UD 1_2: RECORDANDO HTML 5

Bajo las siglas HTML5 se agrupan varias tecnologías de desarrollo web:

- HTML5: para contenido
- CSS: para dar estilo al contenido
- JavaScript: para añadir funcionalidad

ESTRUCTURA DE LA PÁGINA WEB

En HTML5 se pretende separar el contenido de su forma de visualización para lo cual hay una serie de etiquetas semánticas que nos permiten estructurar el contenido y que hay que utilizarlas adecuadamente (header, footer, aside…)

<!DOCTYPE>

En primer lugar, necesitamos indicar el tipo de documento que estamos creando. Esto en HTML5 es extremadamente sencillo:

```
<!doctype html>
```

<HTML>

Después de declarar el tipo de documento, debemos comenzar a construir la estructura HTML. Como siempre, la estructura tipo árbol de este lenguaje tiene su raíz en el elemento *<html>.* Este elemento envolverá al resto del código:

```
<!doctype html>
<html lang="es">
…
</html>
```

El atributo *lang* en la etiqueta de apertura *<html>* es el único atributo que necesitamos especificar en HTML5. Este atributo define el idioma del contenido del documento que estamos creando.

El código HTML insertado entre las etiquetas *<html>* tiene que ser dividido entre dos secciones principales. La primera sección es la cabecera y la segunda el cuerpo.

<HEAD>

Dentro de la etiqueta *<head>* definiremos el título de nuestra página web, declararemos el set de caracteres correspondiente, proveeremos información general acerca del documento e incorporaremos los archivos externos con estilos y código Javascript.

Excepto por el título la información incorporada en el documento entre estas etiquetas es invisible para el usuario.

<META>

La etiqueta *<meta>* se utiliza para definir los metadatos de nuestra página web

Mediante la etiqueta *<meta charset="">* especificamos el juego de caracteres que se va usar en nuestra página web:

```
<!doctype html>
<html lang="es">
    <head>
        <meta charset="UTF-8">
    </head>
    <body>
    </body>
</html>
```

Tienes más información sobre los juegos de caracteres en http://www.w3schools.com/html/html_charset.asp

Por supuesto, podemos agregar otras etiquetas *<meta>* como *description o keywords* para definir otros aspectos de la página web y que mejoran el posicionamiento de nuestra página web:

```
<!doctype html>
<html lang="es">
    <head>
        <meta charset=" UTF-8">
        <meta name="description" content="Ejemplo de HTML5">
        <meta name="keywords" content="HTML5, CSS3, Javascript">
    </head>
    <body>
    </body>
</html>
```

En el código, el atributo *name* dentro de la etiqueta *<meta>* especifica su tipo y *content* declara su valor, pero ninguno de estos valores es mostrado en pantalla.

<TITLE>

La etiqueta *<title>*, como siempre, simplemente especifica el título del documento. Normalmente este texto es mostrado en la barra superior de la ventana del navegador y es importante para el posicionamiento de la página.

<LINK>

Otro importante elemento que va dentro de la cabecera del documento es *<link>*. Este elemento es usado para incorporar estilos y código Javascript.

```
<!doctype html>
<html lang="es">
    <head>
        <meta charset="iso-8859-1">
        <meta name="description" content="Ejemplo de HTML5"/>
```

```
        <meta name="keywords" content="HTML5, CSS3, JavaScript"/>
        <title>Este texto es el título del documento</title>
        <link rel="stylesheet" href="misestilos.css"/>
    </head>
    <body>
    </body>
</html>
```

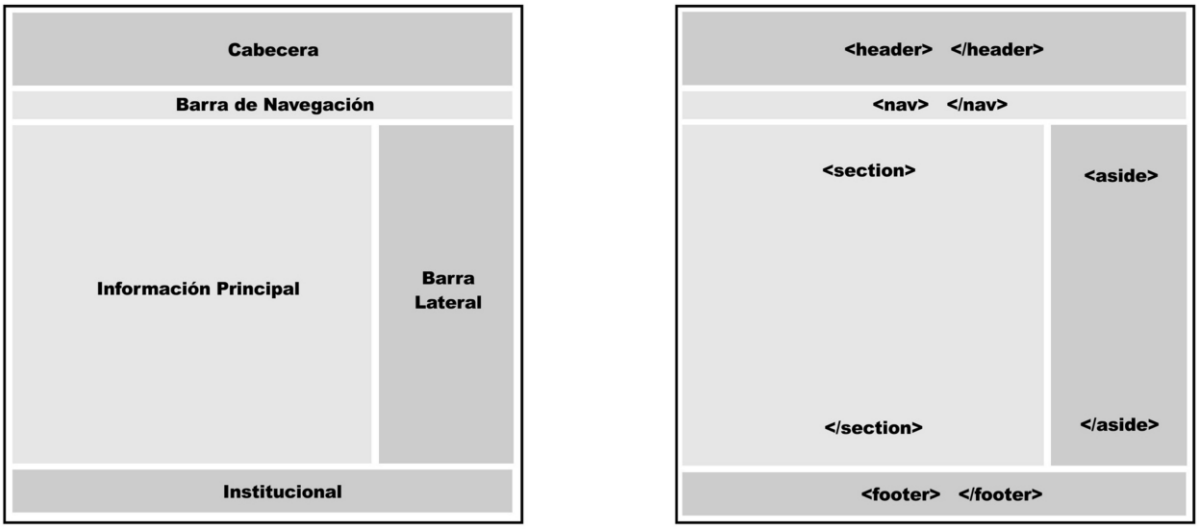
Solo necesitamos dos atributos para incorporar nuestro archivo de estilos: *rel* y *href*. El atributo *rel* significa "relación" y es acerca de la relación entre el documento y el archivo que estamos incorporando por medio de *href*. En este caso, el atributo *rel* tiene el valor *stylesheet* que le dice al navegador que el archivo es un archivo CSS con los estilos requeridos para presentar la página en pantalla.

<BODY>

La estructura del cuerpo (el código entre las etiquetas *<body>*) generará la parte visible del documento. Este es el código que producirá nuestra página web.

HTML5 incorpora elementos que ayudan a identificar cada sección del documento y organizar el cuerpo del mismo: *<header>*, *<nav>*, *<section>*, *<aside>*, *<main>*, *<footer>* y *<article>.* El nombre de estas etiquetas muestra claramente con qué tipo de sección están relacionadas. Además de estas secciones podremos añadir secciones más genéricas con la etiqueta * .*

La siguiente figura representa un diseño común encontrado en muchos sitios webs estos días.



