

DISEÑO DE INTERFACES WEB

U.D. 4 (parte 1/2): Multimedia



Multimedia

- Al inicio, Internet solamente era utilizado para el intercambio de texto.

¿Qué más se intercambia ahora?

- Actualmente se intercambia sonido, vídeos o animaciones por ejemplo.

Multimedia

- Cualquier objeto o sistema que utiliza múltiples medios de expresión físicos o digitales electrónicos para presentar o comunicar información. De allí la expresión multimedios.
- Los medios pueden ser variados: [texto](#), [imágenes](#), [animación](#), [sonido](#), [vídeo](#).
- [Multimedia interactiva](#): Cuando el usuario tiene libre control sobre la presentación de los contenidos, acerca de qué es lo que desea ver y cuándo. A diferencia de una presentación directa, en la que es forzado a visualizar contenido en un orden predeterminado.

Propiedad intelectual

- En la Web abunda el material interactivo, gráfico y multimedia, pero su fácil acceso y su disponibilidad no significan que todo ese material sea gratuito ni que se pueda utilizar como si fuera propio.
- En esta unidad se identificará la legislación relacionada con la salvaguarda y el respeto a los derechos de autor. Dicha legislación regula, a nivel nacional e internacional, los derechos que tiene el material que se ofrece y que ofrecemos en la Web, y no supone una cortapisa a su uso sino un reconocimiento de sus legítimos propietarios y de sus derechos.

Propiedad intelectual

- Los **derechos de propiedad intelectual y de autor** son esenciales para proteger la creatividad humana al ofrecer a los autores incentivos en forma de reconocimiento y recompensas económicas equitativas. Este sistema de derechos garantiza a los creadores la divulgación de sus obras sin temor a que se realicen copias no autorizadas o actos de piratería.
- La **propiedad intelectual** es un concepto amplio, que abarca tanto a los derechos de autor como los llamados derechos conexos a los derechos de autor. Es el conjunto de derechos de autor, personales (morales) y patrimoniales (económicos) que corresponden a los autores sobre las obras de su creación.

Propiedad intelectual

- Los derechos que conforman la propiedad intelectual se dividen en dos grupos:
 - **Derechos morales:** Frente a los sistemas de corte anglosajón, la legislación española es claramente defensora de los derechos morales, reconocidos para los autores y para los artistas-intérpretes. Estos derechos son irrenunciables e inalienables, acompañan al autor o al artista durante toda su vida y a sus herederos al fallecimiento de aquellos. Entre ellos destaca el **derecho al reconocimiento de la condición de autor de la obra** o del reconocimiento del nombre del artista sobre sus interpretaciones o ejecuciones, y el de exigir el **respeto a la integridad de la obra** o actuación y la **no alteración de las mismas**.

Propiedad intelectual

- **Derechos de carácter económico:** Derechos relacionados con la **explotación de la obra** o prestación protegida, transferibles y de duración determinada, que a su vez se subdividen en:
 - **Derechos exclusivos:** Permiten a su **titular autorizar o prohibir** los actos de **explotación** de su obra o prestación protegida por el usuario, y a **exigir de este una retribución** a cambio de la autorización que le conceda.
 - **Derechos de remuneración:** Obligan al usuario al pago de una cantidad dineraria por los actos de explotación que realice, **cantidad esta que es determinada**, bien por la **ley** o en su defecto por las **tarifas generales** de las entidades de gestión.
 - **Derechos compensatorios:** Como el derecho por copia privada que compensa los derechos de propiedad intelectual **dejados de percibir** por razón de las reproducciones de las obras o prestaciones protegidas **para uso exclusivamente privado del copista**.

Propiedad intelectual

- La norma nacional principal es el **Real Decreto Legislativo 1/1996, de 12 de abril**, *por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia*. Ha sido objeto de algunas modificaciones posteriores.
- También de consideración es la **Directiva 2001/29/CE del Parlamento Europeo y del Consejo, de 22 de mayo de 2001**, *relativa a la armonización de determinados aspectos de los derechos de autor y derechos afines a los derechos de autor en la sociedad de la información*.

