

DESARROLLO WEB FULL STACK

Eventos en JavaScript

Introducción a los eventos

DWFS COR



Manejadores de eventos semánticos

Esta técnica consiste en asignar las funciones externas mediante las propiedades DOM de los elementos HTML.

**Es importante saber que el elemento debe existir para seleccionarlo.*

```
<input id="btn" type="button" value="¡Click  
acá!" onclick="console.log('¡Gracias! :~:');" />
```

```
// Se puede transformar en  
function darGracias() {  
    console.log('¡Gracias! :~:');  
}
```

```
document.getElementById('btn').onclick =  
darGracias;
```

```
// o con una función anónima  
document.getElementById('btn').onclick =  
function() {  
    console.log('¡Gracias! :~:');  
}
```



Esperando a que se cargue todo el documento

Asignamos un evento **onload** semánticamente al **window** que es lo mismo que hacer `<body onload="...">`

Si vuelvo a asignar otra función a la misma propiedad se sobrescribe.

```
window.onload = function() {  
    console.log('Página cargada');  
}  
  
function darGracias() {  
    console.log('¡Gracias! :)' );  
}  
  
window.onload = function() {  
    document.getElementById('btn').onclick =  
    darGracias;  
}
```



Flujo de eventos

Además de los eventos básicos que se han visto, los navegadores incluyen un mecanismo relacionado llamado flujo de eventos o "event flow".

Si en una página HTML se define un elemento `<div>` con un botón en su interior, cuando el usuario pulsa sobre el botón, el navegador permite asignar una función de respuesta al botón, otra función de respuesta al `<div>` que lo contiene y otra función de respuesta a la página completa. De esta forma, un solo evento (la pulsación de un botón) provoca la respuesta de tres elementos de la página (incluyendo la propia página).

El orden en el que se ejecutan los eventos asignados a cada elemento de la página es lo que constituye el flujo de eventos. Además, existen muchas diferencias en el flujo de eventos de cada navegador.

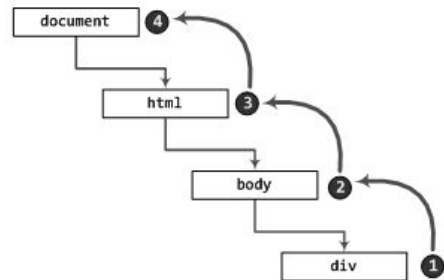


Tipos de flujos de eventos

EVENT BUBBLING:

En este modelo de flujo de eventos, el orden que se sigue es desde el elemento más específico hasta el elemento menos específico.

```
<html onclick="procesaEvento()">
  <head><title>Ejemplo de flujo de eventos</title></head>
  <body onclick="procesaEvento()">
    <div onclick="procesaEvento()">Pincha aqui</div>
  </body>
</html>
```



Tipos de flujos de eventos

EVENT CAPTURING:

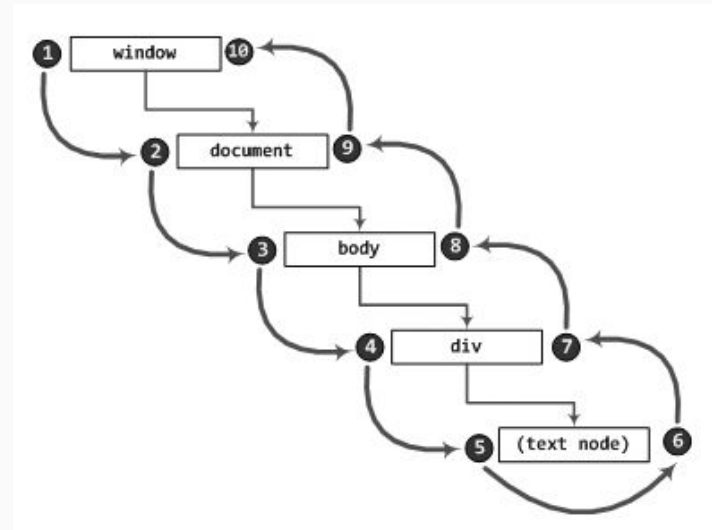
En ese otro modelo, el flujo de eventos se define desde el elemento menos específico hasta el elemento más específico. En otras palabras, el mecanismo definido es justamente el contrario al "event bubbling".



Tipos de flujos de eventos

EVENTOS DOM:

El flujo de eventos definido en la especificación DOM soporta tanto el **bubbling** como el **capturing**, pero el "event capturing" se ejecuta en primer lugar. Los dos flujos de eventos recorren todos los objetos DOM desde el objeto document hasta el elemento más específico y viceversa. Además, la mayoría de navegadores que implementan los estándares, continúan el flujo hasta el objeto window



Manejadores de eventos DOM

La especificación DOM define otros dos métodos denominados **addEventListener()** y **removeEventListener()** para asociar y desasociar manejadores de eventos.

Se requieren tres parámetros: el nombre del "event listener", una referencia a la función encargada de procesar el evento y el tipo de flujo de eventos al que se aplica.

El primer argumento no es el nombre completo del evento, sino que se debe eliminar el prefijo **on**.