Más información: http://www.mclibre.org/consultar/htmlcss/html/html-secciones.html

<HEADER>

El elemento *<header>* no debe ser confundido con *<head>* usado antes para construir la cabecera del documento. La intención de *<header>* es proveer información introductoria (títulos, subtítulos, logos), para el cuerpo o secciones específicas dentro del cuerpo:

<NAV>

La etiqueta *<nav>* se utiliza para los menús de navegación

<MAIN>

La etiqueta *<main>* se utiliza para indicar el comienzo y el final del área principal de la página web. El área principal del contenido consiste en el contenido que está directamente relacionado con el tema central de la página web. Este contenido debe ser único al documento, excluyendo cualquier contenido que se repita a través de un conjunto de documentos como barras laterales, enlaces de navegación, información de derechos de autor, logos del sitio y formularios de búsqueda (a menos, claro, que la función principal del documento sea un formulario de búsqueda). Evidentemente, la etiqueta *<main>* debe ser única en la página web.

<SECTION>

La información dentro de las diferentes partes del cuerpo (main, aside…) puede aparecer dividida en varias secciones. Para marcar estas secciones tenemos la etiqueta *<section>*

<ARTICLE>

A su vez las secciones se pueden dividir en diferentes artículos mediante la etiqueta *<article>*.

<ASIDE>

Esta etiqueta se utiliza para contener datos relacionados con la información principal pero que no son relevantes o igual de importantes.

Aunque normalmente se identifica con las barras laterales este elemento puede estar ubicado en cualquier parte del diseño y ser usado siempre y cuando su contenido no sea considerado como el contenido principal del documento.

<FOOTER>

Para el indicar el final de la página web o pie de página, tenemos la etiqueta *<footer>*.

Esta parte de la página web es normalmente usada para compartir información general sobre el autor o la organización detrás del proyecto.

Generalmente, el elemento *<footer>* representará el final del cuerpo de nuestro documento y tendrá el propósito descrito anteriormente. Sin embargo, *<footer>* puede ser usado múltiples veces dentro del cuerpo para representar también el final de diferentes secciones (del mismo modo que la etiqueta *<header>*).

EJEMPLO

```
<!doctype html>
<html lang="es">
    <head>
    <meta charset="iso-8859-1"/>
    <meta name="description" content="Ejemplo de HTML5"/>
```

```
    <meta name="keywords" content="HTML5, CSS3, JavaScript"/>
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css"/>
    </head>
    <body>
        <header>
            <h1>Este es el título principal del sitio web</h1>
        </header>
        <nav>
            <ul>
                <li>principal</li>
                <li>fotos</li>
                <li>videos</li>
                <li>contacto</li>
            </ul>
        </nav>
        <section>
            <article>
                Este es el texto de mi primer mensaje
            </article>
            <article>
                Este es el texto de mi segundo mensaje
            </article>
        </section>
        <aside>
            <blockquote>Mensaje número uno</blockquote>
            <blockquote>Mensaje número dos</blockquote>
        </aside>
        <footer>
            Derechos Reservados &copy; 2010-2011
        </footer>
    </body>
</html>
```

FORMATOS DE TEXTOS

Las etiquetas que tenemos para formatear los textos se deben utilizar siempre que aporten significado semántico, si lo que se pretende es dar estilo se debería hacer mediante CSS:

<H1>, <H2>, <H3>, <H4>, <H5> Y <H6>

Existen seis niveles de encabezados, que podemos definir con las etiquetas semánticas *<h1>, <h2>, <h3>, <h4>, <h5> y <h6>*, siendo *<h1>* el encabezado más importante de nuestra página web y el h6 el de menor importancia.

Se debe empezar por el encabezado *<h1>* y este debería ser único en la página. Hemos de anidarlos de forma correcta (y sin saltos), es decir después de un encabezado de segundo nivel puede ir uno de tercer nivel u otro de segundo nivel, pero no uno de cuarto nivel.

<P>

La etiqueta *<p>* se utiliza para marcar los párrafos.

<SPAN>

La etiqueta ** se utiliza para textos en línea, por ejemplo para marcar parte de un párrafo con el fin de poderle dar un aspecto diferenciado mediante CSS o añadirle alguna funcionalidad mediante JavaScript.

<STRONG>

La etiqueta ** se utiliza para dar énfasis (negrita) a un texto.

<EM>

La etiqueta *<en>* se utiliza para dar énfasis (cursiva) a un texto.

<MARK>

La etiqueta *<mark>* se utiliza para marcar un texto.

<DEL>

La etiqueta *<del>* se utiliza para mostrar el texto como tachado

<SMALL>

La etiqueta *<small>* se utiliza para la “letra pequeña”

La etiqueta *
* se utiliza para hacer un salto de página, pero no se debería usar, los saltos de página se deberían controlar mediante los estilos.

<HR>

La etiqueta *<hr>* crea una línea horizontal, a modo de separación

ENLACES

La etiqueta *<a>*, sirve para definir enlaces a otros sitios web o recursos y también para enlazar otra parte dentro de la misma página web.

Por defecto los navegadores suelen visualizar los enlaces con una serie de colores predefinidos, aunque estos se pueden cambiar. Los enlaces no visitados se visualizan normalmente con el color azul y los enlaces visitados con el color purpura.

La sintaxis para esta etiqueta con todos sus atributos es la siguiente:

Texto enlace

Entre los atributos a destacar que tiene la etiqueta se encuentran los siguientes:

- *href*: este atributo permite establecer la URL a la que se quiere acceder al pinchar sobre el enlace. Se puede especificar una URL absoluta, relativa o un ancla (enlace en el mismo documento).
  Si se realiza un enlace en el mismo documento a un objeto concreto se ha de utilizar como identificador del objeto el nombre del mismo que se haya proporcionado mediante el atributo *id*.
- *type*: sirve para especificar el tipo MIME del documento con el que se está realizando el enlace. Se pueden consultar los tipos mime en la siguiente dirección http://www.htmlquick.com/es/reference/mime-types.html
- *target*: el atributo permite especificar el destino por defecto para todos los enlaces y formularios del documento web.
  Entre los posibles valores que se le pueden aplicar al atributo target, encontramos los siguientes:
  - *_blank*: especifica que el destino sea una nueva ventana del navegador, o una nueva pestaña.
  - *_self*: es el valor por defecto. Abre el destino en la misma pestaña donde se haya realizado el clic.
- *download*: este es un atributo booleano que indica al navegador que debe descargar el archivo en lugar de leerlo.