Propiedad intelectual

- Tratado de la Unión de Berna de 1886 (revisado en varias ocasiones), y los Tratados de la **OMPI** (Organización Mundial de la Propiedad Intelectual) de 20 de diciembre de 1996, uno sobre derechos de autor y otro sobre derechos afines.
- En otras jurisdicciones y para la OMPI, la expresión intellectual property engloba tanto los derechos de **propiedad industrial** (marcas, patentes, diseño industrial, denominaciones de origen) como los derechos de **propiedad intelectual** (derechos de autor y derechos afines).

Propiedad intelectual

- Las **obras** que son objeto de propiedad intelectual son todas las creaciones originales literarias, artísticas o científicas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro.
- No puede incluirse en la definición anterior las ideas, la información, y todo conocimiento que es patrimonio común y no es susceptible de apropiación.
- La ley excluye expresamente las disposiciones legales y reglamentarias, sus correspondientes proyectos, las resoluciones de órganos jurisdiccionales, actos, acuerdos, deliberaciones y dictámenes de organismos públicos y traducciones oficiales de todos ellos.

Licencias

- Las preguntas que surgen en función de los derechos de propiedad intelectual y de tus necesidades como creador y autor de sitios web pasarían por ser capaz de responder a preguntas como las siguientes:

¿Dónde encuentro recursos digitales para enriquecer mi trabajo?
¿Puedo utilizarlos libremente? ¿Tengo obligación de citar la fuente u origen de los recursos que utilice? ¿Cómo sé qué tipo de uso puedo hacer de los recursos que he encontrado en la Web? ¿Hay alguna forma de enriquecer con mi trabajo a la comunidad Web? ¿Qué derechos tengo sobre los materiales que produzco?

- Muchas de estas preguntas tienen su respuesta en la licencia con la que se ofrezcan los recursos en la Web.

Licencias

- Una **licencia de software** es un **contrato** entre el:

1. **Licenciante:** Autor / titular de los derechos de explotación / distribución.
2. **Licenciatario:** Usuario consumidor / usuario profesional o empresa.

para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas.

Licencias

- El **copyright** designa únicamente los derechos de explotación de una obra, no hace relación a los derechos morales. El símbolo © asociado a un nombre indica titularidad de derechos de explotación.
- El **copyleft** o “todo libre” es un movimiento social y cultural alternativo al sistema tradicional del copyright que aboga por el uso de licencias libres para compartir y reutilizar las obras de creación.
- **Creative Commons** plantea un paso intermedio entre el férreo sistema de copyright y el “libre total” Copyleft. El autor otorga a la comunidad cierta libertad de uso sobre su obra aunque bajo determinadas condiciones, propone “algunos derechos reservados”.

Licencias

- Tras un proceso de adaptación existe en España desde el 2004 **diferentes modelos posibles de licencias Creative Commons** a elegir, en función de lo que se pretenda. La iconografía utilizada para distinguir estas licencias es:



Reconocimiento: El material creado por un artista puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos.



Reconocimiento – Sin obra derivada: El material creado por un artista puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se pueden realizar obras derivadas.



Reconocimiento – Sin obra derivada – No comercial : El material creado por un artista puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial. No se pueden realizar obras derivadas.



Reconocimiento – No comercial: El material creado por un artista puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial.



Reconocimiento – No comercial – Compartir igual : El material creado por un artista puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.



Licencias

- Algunos sitios donde se pueden descargar imágenes con este tipo de licencia son:

<http://search.creativecommons.org/>

<http://www.pixsy.com/>

<http://www.flickr.com/>

<http://www.everystockphoto.com/>

Imágenes

- Cuando se habla de **formatos de imagen** en la Web, tradicionalmente se han contemplado las imágenes de mapa de bits, dentro de las que se encuentran imágenes que se pueden identificar a través de sus extensiones:
 - **.JPEG** (Joint Photographic Experts Group): Diseñado para la compresión de *imágenes fotográficas*, basándose en el hecho de que el ojo humano no es perfecto y no es capaz de captar toda la información que se puede almacenar una imagen de 24 bits.
 - **.GIF** (Graphic Image File Format): Hace uso de un formato más adecuado para aquellas imágenes *sencillas*, de formas simples y en las que no existe un elevado número de colores.
 - **.PNG** (Portable Network Graphics): Es el más adecuado para imágenes *renderizadas*, es decir que provienen de un modelo, ya que se logran unos degradados más suaves y una buena definición de las líneas. Compresión sin pérdida.

