z]{1,5}$/;

if ( texto.match(RegExPattern) ) { //comprovamos el matching con match
    alert('Password valido');
} else {      alert('Password invalido');    }
```

cristian@gmail.co
m

Guille@hotmail.com

Empieze por

FIN

$^ [A-z0-9]^+$

Cualquier texto
sin espacios

@

Una @

$[A-z0-9]^+$

Cualquier texto
sin espacios

.

Un punto

$[A-Za-z0-9]\{2,3\} \$$

De dos a tres
caracteres sin
espacios

$^$ inicio , $\$$ final

$[A-z]$ rango permitido , $^[^A-z]$ rango prohibido , $(a|b|c)$ alternativas

$\{1,5\}$ de 1 a 5 repeticiones , $*$ de 0 a infinitas repeticiones , $+$ de 1 a infinitas repeticiones

•Utiliza los elementos para crear la expresión regular que concuerde con un email:

$^ + @ .$

$[A-Za-z0-9]\{2,3\}$

https://www.w3schools.com/jsref/dom_obj_event.asp
<https://developer.mozilla.org/en-US/docs/Web/Events>

EVENTOS

- Un evento define qué interacción se debe producir para ejecutar un código.

MOUSE EVENTS

onclick, ondblclick,
onmousedown, onmousemove,
onmouseover, onmouseout,
onmouseup

KEYBOARD EVENTS

- onkeydown, onkeypress, onkeyup

OBJECT EVENTS

- onload, onresize, onscroll, onunload

FORM EVENTS

- onblur, onfocus, onselect, onchange , onsubmit, onreset
-

ESTABLECER CAPTURADOR EN CÓDIGO

```
<div id="divAzul" onclick="ejecutaEsto();" > </div>  
<body onload="ejecutaEsto()">
```

VINCULANDO MÚLTIPLES FUNCIONES A UN EVENTO

```
<body onload="ejecutaEsto(); estoTambien();" >
```

PODEMOS ASIGNAR FUNCIÓN ANÓNIMA O FUNCIÓN CREADA

```
var miNodo= document.getElementById("ex");  
var miFuncion = function (){ //codigo a ejecutar };  
miNodo.onclick = function(){    }; //ejecuta al clickar en miNodo  
window.onload = miFuncion //ejecuta al cargar la web
```

PROBLEMA VINCULANDO MÚLTIPLES FUNCIONES AL MISMO EVENTO

```
var miFuncion = function (){ //codigo a ejecutar };  
var miFuncion2 = function (){ //codigo a ejecutar2 };  
miNodo.onclick = miFuncion ; //ejecuta al clickar en miNodo  
miNodo.onclick = miFuncion2 ; //ejecuta al clickar en miNodo  
//qué funciones ejecutará?
```

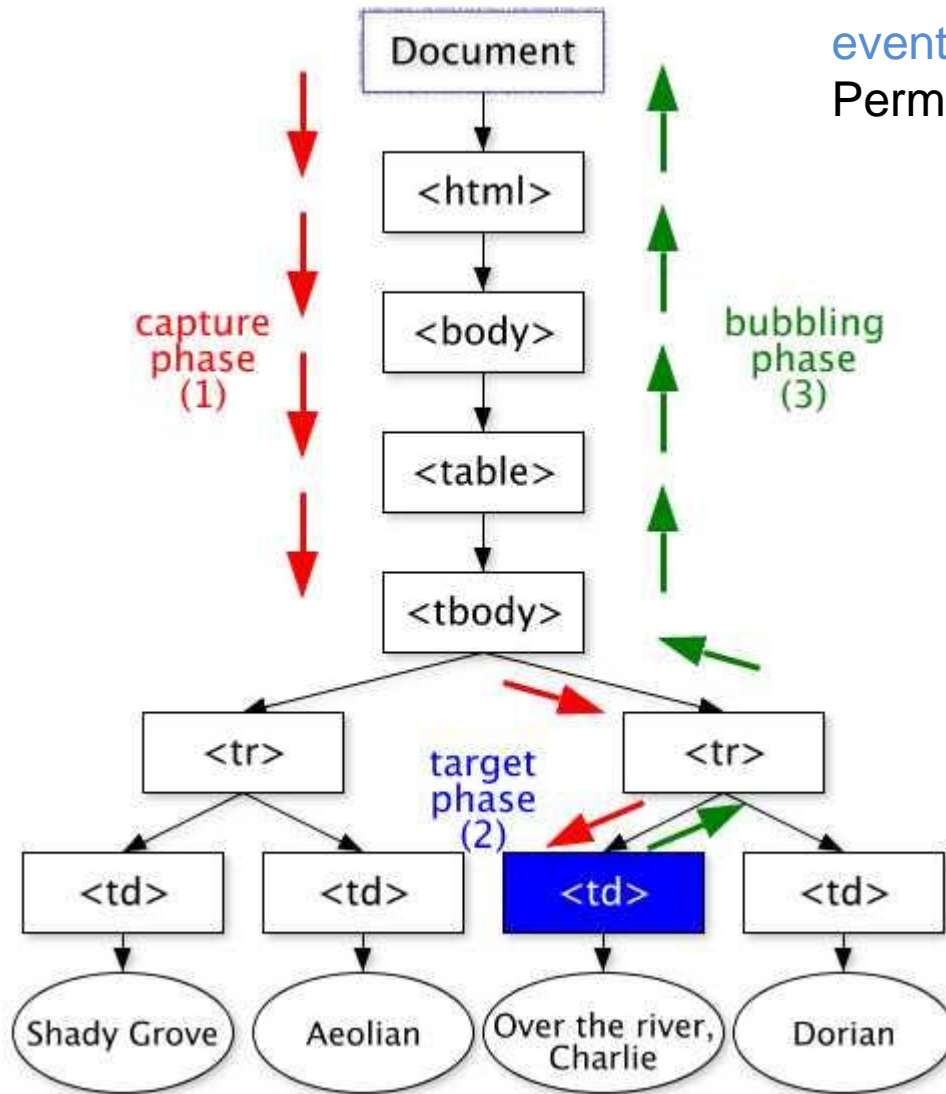
addEventListener PARA AÑADIR EVENTOS