Ejemplo:

```
<!doctype html>
<html>
   <head>
      <title>Ejemplos Etiqueta a</title>
      <meta charset="utf-8">
      <meta name="author" content="Jose Saez">
      <meta name="description" content="Ejercicios y prácticas">
   </head>
   <body>
      <p>Ejercicios de programación HTML5</p>
      <ul>
```

```
        <li><a href="#parrafo1">Enlace Al Primer Párrafo</a></li>
        <li><a href="#parrafo2">Enlace Al Segundo Párrafo</a></li>
        <li><a href="#parrafo3">Enlace Al Tercero Párrafo</a></li>
    </ul>
    <p id="parrafo1">Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno estándar de las industrias desde el año 1500, cuando un impresor (N. del T. persona que se dedica a la imprenta) desconocido usó una galería de textos y los mezcló de tal manera que logró hacer un libro de textos espécimen. No sólo sobrevivió 500 años, sino que también ingresó como texto de relleno en documentos electrónicos, quedando esencialmente igual al original. Fue popularizado en los 60s con la creación de las hojas "Letraset", las cuales contenían pasajes de Lorem Ipsum, y más recientemente con software de autoedición, como por ejemplo Aldus PageMaker, el cual incluye versiones de Lorem Ipsum.</p>
    <p id="parrafo2">Es un hecho establecido hace demasiado tiempo que un lector se distraerá con el contenido del texto de un sitio mientras que mira su diseño. El punto de usar Lorem Ipsum es que tiene una distribución más o menos normal de las letras, al contrario de usar textos como por ejemplo "Contenido aquí, contenido aquí". Estos textos hacen parecerlo un español que se puede leer. Muchos paquetes de autoedición y editores de páginas web usan el Lorem Ipsum como su texto por defecto, y al hacer una búsqueda de "Lorem Ipsum" va a dar por resultado muchos sitios web que usan este texto si se encuentran en estado de desarrollo. Muchas versiones han evolucionado a través de los años, algunas veces por accidente, otras veces a propósito (por ejemplo insertándole humor y cosas por el estilo).</p>
    <p id="parrafo3">Al contrario del pensamiento popular, el texto de Lorem Ipsum no es simplemente texto aleatorio. Tiene sus raíces en una pieza clásica de la literatura del Latín, que data del año 45 antes de Cristo, haciendo que este adquiera más de 2000 años de antigüedad. Richard McClintock, un profesor de Latín de la Universidad de Hampden-Sydney en Virginia, encontró una de las palabras más oscuras de la lengua del latín, "consecteur", en un pasaje de Lorem Ipsum, y al seguir leyendo distintos textos del latín, descubrió la fuente indudable. Lorem Ipsum viene de las secciones 1.10.32 y 1.10.33 de "de Finnibus Bonorum et Malorum" (Los Extremos del Bien y El Mal) por Cícero, escrito en el año 45 antes de Cristo. Este libro es un tratado de teoría de éticas, muy popular durante el Renacimiento. La primera linea del Lorem Ipsum, "Lorem ipsum dolor sit amet..", viene de una línea en la sección 1.10.32</p>
    <p><a href="http://www.google.es" title="Buscador Google">Buscador Google</a></p>
    <p><a href="imagenes/imagen1.jpg">Imagen 1</a></p>
  </body>
</html>
```

IMÁGENES

<IMG>

Las imágenes utilizadas en una página web pueden ser de dos tipos: de contenido o de decoración. En el primer caso, si la imagen pertenece al contenido y tema tratado en esa página, debería incluirse mediante una etiqueta HTML *<img>,* pero si por el contrario pertenece a la decoración de la página, deberíamos incluirla mediante estilos (CSS).

Entre los atributos más importantes de esta etiqueta encontramos:

- *src*: indica el nombre o la URL de la imagen a mostrar.
- *alt*: establece un texto alternativo para mostrar en el caso que la imagen no se pueda mostrar.

```
<img src=”imágenes/calles.jpg” alt=”Calles con tiendas”>
```

<FIGURE> Y <FIGCAPTION>

Con estas dos etiquetas podemos asociar una imagen con su pie de imagen:

```
<figure>
    <img src=”imágenes/calles.jpg” alt=”Calles con tiendas”>
    <figcaption>Oxford Street en hora punta</figcaption>
</figure>
```

En el primer articulo, hemos insertado una imagen. Esta es una práctica común, a menudo el texto es enriquecido con imágenes o videos. Las etiquetas *<figure>* nos permiten envolver estos complementos visuales y diferenciarlos así de la información más relevante.

Normalmente, unidades de información como imágenes o videos son descritas con un corto texto debajo. HTML5 provee un elemento para ubicar e identificar estos títulos descriptivos. Las etiquetas *<figcaption>* encierran el texto relacionado con *<figure>* y establecen una relación entre ambos elementos y su contenido.

<PICTURE>

La etiqueta *<picture>* permite mayor flexibilidad al especificar qué imágenes utiliza el sitio. Gracias a este elemento será posible escribir código HTML limpio y semántico, dejando que el navegador haga todo el trabajo de seleccionar la mejor imagen para cada situación.

Ejemplo:

```
<picture>
    <source media="(min-width: 650px)" srcset="images/kitten-stretching.png">
    <source media="(min-width: 465px)" srcset="images/kitten-sitting.png">
    <img src="images/kitten-curled.png" alt="a cute kitten">
</picture>
```

Se añade el elemento la etiqueta *<img>* al final por si el navegador no soporta la etiqueta *<picture>*

Más información:
http://librosweb.es/tutorial/el-nuevo-elemento-picture-de-html5-para-crear-imagenes-responsive/#toc-pixel-density-descriptors

TABLAS

Antes de nada, hay que dejar claro que las tablas sólo se deben utilizar para presentar datos tabulados, no para maquetar.