Imágenes

- Los **formatos de gráficos vectoriales** para la Web se pueden dividir en dos tipos: los soportados mediante el plug-in del navegador y los basados en lenguajes de marcado.
 - Los primeros son los gráficos vectoriales clásicos, que necesitan del plug-in correspondiente para poder ser visualizados en un navegador concreto.
 - En las imágenes vectoriales basadas en lenguajes de marcado, los elementos de esas imágenes se definen con texto, que después es interpretado por el navegador, al igual que un archivo HTML.
- Los estándares más populares de formatos de imagen vectorial son:
 - *.eps* (archivo postscript), *.ps*, *.pdf*, *.fla* y *.swf* (definidos por Adobe), *.wmf* (Microsoft), y *svg* (W3C).

Imágenes

- Software para crear y procesar imágenes:
 - **Photoshop**: Creación, edición y retoque de imágenes. Los formatos propios son PSD y PDD, que guardan capas, canales, guías y cualquier modo de color.
 - **GIMP** (Image Manipulation Program): Alternativa al anterior en el ámbito de software libre GNU.
 - **Otros**: Pixlr, Paint, etc.

<https://www.educaciontrespuntocero.com/recursos/programas-gratis-para-editar-imagenes/>

Imágenes

- Optimización de imágenes:
 - El tamaño de un archivo gráfico viene determinado, entre otros, por los siguientes factores:
 - Dimensiones de la imagen
 - Profundidad o paleta de colores
 - Resolución
 - Formato de archivo (JPG, GIF, PNG, etc.)

Imágenes

- Recomendaciones de Optimización:
 - Conviene definir una resolución de imagen no superior a 96 ppp. Es la que usan las pantallas de ordenador:
<https://ayuda.arcadina.com/es/articles/4965254-como-dar-un-tamano-y-resolucion-optimo-a-nuestras-imagenes-para-web>
 - Reducir el número de colores de la paleta porque ello supone disminuir el tamaño del archivo.
 - Conviene utilizar un programa de tratamiento de imágenes para definir las dimensiones concretas de una imagen antes de insertarla en una página Web.

<https://kinsta.com/es/blog/optimizar-imagenes-para-la-web/>

<https://blog.hubspot.es/marketing/guia-para-el-uso-y-optimizacion-de-imagenes>

Imágenes

- Recomendaciones de Optimización:
 - Guardar los originales de las imágenes favoritas en formato BMP, TIFF ó JPEG sin comprimir. A partir de ellas se puede crear una copia en formato GIF (PNG) o JPEG con las dimensiones, resolución y paletas optimizadas para publicarlas.
 - Las imágenes GIF son más adecuadas para dibujos, gráficos y logotipos. Son aquellas donde predominan los colores sólidos y una paleta con un número reducido de colores.
 - Las imágenes JPEG se adaptan mejor a fotografías e imágenes con degradados complejos. Admiten color de 24 bits y gracias a su compresión ofrecen una imagen más brillante que ocupa menos espacio.

Imágenes

- Herramientas software de Optimización:

- Photoshop, GIMP, Pixlr, etc.

- **Otras:**

- <https://www.media.io/es/image-upscaler-tips/photo-enhancer-tools.html>

- <https://www.whatsnew.com/2017/03/18/6-herramientas-online-para-reducir-y-optimizar-imagenes-en-pocos-segundos/>

Imágenes

- Una **imagen** se incluye en una página Web con la etiqueta ``, independientemente de su tipo:

```

```

- Permite incorporarla en el contenido que se va a representar.
- El atributo `src` sirve para indicar la URL donde se encuentra la imagen, `alt` permite definir un texto alternativo a la imagen que se mostrará en navegadores sin soporte gráfico, `width` y `height` para representarla con un determinado tamaño.
- No tiene etiqueta de cierre.
- `<figure>` y `<figcaption>`: Pies de figura.

