

The logo for neitek, featuring the word "neitek" in a lowercase, sans-serif font. The "i" is green, and there is a green underline under the "k". The logo is set against a white background within a dark brown rectangular box.

neitek

# JavaScript

## parte 3

Carolina Caballero

# Agenda

- Eventos
- Formularios
- Objetos Predefinidos

Eventos

# Eventos de JavaScript

- Los Eventos de HTML son “cosas” que les pasan a los elementos HTML.
- JavaScript puede “reaccionar” a esos eventos.
- Un evento puede ser algo que el browser hace, o que el usuario hace

## Ejemplos de Eventos

- Cuando se termina de cargar una página.
- Un click del mouse
- Carga de una imagen o página web
- Pasar el mouse por encima de un punto del documento
- Seleccionar un campo de un formulario
- Enviar un formulario XHTML
- Presionar una tecla

# Sintaxis

- Se puede definir con comillas simples o dobles:

```
<element event='some JavaScript'>
```

```
<element event="some JavaScript">
```

- Ejemplo:

```
<button onclick="document.getElementById('demo').innerHTML = Date()">The time is?</button>
```

# Modelo Básico de Eventos

- Cada elemento o etiqueta XHTML define su propia lista de posibles eventos
- Un mismo tipo de evento puede estar definido para varios elementos XHTML diferentes
- Un mismo elemento XHTML puede tener asociados varios eventos diferentes

# Modelo Básico de Eventos

Evento	Descripción	Elementos para los que esta definido
onblur	Deseleccionar un elemento	<buton>, <input>, <label>, <select>, <textarea>, <body>
onchange	Deseleccionar un elemento que se ha modificado	<input>, <select>, <textarea>
onclick	Presionar y soltar el ratón	Todos los elementos
ondblclick	Presionar dos veces seguidas con el ratón	Todos los elementos
onfocus	Seleccionar un elemento	<buton>, <input>, <label>, <select>, <textarea>, <body>
onkeydown	Pulsar una tecla (sin soltar)	Elementos de formulario y <body>
onkeypress	Pulsar una tecla	Elementos de formulario y <body>
onkeyup	Soltar una tecla pulsada	Elementos de formulario y <body>
onload	La página se ha cargado completamente	<body>



# Modelo Básico de Eventos

Evento	Descripción	Elementos para los que esta definido
onmousedown	Pulsar (sin soltar) un boton del ratón	Todos los elementos
onmousemove	Mover el ratón	Todos los elementos
onmouseout	El ratón sale de elemento (pasa por encima de otro elemento)	Todos los elementos
onmouseover	El ratón entra en el elemento (pasa por encima del elemento)	Todos los elementos
onmouseup	Soltar el botón que estaba pulsando en el ratón	Todos los elementos
onreset	Inicializar el formulario	<form>
onresize	Se ha modificado el tamaño de la ventana del navegador	<body>
onselect	Seleccionar un texto	<input>, <textarea>
onsubmit	Enviar el formulario	<form>
onunload	Se abandona la página	<body>

# Manejadores de Eventos (event handlers)

- Son las funciones o código de JavaScript que se define para los eventos.
- Existen diferentes formas para definirlos:
  - Como atributos de los elementos XHTML
  - Como funciones Javascript externas
  - Manejadores semánticos

# Manejadores de Eventos como atributos XHTML

- Es el más sencillo y el menos profesional
- El código se incluye en un atributo del propio elemento

```
<input type="button" value="Píname y verás"  
onclick="alert('Gracias por pinchar');"/>
```

# Variable this

- Javascript define una variable especial llamada **this** que se crea automáticamente.
- Sin *this*:

```
<div id="contenidos" style="width:150px; height:60px; border:thin solid silver"
onmouseover="document.getElementById('contenidos').style.borderColor='black';"
onmouseout="document.getElementById('contenidos').style.borderColor='silver';">
  Sección de contenidos...
</div>
```