Para crear tablas existen una serie de etiquetas:

- *<table>*: esta etiqueta se utiliza para marcar el inicio y fin de la tabla
- *<tr>*: esta etiqueta se utiliza para las filas
- *<td>*: esta etiqueta se utiliza para las celdas
- *<th>*: esta etiqueta se utiliza para las celdas que contienen información de encabezado
- *<caption>*: esta etiqueta se utiliza para dar el título a la tabla

Para asociar las celdas con sus encabezados tenemos dos atributos *scope* y *headers*

El atributo *scope* indica el ámbito sobre el que actúa el encabezado.

- Con el valor *row* para encabezados que actúan sobre las filas.
- Con el valor *col* para encabezados que actúan sobre las columnas.

Horario de Primero A

	Lunes	Martes	Miércoles
8:00-9:00	Matemáticas	Inglés	Lengua
9:00-10:00	Lengua	Conocimiento del medio	Matemáticas

```
<table>
<caption>Horario de Primero A</caption>
  <tr>
    <td> </td>
    <th scope="col">Lunes</th>
    <th scope="col">Martes</th>
    <th scope="col">Miércoles</th>
  </tr>
  <tr>
    <th scope="row">8:00-9:00</th>
    <td>Matemáticas</td>
    <td>Inglés</td>
    <td>Lengua</td>
  </tr>
  <tr>
    <th scope="row">9:00-10:00</th>
    <td>Lengua</td>
    <td>Conocimiento del medio</td>
    <td>Matemáticas</td>
  </tr>
</table>
```

Mediante los atributos *id* y *headers* podremos indicar cuáles son los encabezados correspondientes a la celda actual, independientemente de la complejidad de la tabla:

- *id*: se utiliza en las celdas de encabezado *<th>* para proporcionar un identificador que ha de ser único.
- *headers:* se usa en las celdas de datos *<td>*, con el valor de los id correspondientes.

Horario de Primero A

	Lunes	Martes	Miércoles
8:00-9:00	Matemáticas	Inglés	Lengua
9:00-10:00	Lengua	Conocimiento del medio	Matemáticas

```
<table>
<caption>Horario de Primero A</caption>
  <tr>
    <td> </td>
    <th id="l">Lunes</th>
    <th id="m">Martes</th>
    <th id="x">Miércoles</th>
  </tr>
  <tr>
    <th id="h_1">8:00-9:00</th>
    <td headers="h_1 l">Matemáticas</td>
    <td headers="h_1 m">Inglés</td>
    <td headers="h_1 x">Lengua</td>
  </tr>
  <tr>
    <th id="h_2">9:00-10:00</th>
    <td headers="h_2 l">Lengua</td>
    <td headers="h_2 m">Conocimiento del
medio</td>
    <td headers="h_2 x">Matemáticas</td>
  </tr>
</table>
```

Las celdas también tienen estos dos atributos:

- *rowspan*: hace que la celda ocupe dos celdas
- *colspan*: hace que la celda ocupe dos columnas

LISTAS

El lenguaje HTML define tres tipos diferentes de listas para agrupar los elementos:

- Listas no ordenadas (se trata de una colección simple de elementos en la que no importa su orden)
- Listas ordenadas (similar a la anterior, pero los elementos están numerados y, por tanto, importa su orden)
- Listas de definición (un conjunto de términos y definiciones similar a un diccionario).

LISTAS NO ORDENADAS

Las listas no ordenadas son las más sencillas y las que más se utilizan. Una lista no ordenada es un conjunto de elementos relacionados entre sí pero para los que no se indica un orden o secuencia determinados. La etiqueta *<ul>* encierra todos los elementos de la lista y la etiqueta *<li>* cada uno de sus elementos.

```
<ul>
    <li>Inicio</li>
    <li>Noticias</li>
    <li>Artículos</li>
    <li>Contacto</li>
</ul>
```

El navegador por defecto muestra los elementos de la lista tabulados y con una pequeña viñeta formada por un círculo negro. Como ya se sabe, el aspecto con el que se muestran los elementos de las listas se puede modificar mediante las hojas de estilos CSS.

LISTAS ORDENADAS

Las listas ordenadas son casi idénticas a las listas no ordenadas, salvo que en este caso los elementos relacionados se muestran siguiendo un orden determinado. Cuando se crea por ejemplo una lista con las instrucciones de un producto, es importante el orden en el que se realiza cada paso. Cuando se muestra un índice o tabla de contenidos en un libro, es importante el orden de cada elemento del índice.

En todos estos casos, la lista más adecuada es la lista ordenada, que se define mediante la etiqueta *<ol>*. Los elementos de la lista se definen mediante la etiqueta *<li>*, la misma que se utiliza en las listas no ordenadas.

```
<ol>
    <li>Enchufar correctamente</li>
    <li>Comprobar conexiones</li>
    <li>Encender el aparato</li>
</ol>
```

El navegador muestra la lista de forma muy parecida a las listas no ordenadas, salvo que en este caso no se emplean viñetas gráficas en los elementos, sino que se numeran de forma consecutiva. La numeración es al revés si ponemos el atributo *reversed*. El tipo de numeración empleada también se puede modificar aplicando hojas de estilos CSS a los elementos de la lista.

LISTAS DE DEFINICIÓN

El funcionamiento de las listas de definición es similar al de un diccionario, ya que cada elemento de la lista está formado por términos y definiciones. La etiqueta *<dl>* crea la lista de definición y las etiquetas *<dt>* y *<dd>* definen respectivamente el término y la descripción de cada elemento de la lista.

```
<dl>
    <dt>SGML</dt>
    <dd>Metalenguaje para la definición de otros lenguajes de marcado</dd>
    <dt>XML</dt>
    <dd>Lenguaje basado en SGML y que se emplea para describir datos</dd>
    <dt>RSS</dt>
    <dt>GML</dt>
    <dt>XHTML</dt>
    <dt>SVG</dt>
    <dt>XUL</dt>
    <dd>Lenguajes derivados de XML para determinadas aplicaciones</dd>
</dl>
```

FORMULARIOS

Los formularios web permiten a los usuarios insertar datos, tomar decisiones, comunicar información y cambiar el comportamiento de una aplicación.

<FORM>

La etiqueta que contiene los formularios es la etiqueta *<form>*.

```
<form name="miformulario" id="miformulario" method="get">
…
</form>
```

La etiqueta *<form>* requiere de unos pocos atributos para determinar la forma en la que la información va a ser procesada y enviada:

- *name:* es el nombre del formulario**.**
- *method:* determina el método utilizado para enviar el formulario al servidor. Hay dos valores posibles: *get* y *post*. El método *get* envía los datos en la URL y por lo general no tienen más de 255 caracteres. Con el método *post* la información a enviar puede ser ilimitada y esta no es visible para el usuario.
- *action:* declara la url del archivo en el servidor que procesa la información enviada por el formulario.