Imágenes

- Posicionar notas sobre una imagen:
 - Para **mostrar** información adicional y **comentarios sobre diferentes zonas de una imagen**.
 - Es posible definir diferentes zonas “pinchables” dentro de una misma imagen.
 - Ejemplo: <https://uniwebsidad.com/libros/css-avanzado/capitulo-6/mapas-de-imagen?from=librosweb>

Imágenes

■ Sprites CSS:

- Consiste en **agrupar todas las imágenes/iconos de la Web en una sola**. De esta forma, en una sola petición, se obtienen todas las imágenes. Resulta más óptimo hacer una petición de una imagen un poco más pesada, que una por imagen.
- Se crea una imagen grande que incluya todas las individuales y se va desplazando la imagen de fondo.
- Ejemplo: <https://uniwebsidad.com/libros/css-avanzado/capitulo-1/rollovers-y-sprites?from=librosweb>
- Base de datos de Sprites CSS: <https://spritedatabase.net/>
- Generador de Sprites CSS: <https://spritegen.website-performance.org/>

Imágenes

- Mapas de imagen:
 - Es una lista de coordenadas relacionadas (mapa poligonal) de una imagen para hipervincular áreas de la misma a diferentes destinos (a diferencia de un enlace de imagen normal, en el que toda el área de la misma se vincula a un solo destino). Vincula varias partes de una imagen sin dividirla en archivos separados.
 - Combinando las etiquetas `<map>` y `<area>` junto con el atributo `usemap` de la etiqueta `` también es posible definir zonas “pinchables” dentro de la imagen.
 - Ejemplo: <https://uniwebsidad.com/libros/xhtml/capitulo-6/mapas-de-imagen>

Ejercicios

- Realiza todos los ejercicios del bloque 'Ejercicios 4.1'.

Animaciones

■ Con **GIF**:

- Formato creado en 1987 por Steve Wilhite, también compatible con animaciones sencillas.
- Permite comprimir imágenes sin perder información. Se pueden guardar varias imágenes independientes superpuestas.
- La **animación** consiste en una serie de imágenes que se muestran de forma consecutiva durante un intervalo de tiempo. Esta secuencia se hace en modo bucle para se repita indefinidamente generando un pequeño vídeo sin sonido.
- **Herramientas de creación:** Picasion, GIMP, GIF Construction Set Professional, Photoshop, etc.

Animaciones

■ Con **Sprites CSS**:

- La regla *@keyframes* es el núcleo de la definición de la animación en CSS. Dentro de ella se especifican el estilo que se aplicará a la evolución de la animación. Sus cotas *from...to* marcan los estados inicial y final de la animación, representa el valor inicial 0% y el final 100%, interpolándose automáticamente los valores intermedios.
- El atributo *animation* define las propiedades que tendrá la animación en sí, al que se le pasan parámetros:
`animation: name duration timing-function delay iteration-count direction;`
- Ejemplo: <https://desarrolloweb.com/articulos/animaciones-sprites-css3-javascript.html>

Ejercicios

- Realiza todos los ejercicios del bloque 'Ejercicios 4.3'.

Códec

- Es el orden utilizado para diseñar los datos de un archivo de audio o vídeo de tal manera que se pueda utilizar para reproducir, editar o cambiar a otros códecs (transcodificación).
- Se utilizan para organizar los datos de los medios, pero esos datos se mantienen dentro de un contenedor. Hay muchos tipos diferentes, cada uno con sus ventajas e inconvenientes.
- Se utiliza para comprimir y luego descomprimir un archivo de audio o vídeo. Esta compresión puede ser con pérdida o sin pérdida.
- No todos los códecs son compatibles con todos los reproductores audio o vídeo.

Audio

- Para una correcta utilización de este tipo de archivos era necesario *plug-ins* en los navegadores que ampliaban sus capacidades.
- HTML 5 ya incluye soporte para reproducción de audio y vídeo sin necesidad de *plug-ins* (como Adobe Flash, bloqueado por seguridad desde 2021), lo que evita fuentes de errores y cuelgues del navegador. Incluye etiquetas especiales de audio y vídeo que no necesita de un complemento especial para su ejecución.