- Con *this*:

```
<div id="contenidos" style="width:150px; height:60px; border:thin solid silver"
onmouseover="this.style.borderColor='black';"
onmouseout="this.style.borderColor='silver';">
  Sección de contenidos...
</div>
```

# Manejadores de Eventos como Funciones Externas

```
<input type="button" value="Pinchame y verás"
onclick="alert('Gracias por pinchar');" />
```



```
function muestraMensaje() {
    alert('Gracias por pinchar');
}
```

```
<input type="button" value="Pinchame y verás" onclick="muestraMensaje()" />
```

# Manejadores de Eventos Semánticos

```
<input type="button" value="Pinchame y verás"  
onclick="alert('Gracias por pinchar');" />
```



```
<script>  
// Función externa  
function mensaje() {  
    alert('Gracias por pinchar');  
}  
  
// Asignar la función externa al elemento  
window.onload = function() {  
    document.getElementById("btn1").onclick = mensaje;  
}  
</script>
```

```
<input id="btn1" type="button" value="Pinchame y verás" />
```

- File: Sample18.html

# Ejercicios

- Escribe una página con la siguiente funcionalidad:
  - Botón al dar click desplegar Fecha y Hora en un campo field.
  - Un recuadro, que al dar doble click se haga 10px mas grande a lo ancho.
  - Al cargar la pagina pida el nombre al usuario, desplegarlo en un tag `<h1>`
  - Boton que cambie de color por al menos 2 eventos diferentes.

# Agenda

- Eventos
- Formularios
- Objetos Predefinidos



Formularios

# Formularios

- Una de las funciones principales de JavaScript es realizar validaciones de los datos capturados desde el lado del cliente y no en el servidor.
- Los campos deben definirse en un objeto “form” para poderlos posteriormente validar.
- Se puede tener mas de una forma en una pagina.

# Formularios

- Se puede acceder a los formularios a través de la variable forms del objeto document

```
document.forms[0];
```

- Además crea automáticamente un array llamado elements para cada formulario

```
document.forms[0].elements[0];
```

# Formularios

- Es una buena práctica es asignarle un nombre a los *forms*, de esta manera el código puede acceder con facilidad.

```
var formularioPrincipal = document.formulario;  
var formularioSecundario = document.otro_formulario;
```

```
<form name="formulario" >
```

```
...
```

```
</form>
```

```
<form name="otro_formulario" >
```

```
...
```

```
</form>
```

# Formularios

- Los elementos de los formularios también se pueden acceder directamente con el atributo *name*.

```
var formularioPrincipal = document.formulario;  
var primerElemento = document.formulario.elemento;  
  
<form name="formulario">  
  <input type="text" name="elemento" />  
</form>
```

- O directamente con *getElementById*
- File: Sample19.html

# Propiedades de cada Elemento

- *type*: tipo del elemento
- *form*: referencia al formulario al que pertenece el elemento
- *name*: atributo name del XHTML
- *value*: permite leer y modificar el atributo valor del elemento

# Eventos más usados

- onclick - click del mouse sobre el elemento
- onchange - elemento ha cambiado de valor
- onfocus - el usuario selecciona un elemento
- onblur - cuando el elemento pierde el foco

# Obtener valor de los campos

- En el cuadro de texto y textarea se usa la propiedad *value*

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

```
var valor = document.getElementById("texto").value;
```

```
<textarea id="parrafo"></textarea>
```

```
var valor = document.getElementById("parrafo").value;
```



# Obtener valor de los campos

- Para radiobutton y checkbox se usa la propiedad *checked*

```
<input type="radio" value="si" name="pregunta" id="pregunta_si"/> SI  
<input type="radio" value="no" name="pregunta" id="pregunta_no"/> NO  
<input type="radio" value="nsnc" name="pregunta" id="pregunta_nsnc"/> NS/NC
```

```
var elementos = document.getElementsByName("pregunta");  
  
for(var i=0; i<elementos.length; i++) {  
    alert(" Elemento: " + elementos[i].value + "\n Seleccionado: " + elementos[i].checked);  
}
```