<INPUT>

El elemento más importante en un formulario es *<input>*. Este elemento genera un campo de entrada para que el usuario introduzca datos en él. Las características del elemento y el tipo de datos permitidos dependerán del valor del atributo type. Este atributo determina el tipo de entrada que se espera de los usuarios.

Vamos a ver los tipos de input más conocidos

TIPO TEXT

Genera un campo de entrada para insertar texto genérico.

```
<input type=”text” name=”nombre” id=”nombre”>
```

TIPO HIDDEN

Este tipo esconde el campo de entrada, no es visible para el usuario. Por lo general se utiliza para enviar información complementaria.

TIPO PASSWORD

Genera un campo de entrada para introducir una contraseña. Enmascara los caracteres para ocultar la información probada. Los caracteres introducidos por el usuario por lo general aparecen en la pantalla como asteriscos o puntos, dependiendo del navegador.

```
<input type=”password” name=”contrasena” >
```

TIPO CHECKBOX

Este valor genera una casilla de verificación. Este tipo de entrada requiere la declaración del atributo *value* para especificar el valor que debe enviarse. El valor se envía solo si la casilla de verificación es seleccionada (si no se especifica valor se envía el valor on). Si queremos que la casilla aparezca seleccionada tendremos que añadir el atributo *checked*.

<input type=”checkbox” name=”terminos” value=”acepto_terminos” checked>Acepto lo terminos

TIPO RADIO

Genera un botón de radio para seleccionar una opción entre varias propuestas. Utilizando el mismo valor para el atributo *name*, es posible agrupar varias opciones juntas. El valor enviado será el del atributo *value* del botón seleccionado. El atributo *checked* declara el botón que se selecciona de manera predeterminada.

<input type=”radio” value=”M” name=”sexo” checked>Mujer
<input type=”radio” value=”H” name=”sexo”>Hombre

TIPO FILE

Genera un campo de entrada para seleccionar un archivo del ordenador del usuario.

<input type=”file” name=”archivo”>

TIPO BUTTON

Genera un botón que no realiza ninguna acción. Se trata de un botón de usos múltiples que tiene que ser controlado por código JavaScript.

<*button type="button">Estoy en huelga...</button>*

El atributo *type* tiene tres posibles valores:

- *button:* no tiene ninguna acción preasignada
- *reset:* borra todos los valores cambiados del formulario
- *submit:* envía el formulario al pinchar en el botón

TIPO SUBMIT

Genera un botón para enviar el formulario. Utiliza el atributo *value* para declarar el texto que aparecerá en el botón.

<input type=”submit” value=”Aceptar”>

TIPO RESET

Genera un botón para borrar todos los valores cambiados del formulario. Utiliza el atributo *value* para declarar el texto que aparecerá en el botón.

<input type=”reset” value=”Limpiar”>

TIPO IMAGE

Este tipo carga una imagen que se utilizará como un botón de envío. En el atributo *src* se especifica la url de la imagen. Al pinchar sobre el botón se envía el formulario

<input type="image" src="enviar.png">

TIPO EMAIL

Genera un campo de texto que será controlado por el navegador para validar si se trata de un email o no. Si la validación falla, el formulario no será enviado

<input type="email" name="miemail”>

TIPO SEARCH

Genera un campo de texto igual que el caso de tipo *text,* es solo una indicación para los navegadores. Al detectar este tipo de campo algunos navegadores cambiarán el diseño del elemento para ofrecer al usuario un indicio de su propósito.

<input type="search" name="busqueda">

TIPO URL

Este tipo de campo trabaja exactamente igual que el tipo *email* pero es específico para direcciones web. Está destinado a recibir solo URLs absolutas y retornará un error si el valor es inválido.

<input type="url" name="miurl">

TIPO NUMBER

Genera un campo de texto para introducir algún número. El tipo *number* es sólo válido cuando recibe una entrada numérica. Existen algunos atributos nuevos que pueden ser útiles para este campo:

- *min*: el valor de este atributo determina el mínimo valor aceptado para el campo.
- *max*: el valor de este atributo determina el máximo valor aceptado para el campo.
- *step*: el valor de este atributo determina el tamaño en el que el valor será incrementado o disminuido en cada paso. Por ejemplo, si declara un valor de 5 para *step* en un campo que tiene un valor mínimo de 0 y máximo de 10, el navegador sólo permitirá los valores 0, 5 y 10 permitirá especificar valores entre 0 y 5 o entre 5 y 10.

<input type="number" name="numero" min="0" max="10" step="5">

No es necesario especificar ambos atributos (*min* y *max*), y el valor por defecto para *step* es 1.

TIPO RANGE

Este tipo le permite al usuario seleccionar un valor a partir de una serie de valores o rango. Normalmente es mostrado en pantalla como un puntero deslizable.

El tipo *range* usa los atributos *min* y *max* para configurar los límites del rango. También puede utilizar el atributo *step* para establecer el tamaño en el cual el valor del campo será incrementado o disminuido en cada paso.

<input type="range" name="numero" min="0" max="10" step="5">

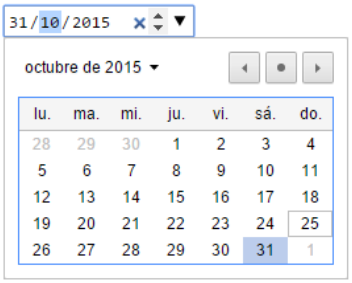
Podemos declarar el valor inicial utilizando el atributo *value*.

TIPO DATE

Gracias al tipo *date* es el navegador el que se encarga de construir un calendario o las herramientas necesarias para facilitar el ingreso de este tipo de datos.

<input type="date" name="fecha">

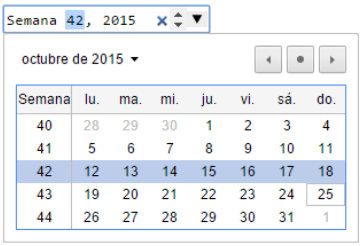
La interface no fue declarada en la especificación. Cada navegador provee su propia interface y a veces adaptan el diseño al dispositivo en el cual la aplicación está siendo ejecutada. Normalmente el valor guardado en el campo *value*, que es el que envía el formulario tiene la sintaxis año-mes-día, aunque la visualicemos en otro formato.