Audio

- **Tipos de formatos:**

- Sin compresión: WAV, PCM, AIFF, AU.
- Comprimido sin pérdidas: MPEG-4 A, FLAC, AIFF-C, SHN, TTA, ATRAC, ALAC.
- Comprimido con pérdidas: MP3, OGG Vorbis, AAC, WMA, Musepack, Opus, DSD, MQA.

<https://protecciondatos-lopd.com/empresas/tipos-formato-audio/>

- **Conversores de formatos:**

<https://recoverit.wondershare.es/audio-recovery/top-10-audio-converter.html>

Audio

■ Inserción:

<audio>: Reproduce audio tanto de archivos de audio como de vídeo.

Evento	Descripción
autoplay	Si está presente, la reproducción es automática e inmediata
preload	Precarga el audio (junto con la página aunque no se haya pulsado <i>play</i>) Valores: <ul style="list-style-type: none">▪ none: sin precarga▪ metadata: solo se precarga metadatos▪ auto: petición de precarga, pero el navegador decide
controls	Se muestran los controles
loop	Se vuelve a reproducir el audio cuando finaliza
muted	Si está presente, inicialmente el audio está silenciado
src	URL del audio

Audio

- El reproductor de audio puede modificarse con las propiedades globales de la etiqueta <audio>.
- Por ejemplo: Cambiar el color de fondo.

```
<audio controls style="background-color:#00FF00">
```

```
<source src="audio.mp3" type="audio/mp3">
```

```
<source src="audio.ogg" type="audio/ogg">
```

```
<source src="audio.wav" type="audio/mpeg">
```

```
El audio no puede ser reproducido
```

```
</audio>
```

Vídeo

- **Tipos de formatos:**

- MPEG-4, AVI, MKV, FLV, MOV, WMV, DIVX, H.264, XVID, RM, 3GP, WebM.

<https://protecciondatos-lopd.com/empresas/tipos-formato-video/>

<https://aulacm.com/formatos-video-xvid-mp4-h264/>

Formato	Códec de vídeo	Códec de audio	Tipo MIME
WebM	VP8	Vorbis	video/webm
Ogg	Theora	Vorbis	video/ogg
MP4	H.264	AAC	video/mp4

- **Conversores de formatos:**

<https://uniconverter.wondershare.es/video-converter/free-video-converter-windows-10.html>

Vídeo

■ Inserción:

<video>: Reproduce vídeo de archivos de vídeo, también reproduce audio de archivos de audio.

Evento	Descripción
autoplay	Si está presente, la reproducción es automática e inmediata
preload	Precarga el vídeo (junto con la página aunque no se haya pulsado <i>play</i>) Valores: <ul style="list-style-type: none">▪ none: sin precarga▪ metadata: solo se precarga metadatos (anchura, altura, primer <i>frame</i>, ...)▪ auto: petición de precarga, pero el navegador decide
controls	Se muestran los controles
loop	Se vuelve a reproducir el vídeo cuando finaliza
poster	Imagen estática mientras el vídeo se descarga Por ejemplo: poster="imagen.jpg"
height	Altura del reproductor de vídeo
width	Anchura del reproductor de vídeo
muted	Si está presente, inicialmente el vídeo está silenciado
src	URL del vídeo

Vídeo

- El reproductor de vídeo puede modificarse con las propiedades globales de la etiqueta <video>.
- Por ejemplo: Cambiar el color de fondo.

```
<video width="640" height="480" controls  
  style="background-color:#00FF00">
```

```
<source src="vídeo.mp4" type="video/mp4">
```

El vídeo no puede ser reproducido

```
</video>
```

Vídeo

- Otras etiquetas multimedia:

<source>: Define los recursos multimedia para los elementos multimedia, como las fuentes de audio y vídeo. Se pueden indicar varias fuentes y el reproductor reproducirá la primera que pueda cargar.