- File: Sample21.html

# Obtener valor de los campos

- Para listas desplegables

```
<select id="opciones" name="opciones">
  <option value="1">Primer valor</option>
  <option value="2">Segundo valor</option>
  <option value="3">Tercer valor</option>
  <option value="4">Cuarto valor</option>
</select>
```

```
// Obtener la referencia a la lista
var lista = document.getElementById("opciones");

// Obtener el índice de la opción que se ha seleccionado
var indiceSeleccionado = lista.selectedIndex;
// Con el índice y el array "options", obtener la opción seleccionada
var opcionSeleccionada = lista.options[indiceSeleccionado];

// Obtener el valor y el texto de la opción seleccionada
var textoSeleccionado = opcionSeleccionada.text;
var valorSeleccionado = opcionSeleccionada.value;
```

```
alert("Opción seleccionada: " + textoSeleccionado + "\n Valor de la opción: "
+ valorSeleccionado);
```

● File: Sample21.html

# Ejercicio

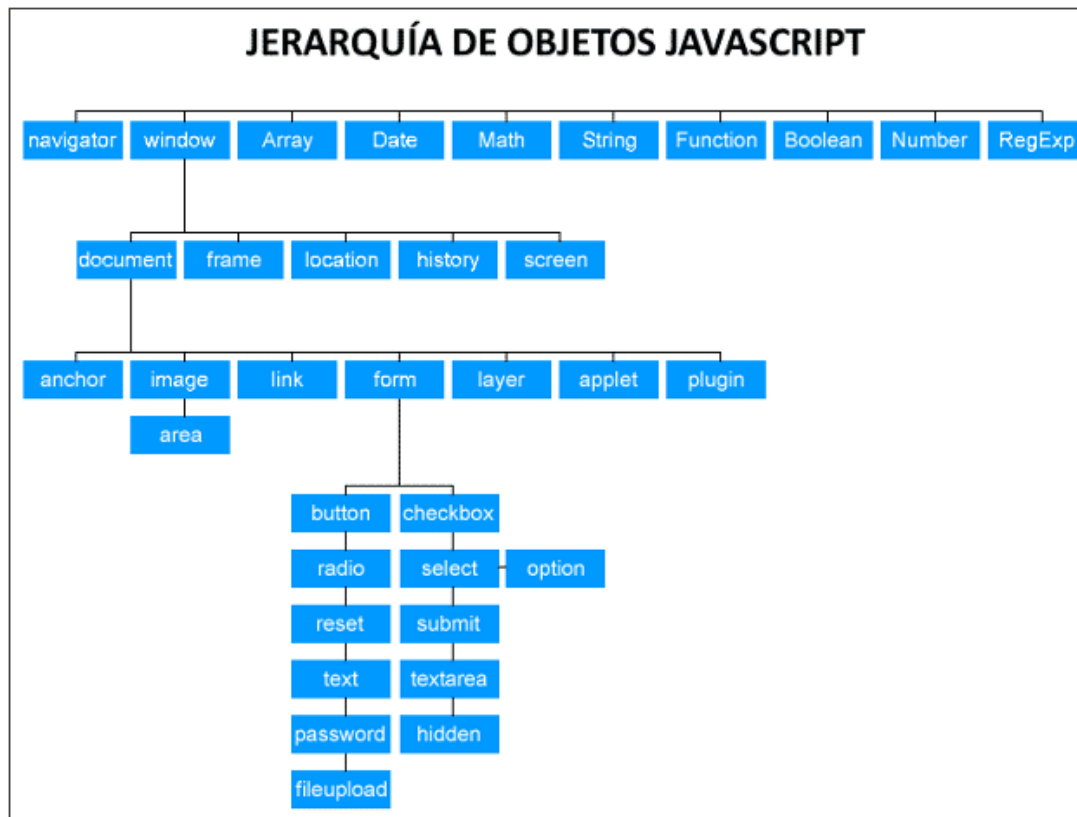
- Hacer una página para capturar información y validar los campos:
  - Nombre – obligatorio, max 60, no acepta números
  - Fecha de Nacimiento – 3 campos: texto para día (2 dig), lista de seleccion para mes, texto para año (4 digitos), calcular y desplegar edad actual.
  - RFC – opcional, valida formato RFC (YYMMDDAAAA-\*\*\*)
  - Sexo – Radio button, validar que seleccione una opción.
  - Para los campos Nombre y Edad, no debe de permitir moverse a otro campo hasta que lo haya capturado.
  - Presentar un resumen de lo capturado.