TIPO WEEK

Este tipo de campo ofrece una interface similar a *date*, pero para seleccionar una semana completa. Normalmente el valor esperado tiene la sintaxis 2011-W50 donde 2011 es al año y 50 es el número de la semana.

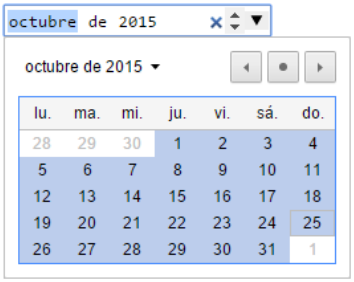
<input type="week" name="semana">



TIPO MONTH

Este tipo es específico para seleccionar meses. Normalmente el valor esperado tiene la sintaxis año-mes.

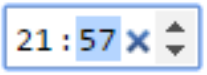
<input type="month" name="mes">



TIPO TIME

El tipo de campo *time* es similar a *date*, pero solo para la hora. Normalmente el valor esperado tiene la sintaxis hora:minutos:segundos, pero también puede ser solo hora:minutos.

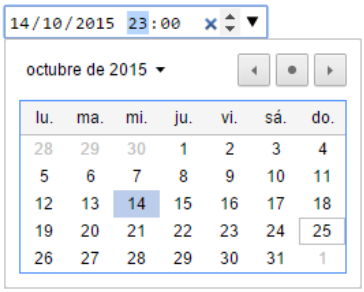
<input type="time" name="hora" >



TIPO DATETIME-LOCAL

El tipo de campo *datetime-local* es para ingresar la fecha y hora.

<input type="datetime-local" name="tiempolocal">

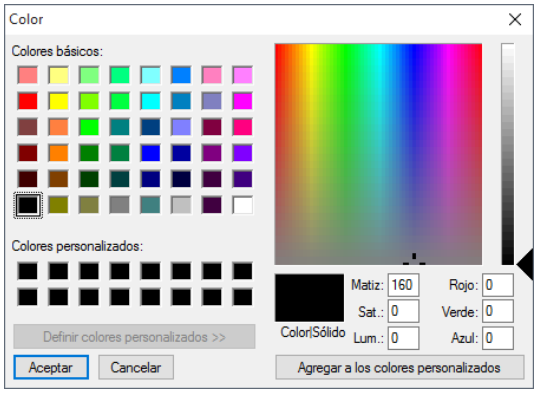


TIPO COLOR

Existe también un tipo predefinido para seleccionar colores. Normalmente el valor esperado para este campo es un número hexadecimal, como #00FF00.

<input type="color" name="micolor">

Al pinchar sobre le color se abre la siguiente ventana



ATRIBUTOS DE INPUT

El elemento *<input>* admite varios atributos, algunos dependen del tipo:

- *type*: es el que indica el tipo de control de datos que se ha elegido.
- *src*: en el caso de que sea de tipo *image* indica la url de la imagen.
- *value*: indica el valor que enviará el formulario.
- *alt*: cuando el tipo es una imagen y esta no se puede cargar, se muestra este texto alternativo.
- *checked*: indica si el elemento está seleccionado por defecto, solo se utiliza en algunos tipos como *checkbox* y *radio*.
- *disabled*: es un valor booleano que indica que el elemento está desactivado, con lo cual no admite entrada de datos, y suele aparecer en gris claro por defecto.
- *readonly*: valor booleano que indica que el campo no se puede modificar, solo lectura.
- *size*: indica el tamaño del campo.
- *maxlength*: indica la cantidad de caracteres que se pueden introducir en el campo.
- *autofocus*: indica que tendrá el foco cuando se cargue la página
- *name*: nombre que identifica al campo de datos.
- *id*: identificador que identifica el campo de datos, debe ser único en la página web
- *autocomplete:* puede tomar dos valores: *on* y *off*. El valor por defecto es *on*. Cuando es configurado como *off*, el elemento *<input>* tiene la función de autocompletar desactivada, y por tanto no muestra los textos de las entradas anteriores como posibles valores. También puede ser usado en el elemento *<form>* para afectar a todos los elementos del formulario.
- *novalidate:* es un atributo booleano cuyo por defecto todos los campos se validan a menos que el atributo *novalidate* esté presente. A nivel de formulario existe el atributo análogo *formnovalidate.*

```
<form name="miformulario" method="get" action="procesar.php">
    <input type="email" name="miemail"/>
    <input type="submit" value="Enviar"/>
    <input type="submit" value="Grabar" formnovalidate/>
</form>
```

En el ejemplo, el formulario va a ser validado, pero se ofrece un Segundo botón de envío con el atributo *formn ovalidate* para que el formulario pueda ser enviado sin pasar por el proceso de validación.

- *placeholder*: el atributo *placeholder* representa una sugerencia corta, una palabra o frase provista para ayudar al usuario a ingresar la información correctamente. El valor de este atributo es presentado en pantalla por los navegadores dentro del campo, como una marca de agua que desaparece cuando el elemento es enfocado.

  <input type="search" name="busqueda" placeholder="escriba su búsqueda">

- *pattern*: el atributo *pattern* es para propósitos de validación. Usa expresiones regulares para personalizar reglas de validación. Algunos de los tipos de campo ya estudiados validan cadenas de texto específicas, pero no permiten hacer validaciones personalizadas, como por ejemplo un código postal que consiste en 5 números. No existe ningún tipo de campo predeterminado para esta clase de entrada.

  El atributo *pattern* nos permite crear nuestro propio tipo de campo para controlar esta clase de valores no ordinarios. Se puede incluir el atributo *title* para personalizar mensajes de error.

  <input pattern="[0-9]{5}" name="codigopostal" title="inserte los 5 números de su código postal">

  Par más información mira el apartado de expresiones regulares (al final de este documento).
- *multiple:* el atributo *multiple* es otro atributo booleano que puede ser usado en algunos tipos de campo (por ejemplo, *email* o *file*) para permitir el ingreso de entradas múltiples en el mismo campo. Los valores insertados deben estar separados por coma para ser válidos.

  <input type="email" name="miemail" multiple>

  El código de arriba permite la inserción de múltiples valores separados por coma, y cada uno de ellos será validado por el navegador como una dirección de email.