Si el tercer parámetro es true, el manejador se emplea en la fase de **capture**. Si el tercer parámetro es false, el manejador se asocia a la fase de **bubbling**.



addEventListener

Agregando un
manejador.

```
function darGracias() {  
    console.log('¡Gracias! :~');  
}  
  
document.getElementById('btn').addEventListener  
( 'click', darGracias, false );
```



addEventListener

Agregar otro
manejador.

```
function darGracias() {  
    console.log('¡Gracias! :~');  
}
```

```
function otraCosa() {  
    console.log('Y daleeee...');  
}
```

```
document.getElementById('btn').addEventListener  
(  
    'click', darGracias, false  
);  
document.getElementById('btn').addEventListener  
(  
    'click', otraCosa, false  
);
```



removeEventListener

Removiendo
manejadores.

```
function darGracias() {  
    console.log('¡Gracias! :~');  
}
```

```
function otraCosa() {  
    console.log('Y daleeee...');  
}
```

```
document.getElementById('btn').removeEventListe  
ner('click', darGracias, false);
```



El objeto “event”

Cuando se produce un evento, no es suficiente con asignarle una función responsable de procesar ese evento. Normalmente, la función que procesa el evento necesita información relativa al evento producido: la tecla que se ha pulsado, la posición del mouse, el elemento que ha producido el evento, etc.

El objeto **event** es el mecanismo definido por los navegadores para proporcionar toda esa información. Se trata de un objeto que se crea automáticamente cuando se produce un evento y que se destruye de forma automática cuando se han ejecutado todas las funciones asignadas al evento.

El **estándar DOM** especifica que el objeto **event** es el único parámetro que se debe pasar a las funciones encargadas de procesar los eventos. Por tanto, en los navegadores que siguen los estándares, se puede acceder al objeto event a través de los argumentos de la función, siempre es el primero.



El objeto “event”

Siempre aparece automáticamente como primer parámetro de la función.

```
function darGracias(event) {  
    console.log(event);  
    console.log(event.target); // this  
}
```

```
document.getElementById('btn').addEventListener  
( 'click', darGracias, false );
```



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>altKey</code>	Boolean	Devuelve <code>true</code> si se ha pulsado la tecla <code>ALT</code> y <code>false</code> en otro caso
<code>bubbles</code>	Boolean	Indica si el evento pertenece al flujo de eventos de bubbling



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>button</code>	Número entero	El botón del mouse que ha sido pulsado. Valores: 0- Ningún botón pulsado 1 - Se ha pulsado el botón izquierdo2- Se ha pulsado el botón derecho3- Se pulsan a la vez el botón izquierdo y el derecho4- Se ha pulsado el botón central5- Se pulsan a la vez el botón izquierdo y el central6- Se pulsan a la vez el botón derecho y el central7' - Se pulsan a la vez los 3 botones



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>cancelable</code>	Boolean	Indica si el evento se puede cancelar
<code>cancelBubble</code>	Boolean	Indica si se ha detenido el flujo de eventos de tipo <i>bubbling</i>



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>charCode</code>	Número entero	El código unicode del carácter correspondiente a la tecla pulsada
<code>clientX</code>	Número entero	Coordenada X de la posición del ratón respecto del área visible de la ventana



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>clientY</code>	Número entero	Coordenada Y de la posición del ratón respecto del área visible de la ventana
<code>ctrlKey</code>	Boolean	Devuelve <code>true</code> si se ha pulsado la tecla <code>CTRL</code> y <code>false</code> en otro caso



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>currentTarget</code>	Element	El elemento que es el objetivo del evento
<code>detail</code>	Número entero	El número de veces que se han pulsado los botones del ratón



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>eventPhase</code>	Número entero	La fase a la que pertenece el evento: 0 – Fase capturing 1 – En el elemento destino 2 – Fase bubbling
<code>isChar</code>	Boolean	Indica si la tecla pulsada corresponde a un carácter



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>keyCode</code>	Número entero	Indica el código numérico de la tecla pulsada
<code>metaKey</code>	Número entero	Devuelve <code>true</code> si se ha pulsado la tecla <code>META</code> y <code>false</code> en otro caso



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>pageX</code>	Número entero	Coordenada X de la posición del ratón respecto de la página
<code>pageY</code>	Número entero	Coordenada Y de la posición del ratón respecto de la página



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>preventDefault()</code>	Función	Se emplea para cancelar la acción predefinida del evento
<code>relatedTarget</code>	Element	El elemento que es el objetivo secundario del evento (relacionado con los eventos de ratón)



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>screenX</code>	Número entero	Coordenada X de la posición del ratón respecto de la pantalla completa
<code>screenY</code>	Número entero	Coordenada Y de la posición del ratón respecto de la pantalla completa



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>shiftKey</code>	Boolean	Devuelve <code>true</code> si se ha pulsado la tecla <code>SHIFT</code> y <code>false</code> en otro caso
<code>stopPropagation()</code>	Función	Se emplea para detener el flujo de eventos de tipo bubbling



¿Qué información contiene el objeto event?

Propiedad/Método	Devuelve	Descripción
<code>target</code>	Element	El elemento que origina el evento
<code>timeStamp</code>	Número	La fecha y hora en la que se ha producido el evento
<code>type</code>	Cadena de texto	El nombre del evento



Práctica



A partir de la página web proporcionada, completar el código JavaScript para que:

1. Cuando se clickee sobre el primer botón, se oculte su sección relacionada.
2. Cuando se vuelva a clicar sobre el mismo botón, se muestre otra vez esa sección de contenidos.
3. Completar el resto de enlaces de la página para que su comportamiento sea idéntico al del primer botón.
4. Cuando una sección se oculte, debe cambiar el mensaje del botón asociado.