# Agenda

- Eventos
- Formularios
- Objetos Predefinidos

# Objetos Predefinidos

# Jerarquia de objetos



# Objeto Window

- Representa la ventana en el navegador.

Property	Description
<u>closed</u>	Returns a Boolean value indicating whether a window has been closed or not
<u>defaultStatus</u>	Sets or returns the default text in the statusbar of a window
<u>document</u>	Returns the Document object for the window ( <a href="#">See Document object</a> )
<u>frameElement</u>	Returns the <iframe> element in which the current window is inserted
<u>frames</u>	Returns all <iframe> elements in the current window
<u>history</u>	Returns the History object for the window ( <a href="#">See History object</a> )
<u>innerHeight</u>	Returns the inner height of a window's content area
<u>innerWidth</u>	Returns the inner width of a window's content area
<u>length</u>	Returns the number of <iframe> elements in the current window
<u>localStorage</u>	Returns a reference to the local storage object used to store data. Stores data with no expiration date
<u>location</u>	Returns the Location object for the window ( <a href="#">See Location object</a> )
<u>name</u>	Sets or returns the name of a window
<u>navigator</u>	Returns the Navigator object for the window ( <a href="#">See Navigator object</a> )
<u>opener</u>	Returns a reference to the window that created the window
<u>outerHeight</u>	Returns the outer height of a window, including toolbars/scrollbars
<u>outerWidth</u>	Returns the outer width of a window, including toolbars/scrollbars

# Objeto Window

<u>parent</u>	Returns the parent window of the current window
<u>screen</u>	Returns the Screen object for the window ( <u>See Screen object</u> )
<u>screenLeft</u>	Returns the horizontal coordinate of the window relative to the screen
<u>screenTop</u>	Returns the vertical coordinate of the window relative to the screen
<u>screenX</u>	Returns the horizontal coordinate of the window relative to the screen
<u>screenY</u>	Returns the vertical coordinate of the window relative to the screen
sessionStorage	Returns a reference to the local storage object used to store data. Stores data for one session (lost when the browser tab is closed)
scrollX	An alias of <u>pageXOffset</u>
scrollY	An alias of <u>pageYOffset</u>
<u>self</u>	Returns the current window
<u>status</u>	Sets or returns the text in the statusbar of a window
<u>top</u>	Returns the topmost browser window



# Objeto Window

Method	Description
<u>alert()</u>	Displays an alert box with a message and an OK button
<u>atob()</u>	Decodes a base-64 encoded string
<u>blur()</u>	Removes focus from the current window
<u>btoa()</u>	Encodes a string in base-64
<u>clearInterval()</u>	Clears a timer set with setInterval()
<u>clearTimeout()</u>	Clears a timer set with setTimeout()
<u>close()</u>	Closes the current window
<u>confirm()</u>	Displays a dialog box with a message and an OK and a Cancel button
<u>focus()</u>	Sets focus to the current window
<u>getComputedStyle()</u>	Gets the current computed CSS styles applied to an element
<u>getSelection()</u>	Returns a Selection object representing the range of text selected by the user
<u>matchMedia()</u>	Returns a MediaQueryList object representing the specified CSS media query string