<track>: Se utiliza para indicar subtítulos o textos relacionados con el contenido reproductivo (capítulos, etc.) que serán visibles mientras el vídeo se reproduce.

```
<video width="320" height="240" controls>
  <source src= "vídeo.mp4" type="video/mp4">
  <source src= "vídeo.ogg" type="video/ogg">
  <track src= "subtítulos_en.vtt" kind="subtitles" srclang="en" label="English">
  <track src= "subtítulos_es.vtt" kind="subtitles" srclang="es" label="Español">
</video>
```

Vídeo

■ Subtítulos:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Video y audio</title>
  </head>
  <body>
    <video width="720" controls>
      <source src="agent327.mp4" type="video/mp4">
      <source src="agent327.ogg" type="video/ogg">
      <track src="agent327.en.srt" kind="subtitles" srclang="en" label="English">
      <track src="agent327.es.srt" kind="subtitles" srclang="es" label="Español">
      Video cannot be displayed
    </video>
  </body>
</html>
```



Ejercicios

- Realiza todos los ejercicios del bloque 'Ejercicios 4.4'.

DISEÑO DE INTERFACES WEB

**U.D. 4 (parte 2/2):
Interactividad**



Eventos con CSS

- Los eventos son una característica de los documentos HTML que permite agregar **interactividad** entre el sitio Web y el usuario, al ejecutar programas en el lado cliente cuando el usuario (u otro programa) realiza una acción. Por ejemplo, se puede hacer que un párrafo cambie el color de su texto cuando el usuario pase el puntero del ratón sobre él.
- Eventos CSS con selectores por pseudoclase dinámica: <https://www.wextensible.com/como-se-hace/pestanyas-tabs/referencias-eventos-css.html>

Eventos con JavaScript

- La ejecución de acciones asociadas a eventos en el navegador se realiza principalmente con JavaScript, por lo que esta parte se imparte en el módulo “Desarrollo Web en Entorno Cliente”.

Modelo de gestión de eventos

- Los eventos son mecanismos que se accionan cuando el usuario realiza un cambio sobre una página Web.
- El encargado de crear la jerarquía de objetos que compone una página Web es el DOM (*Document Object Model*).
- Por tanto, es el DOM el encargado de gestionar los eventos.

Modelo de gestión de eventos

- Para poder controlar un evento se necesita un **manejador**.
- El manejador es la palabra reservada que indica la acción que va a manejar.
- En el caso del evento *click*, el manejador sería `onClick`. Ejemplo:

```

```

Modelo de gestión de eventos

- El ejemplo anterior se puede realizar de otro modo llamando a una función:

```
<html>
  <head>
    <title>Página de Evento del ratón</title>
    <script>
      function funcion() {
        alert("Click en imagen");
      }
    </script>
  </head>
  <body>
    