```
var miNodo= document.getElementById("ex");  
var miFuncion = function (){ //codigo a ejecutar };  
var miFuncion2 = function (){ //codigo a ejecutar2 };  
  
miNodo.addEventListener("click", miFuncion ,false);  
miNodo.addEventListener("click", miFuncion2 ,false);  
  
window.addEventListener("load", miFuncion ,false);  
window.addEventListener("load", miFuncion2 ,false);
```

EL ÚLTIMO PARÁMETRO INDICA LA FASE DE CAPTURA

False el evento se captura en la bubbling fase.

True el evento se captura en la fase de captura del evento.



`event.stopPropagation();`

Permite detener la propagación del evento

- Siempre que una función es llamada por un evento recibe como parámetro una variable del tipo Event según el tipo de event realizado(http://www.w3schools.com/jsref/dom_obj_event.asp)
- Por ejemplo, en un MouseEvent podremos obtener el botón clicado, o en un keyboardEvent la tecla pulsada.

```
miDiv.onclick = miFuncion ; //ejecuta miFuncion al clicar sobre el
elemnto miDiv

var miFuncion = function (event){
    console.log("boton pulsado:"+event.button);
};
```

- Para saber la letra pulsada, deberemos preguntar a keyCode, charCode o which según el evento realizado y navegador (<http://asquare.net/javascript/tests/KeyCode.html>)

```
miInput.onKeyDown = miFuncion2 ;//ejecuta miFuncion al escribir
sobre el elemnto miInpu

var miFuncion2 = function (event){
    console.log("tecla pulsada:"+event.keyCode);
};
```

- Suponiendo que **funClick** sea la función que se ejecuta al capturar un evento que no queremos que realice su acción por defecto:
- En éste ejemplo evitamos el menú desplegable al pulsar el botón derecho

```
window.addEventListener("contextmenu", funClick ,false);  
var funClick = function(evt){  
    evt.preventDefault();  
};
```

- Variables vinculadas a un sitio web almacenadas en el navegador y el servidor.
- Se envían del navegador al servidor y viceversa.
- Se puede acceder siempre desde el servidor, y en ocasiones desde el navegador con JS.

Crear / Modificar /Borrar / Leer una Cookie de forma nativa

```
document.cookie='nom=James';
```

```
document.cookie='apellido=Bond';
```

```
document.cookie='nom=Jaime';
```

```
document.cookie='nom=James;expires=Thu, 2 Aug 2013 20:47:11 UTC;' ;
```

```
textoGuardado=hola; nom=Jaime; apellido=Bond
```

- Podemos utilizar el siguiente código para crear/modificar el valor de una cookie

```
function setCookie(cname,cvalue,exdays) {  
    var d = new Date();  
    d.setTime(d.getTime()+(exdays*24*60*60*1000));  
    var expires = "expires="+d.toGMTString();  
    document.cookie = cname + "=" + cvalue + "; " + expires;  
}
```

- ***cname*** es el nombre de la cookie a guardar y ***cvalue*** es el nombre del valor.
- ***exdays*** es el número de días que va a vivir la cookie. Con un valor negativo la cookie se borrará

Ejemplo de utilización:

```
function guardarValor(){  
    var texto = window.prompt("indica un valor");  
  
    setCookie("textoGuardado",texto,5);  
}
```

- Podemos utilizar el siguiente código para crear/modificar el valor de una cookie

```
function getCookie(cname) {  
    var name = cname + "=";  
    var ca = document.cookie.split(';');  
    for(var i=0; i<ca.length; i++)    {  
        var c = ca[i].trim();  
        if (c.indexOf(name)==0) return  
c.substring(name.length,c.length);  
    }  
    return "";  
}
```

Leer una Cookie (w3schools)

```
function obtineValor(){  
    var texto = getCookie("textoGuardado");  
    console.log(texto)  
}
```

Formulario con dos checkboxes y dos inputs. Se ha de enviar si:
el primer inputs tiene entre 3 y 6 caracteres.
el segundo input tiene una fecha. (`Date.parse("2010-02-20")`)
Los dos checkboxes están seleccionados.

Añade un botón que guarde el valor del primer input en una cookie.

En un formulario, al escribir en un input mostrar por consola el valor que contiene el input.



Linkia FP

Formación Profesional
Oficial a Distancia