# Objeto Window

<u>moveBy()</u>	Moves a window relative to its current position
<u>moveTo()</u>	Moves a window to the specified position
<u>open()</u>	Opens a new browser window
<u>print()</u>	Prints the content of the current window
<u>prompt()</u>	Displays a dialog box that prompts the visitor for input
<u>resizeBy()</u>	Resizes the window by the specified pixels
<u>resizeTo()</u>	Resizes the window to the specified width and height
<u>scroll()</u>	<b>Deprecated.</b> This method has been replaced by the <u>scrollTo()</u> method.
<u>scrollBy()</u>	Scrolls the document by the specified number of pixels
<u>scrollTo()</u>	Scrolls the document to the specified coordinates
<u>setInterval()</u>	Calls a function or evaluates an expression at specified intervals (in milliseconds)
<u>setTimeout()</u>	Calls a function or evaluates an expression after a specified number of milliseconds
<u>stop()</u>	Stops the window from loading

# Objeto Navigator

- Contiene información acerca del navegador (Browser).

Property	Description
<u>appCodeName</u>	Returns the code name of the browser
<u>appName</u>	Returns the name of the browser
<u>appVersion</u>	Returns the version information of the browser
<u>cookieEnabled</u>	Determines whether cookies are enabled in the browser
<u>geolocation</u>	Returns a Geolocation object that can be used to locate the user's position
<u>language</u>	Returns the language of the browser
<u>onLine</u>	Determines whether the browser is online
<u>platform</u>	Returns for which platform the browser is compiled
<u>product</u>	Returns the engine name of the browser
<u>userAgent</u>	Returns the user-agent header sent by the browser to the server

# Objeto Navigator

Method	Description
<u>javaEnabled()</u>	Specifies whether or not the browser has Java enabled
<u>taintEnabled()</u>	<b>Removed in JavaScript version 1.2.</b> Specifies whether the browser has data tainting enabled

# Objeto Screen

- Contiene información acerca la pantalla del visitante

Property	Description
<u>availHeight</u>	Returns the height of the screen (excluding the Windows Taskbar)
<u>availWidth</u>	Returns the width of the screen (excluding the Windows Taskbar)
<u>colorDepth</u>	Returns the bit depth of the color palette for displaying images
<u>height</u>	Returns the total height of the screen
<u>pixelDepth</u>	Returns the color resolution (in bits per pixel) of the screen
<u>width</u>	Returns the total width of the screen

# Objeto History

- Contiene la historia de los URLs visitados por el usuario

Property	Description
<u>length</u>	Returns the number of URLs in the history list

Method	Description
<u>back()</u>	Loads the previous URL in the history list
<u>forward()</u>	Loads the next URL in the history list
<u>go()</u>	Loads a specific URL from the history list

# Objeto Location

- Contiene información acerca del URL actual.

Property	Description
<u>hash</u>	Sets or returns the anchor part (#) of a URL
<u>host</u>	Sets or returns the hostname and port number of a URL
<u>hostname</u>	Sets or returns the hostname of a URL
<u>href</u>	Sets or returns the entire URL
<u>origin</u>	Returns the protocol, hostname and port number of a URL
<u>pathname</u>	Sets or returns the path name of a URL
<u>port</u>	Sets or returns the port number of a URL
<u>protocol</u>	Sets or returns the protocol of a URL
<u>search</u>	Sets or returns the querystring part of a URL

# Objeto Location

Method	Description
<u>assign()</u>	Loads a new document
<u>reload()</u>	Reloads the current document
<u>replace()</u>	Replaces the current document with a new one



# Ejercicio

- Haz una pagina web que haga lo siguiente:
  - Despliegue el nombre de la ventana actual.
  - Despliegue si el browser es compatible con Java.
  - Despliegue el nombre del Navegador.
  - Despliegue si esta habilitado el uso de cookies.
  - Despliegue ancho y alto de la pantalla.
  - Despliegue el nombre del URL actual.
  - Boton para que se mueva la ventana a una posicion en especifico.
  - Boton para cargar la pagina previa del historial
  - Agrega un campo de texto para capturar un URL, y un boton para abrir ese URL en la pantalla actual.

# PREGUNTAS

neitek\_