  </body>
</html>
```

Modelo de gestión de eventos

- La especificación DOM define cuatro grupos de eventos dividiéndolos según su origen:
 - Eventos del ratón.
 - Eventos del teclado.
 - Eventos HTML.
 - Eventos DOM.

Modelo de gestión de eventos

- Eventos del ratón (1):
 - **Click:** Este evento se produce cuando se pulsa sobre el botón izquierdo del ratón. El manejador de este evento es `onclick`.
 - **Dblclick:** Este evento se acciona cuando se hace un doble click sobre el botón izquierdo del ratón. El manejador de este evento es `ondblclick`.
 - **Mousedown:** Este evento se produce cuando se pulsa un botón del ratón. El manejador de este evento es `onmousedown`.
 - **Mouseout:** Este evento se produce cuando el puntero del ratón está dentro de un elemento y este puntero es desplazado fuera del elemento. El manejador de este evento es `onmouseout`.

Modelo de gestión de eventos

- Eventos del ratón (2):
 - **Mouseover:** Este evento al revés que el anterior se produce cuando el puntero del ratón se encuentra fuera de un elemento, y este se desplaza hacia el interior. El manejador de este evento es `onmouseover`.
 - **Mouseup:** Este evento se produce cuando se suelta un botón del ratón que previamente se tenía pulsado. El manejador de este evento es `onmouseup`.
 - **Mousemove:** Se produce cuando el puntero del ratón se encuentra dentro de un elemento. Es importante señalar que este evento se producirá continuamente una vez tras otra mientras el puntero del ratón permanezca dentro del elemento. El manejador de este evento es `onmousemove`.

Modelo de gestión de eventos

- Eventos del teclado:
 - **Keydown:** Este evento se produce cuando se pulsa una tecla del teclado. Si se mantiene pulsada una tecla de forma continua, el evento se produce una y otra vez hasta que se suelte la misma. El manejador de este evento es `onkeydown`.
 - **Keypress:** Este evento se produce si se pulsa una tecla de un carácter alfanumérico (el evento no se produce si se pulsa enter, la barra espaciadora, etc...). En el caso de mantener una tecla pulsada, el evento se produce de forma continuada. El manejador de este evento es `onkeypress`.
 - **KeyUp:** Este evento se produce cuando se suelta una tecla. El manejador de este evento es `onkeyup`.

Modelo de gestión de eventos

- Eventos HTML (1):
 - **Load:** El evento *load* hace referencia a la carga de distintas partes de la página. Este se produce en el objeto `Window` cuando la página se ha cargado por completo. En el elemento `` actúa cuando la imagen se ha cargado. En el elemento `<object>` se acciona al cargar el objeto completo. El manejador es `onload`.
 - **Unload:** El evento *unload* actúa sobre el objeto `Window` cuando la pagina ha desaparecido por completo (por ejemplo, si se pulsa el aspa cerrando la ventana del navegador). También se acciona en el elemento `<object>` cuando desaparece el objeto. El manejador es `onunload`.
 - **Abort:** Este evento se produce cuando el usuario detiene la descarga de un elemento antes de que haya terminado, actúa sobre un elemento `<object>`. El manejador es `onabort`.

Modelo de gestión de eventos

- Eventos HTML (2):
 - **Error:** El evento *error* se produce en el objeto `Window` cuando se ha producido un error en JavaScript. En el elemento `` cuando la imagen no se ha podido cargar por completo y en el elemento `<object>` en el caso de que un elemento no se haya cargado correctamente. El manejador es `onerror`.
 - **Select:** Se acciona cuando se selecciona texto de los cuadros de textos `<input>` y `<textarea>`. El manejador es `onselect`.
 - **Change:** Este evento se produce cuando los cuadros de texto `<input>` y `<textarea>` pierden el foco y el contenido que tenían ha variado. También se producen cuando un elemento `<select>` cambia de valor. El manejador es `onchange`.
 - **Submit:** Este evento se produce cuando se pulsa sobre un botón de tipo submit. El manejador es `onsubmit`.

Modelo de gestión de eventos

- Eventos HTML (3):
 - **Reset:** Este evento se produce cuando se pulsa sobre un botón de tipo reset. El manejador es `onreset`.
 - **Resize:** Este evento se produce cuando se redimensiona el navegador, actúa sobre el objeto `Window`. El manejador es `onresize`.
 - **Scroll:** Se produce cuando varía la posición de la barra de scroll en cualquier elemento que la tenga. El manejador es `onscroll`.
 - **Focus:** Este evento se produce cuando un elemento obtiene el foco. El manejador es `onfocus`.
 - **Blur:** Este evento se produce cuando un elemento pierde el foco. El manejador es `onblur`.

Modelo de gestión de eventos

- Ejercicio: Crea un script donde al pulsar con el botón izquierdo del ratón sobre una imagen, aparezca una ventana emergente con un texto.
- Ejercicio: Crea un script donde al pulsar una tecla dentro de un campo de entrada, aparezca una ventana emergente con un texto.
- Ejercicio: Crea tres campos de entrada, cuando el último obtenga el foco, debe aparecer una ventana emergente con un texto.

Modelo de gestión de eventos

■ Eventos DOM:

- **DOMSubtreeModified:** Este evento se produce cuando se añaden o eliminan nodos en el subárbol de un elemento o documento.
- **DOMNodeInserted:** Este evento se produce cuando se añade un nodo hijo a un nodo padre.
- **DOMNodeRemoved:** Este evento se produce cuando se elimina un nodo que tiene nodo padre.
- **DOMNodeRemovedFromDocument:** Este evento se produce cuando se elimina un nodo del documento.
- **DOMNodeInsertedIntoDocument:** Este evento se produce cuando se añade un nodo al documento.

Modelo de gestión de eventos

- Objeto **window.event**:

- Normalmente, los manejadores de eventos requieren información adicional para procesar sus tareas. Por ejemplo, si una función se encarga de procesar el evento *onclick*, quizás necesite saber en qué posición estaba el ratón en el momento de pinchar el botón. El caso más habitual en el que es necesario conocer información adicional sobre el evento es el de los eventos asociados al teclado para conocer la tecla que se ha pulsado.
- JS permite obtener información sobre el ratón y el teclado mediante un objeto especial llamado **event**. Desafortunadamente, algunos navegadores presentan diferencias en el tratamiento de la información sobre los eventos, sobre todo los relacionados con el teclado.

Modelo de gestión de eventos

- Variable **this**:
 - JavaScript define una variable especial llamada *this* que se crea automáticamente y se emplea en algunas técnicas avanzadas de programación.
 - Por ejemplo, en los eventos, se puede utilizar la variable *this* para referirse al elemento HTML que ha provocado el evento.
 - De esta forma, si a una función (manejador de eventos) se le pasa el parámetro *this*, es posible acceder de forma directa desde dicha función a las propiedades del elemento que ha provocado el evento.

Programación de eventos

- Carga de la página HTML:
 - Una condición para que se genera la estructura de árbol es que la página se cargue completamente.
 - Por este motivo es necesario conocer si se ha cargado y para ello se utiliza el manejador de evento `onload`.