MÁS ELEMENTOS DE LOS FORMULARIOS

<LABEL>

La etiqueta *<label>* permite indicar semánticamente qué texto funciona como etiqueta de un campo de formulario. Para establecer la relación entre la etiqueta y el control del formulario debemos asociarla explícitamente con el atributo *for*, cuyo valor debe ser igual al del *id* del control del formulario, *id*, que, a su vez, debe ser único en la página.

<label for="busqueda">Buscar: </label>
<input type="search" name="busqueda" id="busqueda">

El elemento *<label>* debe usarse con los elementos: *<textarea>*, *<select>*, y los *<input>* de tipo: *text*, *checkbox*, *radio*, *file* y *password*.

Por el contrario, el elemento *<label>* no debe usarse con los *<input>* de tipo *image*, que se etiquetan utilizando el atributo *alt*.

```
<input type="image" name="Enviar" src="boton.gif" alt="Enviar">
```

La etiqueta *<label>* tampoco debe usarse en los *<input>* de tipo *submit* o *reset*, los cuales se etiquetan con su atributo *value*. Tampoco debe usarse con los *<input>* de tipo *hidden*, que no se etiquetan, ni con el elemento *<button>*, en el cual se usa su contenido como etiqueta.

```
<button type="submit" name="Enviar">Enviar</button>
```

<FIELDSET>

La etiqueta *<fieldset>* nos permite agrupar semánticamente una serie de controles de formulario y etiquetarlos en su conjunto con una descripción del grupo, que incluimos mediante el elemento *<legend>*, y de esta manera comprender la relación entre los controles e interactuar con el formulario de forma más rápida y efectiva.

El uso de *<fieldset>* es especialmente importante en los grupos de *<input>* de tipo *radio* y *checkbox*. En estos casos la etiqueta individual de cada uno no transmite plenamente el contexto descriptivo del grupo, y es esencial que se agrupen para facilitar que sean tratados semánticamente como un único conjunto con una descripción adicional a nivel de grupo.

```
<fieldset>
    <legend>Estoy interesado en recibir información sobre:</legend>
    <input type="checkbox" id="deportes" name="deportes" value="dp">
    <label for="deportes">Deportes</label>
    <input type="checkbox" id="moda" name="moda" value="md">
    <label for="moda">Moda</label>

</fieldset>
```

<TEXTAREA>

La etiqueta *<textarea>* genera un campo de texto de varias líneas. El tamaño puede ser declarado utilizando los atributos *rows* y *cols*. Muchos de los atributos del elemento *<input>* también se pueden aplicar aquí.

```
<label for="mitexto">Area de Texto: </label>
<textarea name="mitexto" id="mitexto" rows="5" cols="30"></textarea>
```

<SELECT>

La etiqueta *<select>* genera una lista de opciones que se declaran con la etiqueta *<option>*. El formulario enviará el valor del atributo *name* del elemento *<select>* y el valor del atributo *value* de las opciones seleccionadas. Podemos usar el atributo *selected* para marcar la opción que queremos que aparezca seleccionada. El elemento *<select>* además de muchos de los atributos del elemento *<input>* también va a contar con los atributos *multiple* para permitir que se elija más de una opción y el atributo *size* para indicar las líneas del cuadro *<select>.*

```
<label for="milista">Seleccionar: </label>
<select name="milista" id="milista">
    <option value="1">uno</option>
    <option value="2">dos</option>
    <option value="3">tres</option>
</select>
```

Cuando hay muchas opciones el elemento *<optgroup>* permite hacer grupos con los *<option>*

```
<select name="provincia" >
    <optgroup label="Euskadi">
        <option value="1">Araba</option>
        <option value="2">Bizkaia</option>
        <option value="3">Gipuzkoa</option>
    </optgroup>
    <optgroup label="Cataluña">
        <option value="4">Barcelona</option>
        <option value="5">Gerona</option>
        <option value="6">Lerida</option>
        <option value="7">Tarragona</option>
    </optgroup>
</select>
```

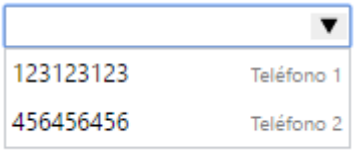
<DATALIST>

El elemento *<datalist>* es un elemento específico de formularios usado para construir una lista de ítems que luego, con la ayuda del atributo *list*, será usada como sugerencia en un campo del formulario.

```
<datalist id="informacion">
    <option value="123123123" label="Teléfono 1">
    <option value="456456456" label="Teléfono 2">
</datalist>
```

Este elemento utiliza el elemento *<option>* en su interior para crear la lista de datos a sugerir. Con la lista ya declarada, lo único que resta es referenciarla desde un elemento *<input>* usando el atributo *list*:

```
<input type="tel" name="telefono" id="telefono" list="informacion"/>
```



<PROGRESS>

Este elemento no es específico de formularios, pero debido a que representa el progreso en la

realización de una tarea, y usualmente estas tareas son comenzadas y procesadas a través de formularios, puede ser incluido dentro del grupo de elementos para formularios. El elemento *<progress>* utiliza dos atributos para configurar su estado y límites. El atributo *value* indica qué parte de la tarea ya ha sido procesada, y *max* declara el valor a alcanzar para que la tarea se considere finalizada.

```
<progress value="5" max="10"></progress>
```

ATRIBUTO REQUIRED

El atributo booleano *required* no dejará que el formulario sea enviado si el campo se encuentra vacío.

```
<input type="email" name="miemail" required/>
```

ATRIBUTO FORM

El atributo *form* es una adición útil que nos permite declarar elementos para un formulario fuera del ámbito de las etiquetas *<form>*. En HTML5 podemos insertar los elementos en cualquier parte del código y luego hacer referencia al formulario que pertenecen usando su nombre y el atributo *form*.

```
<body>
    <nav>
        <input type="search" name="busqueda" id="busqueda" form="formulario">
    </nav>
    <section>
        <form name="formulario" id="formulario" method="get">
            <label for="name">Nombre: </label>
            <input type="text" name="nombre" id="nombre">
            <input type="submit" value="Enviar"/>
        </form>
    </section>
</body>
```

VIDEO Y AUDIO

VIDEO

Con la etiqueta *<video>* podemos introducir videos en nuestra página html de forma sencilla. Esta etiqueta tiene varios atributos para su configuración, pero sólo el atributo *src* es obligatorio, ya que es el atributo que especifica la fuente del video.

```
<video src="multimedia/trailer.mp4" controls></video>
```

A pesar de que el elemento *<video>* y sus atributos son estándar, no existe un formato estándar de video para todos los navegadores por lo que tendremos que tener el video en varios formatos para asegurarnos que se verá en todos los navegadores. OGG es reconocido por Firefox, Google Chrome y Opera, mientras que MP4 trabaja en Safari, Internet Explorer y también Google Chrome.