```
<html>
  <head><title>Página HTML cargada</title></head>
  <body onload="alert('Página cargada completamente');">
    <p>Primer párrafo</p>
  </body>
</html>
```

Programación de eventos

- Comprobar si el árbol DOM está cargado:

```
<html>
  <head><title>Árbol DOM cargado</title>
    <script>
      function cargada() {
        window.onload = "true";
        if (window.onload) { return true; }
        return false;
      }
      function pulsar() {
        if (cargada()) { alert("Página cargada correctamente"); }
      }
    </script>
  </head>
  <body> <p onclick="pulsar();">Primer párrafo</p> </body>
</html>
```

Programación de eventos

- Actuar sobre el DOM al desencadenarse eventos:

```
<html>
  <head><title>DOM con evento</title>
  <script>
    function ratonEncima() {
      document.getElementsByTagName("div")[0].
        childNodes[0].nodeValue = "EL RATÓN ESTÁ ENCIMA";
    }
    function ratonFuera() {
      document.getElementsByTagName("div")[0].
        childNodes[0].nodeValue = "NO ESTÁ EL RATÓN ENCIMA";
    }
  </script>
</head>
<body>
  <div onmouseover="ratonEncima();" onmouseout="ratonFuera();">
    VALOR POR DEFECTO</div>
</body>
</html>
```

Gestor de eventos

- En JavaScript es posible añadir un gestor de eventos a un elemento especificado
- **addEventListener()**: permite asociar una función *callback* a un evento que se produzca sobre el elemento

```
elemento.addEventListener("evento", funcion);
```

```
var b1 = document.getElementById("button1");  
b1.addEventListener("click", pulsarBoton);
```

- **removeEventListener()**: se elimina un EventListener ya creado

```
elemento.removeEventListener("evento", funcion);
```

- Se pueden asociar EventListeners a cualquier objeto del DOM (ya sean elementos HTML, el objeto ventana, el documento...)

```
document.addEventListener("DOMContentLoaded", load);
```

Gestor de eventos

- Un elemento puede tener varios EventListeners asociados
 - Puede tener varios EventListeners asociados al mismo evento

```
<body>
  <button id="button1" onclick="funcion1()">Botón</button>
  <script>
    var b1 = document.getElementById("button1");
    b1.addEventListener("click", funcion2);
    b1.addEventListener("click", funcion3);
    b1.addEventListener("mouseover", funcion4);
    b1.addEventListener("click", funcion5);
    b1.removeEventListener("click", funcion2);

    function funcion1(){
      alert("Dentro de funcion1()");
    }

    function funcion2(){
      console.log("Dentro de funcion2()");
    }

    function funcion3(){
      console.log("Dentro de funcion3()");
    }

    function funcion4(){
      console.log("Dentro de funcion4()");
    }

    function funcion5(){
      console.log("Dentro de funcion5()");
    }
  </script>
</body>
```