Este atributo *src* puede ser reemplazado por el elemento *<source>* y su propio atributo *src* para declarar varias fuentes con diferentes formatos, como en el siguiente ejemplo:

```
<video>
    <source src="multimedia/trailer.mp4">
    <source src="multimedia/trailer.ogg">
</video>
```

Además del atributo *src* también podemos usar los siguientes atributos:

- *autoplay:* el navegador comenzará a reproducir el video automáticamente cuando se cargue la página.
- *controls:* muestra los controles de video provistos por el navegador (play, pause…).
- *loop:* el navegador comenzará a reproducir el video nuevamente cuando llega al final.
- *poster:* este atributo proveerá una imagen que será mostrada mientras el video no se está reproduciendo.

```
<video controls loop poster="imagenes/poster.jpg">
    <source src="multimedia/trailer.mp4">
    <source src="multimedia/trailer.ogg">
</video>
```

AUDIO

Con la etiqueta *<audio>* podemos introducir sonidos/música en nuestra página html de forma sencilla y comparte casi las mismas características del elemento *<video>*.

```
<audio src="multimedia/beach.mp3" controls></audio>
```

Veamos los atributos más utilizados:

- *src:* este atributo especifica la URL del archivo que va a ser reproducido. Al igual que en el elemento *<video>* normalmente será reemplazado por el elemento *<source>* para ofrecer diferentes formatos de audio entre los que el navegador pueda elegir.

- *controls:* este atributo activa la interface que cada navegador provee por defecto para controlar la reproducción del audio.
- *autoplay:* el audio comenzará a reproducirse automáticamente tan pronto como sea posible. Debe ir acompañado por el atributo muted (accesibilidad)
- *loop:* el navegador reproducirá el audio una y otra vez de forma automática.

```
<audio controls>
    <source src="multimedia/beach.mp3"/>
    <source src="multimedia/beach.ogg"/>
</audio>
```

OTROS

<DETAILS> Y <SUMMARY>

Una característica importante de los sitios web es la posibilidad de mostrar información adicional cuando así es solicitado por el usuario. Para evitar el uso de JavaScript y facilitar la creación de esta herramienta, HTML5 incorpora los elementos *<details>* y *<summary>*. El elemento *<details>* declara la herramienta de ampliación de información y, dentro de este elemento, se especifica el título de la herramienta con el elemento *<summary>* además de toda la información que está oculta.

```
<details>
    <summary>Clic para Expandir</summary>
    <p>Hola! Soy Visible!</p>
</details>
```

El código de arriba se vería como la imagen de la izquierda y al pinchar sobre él se vería como en la imagen de la derecha.

DATA-*

Si necesitas declarar datos adicionales, entonces el atributo *data-** es la solución. Este atributo permite asignar información adicional a un elemento sin exponerla al usuario. Se trata de información que está destinada a ser utilizada por la página web y es accesible desde el código Javascript. El nombre del atributo está compuesto por el prefijo *data-* seguido por un nombre personalizado.

```
<p data-firstinitial="J" data-last="Autor">J Autort</p>
```

TRANSLATE

La mayoría de los navegadores actuales traducen automáticamente cualquier documento en el que detectan lengua extranjera, pero en algunos casos hay textos que deben permanecer intactos, como pueden ser nombres, títulos originales, etc. El atributo *translate* se añade para controlar el proceso de traducción desde HTML.

```
<p>My favorite movie is <span translate="no">Mrs. Robinson</span></p>
```

El atributo *translate* puede tener dos valores: yes o no. Por defecto, el valor es yes, salvo que un elemento de mayor jerarquía haya sido cambiado a no.

CONTENTEDITABLE

Con el atributo *contenteditable*, podemos convertir cualquier elemento HTML en uno editable.

```
<p>Mi película preferida es <span contenteditable="true">Casablanca</span></p>
```

Ese atributo admite dos valores posibles: *true* o *false*. Por defecto ningún elemento se puede editar.

SPELLCHECK

Otra característica que se activa automáticamente en los navegadores es la revisión ortográfica. Si bien ésta es una herramienta útil con la que los usuarios esperan contar todo el tiempo, puede ser inapropiada en algunas circunstancias. Para activar o desactivar esta característica se puede utilizar el atributo *spellcheck*.

* Mi película preferida es Casablanca *

Este atributo tiene dos valores posibles, *true* o *false*. La función siempre está activada a menos que se declare el valor de este atributo como *false*

TABINDEX

Con este atributo podemos seleccionar el orden de los elementos obtendrán el foco al recorrer los elementos de la página web con la tecla tabulador

<a href=”anterior.html” tabindex=”2”>
<a href=”recargar.html” tabindex=”1”>
<a href=”siguiente.html” tabindex=”3”>

Por defecto el orden será el orden de escritura en el documento html

EXPRESIONES REGULARES

Las expresiones regulares describen un conjunto de elementos que siguen un patrón. Dicho de otra forma, es una regla que identifica a una serie de elementos que tiene algo en común. Un ejemplo de expresión regular podrían ser todas las palabras que comienzan por la letra "a" minúscula. La expresión regular para este patrón debe asegurarse de que la palabra comienza por "a" minúscula, el resto de palabras que precedan a la cadena podrán ser cualquier carácter.

CARACTERES ESPECIALES DE LAS EXPRESIONES REGULARES

A continuación, vamos describir algunos de los elementos que se utilizan para crear expresiones regulares.

ELEMENTO	SIGNIFICADO
^	**Principio de entrada o línea**. Este carácter indica que las cadenas deberán comenzar por el siguiente carácter. Si éste fuera una "a" minúscula, la expresión regular seria sería, ^a.
$	**Fin de entrada o línea**. Indica que la cadena debe terminar por el elemento precedido al dólar.
*****	**El carácter anterior 0 o más veces**. Indica que el carácter anterior se puede repetir en la cadena 0 o más veces.
+	**El carácter anterior 1 o más veces.** El símbolo más indica que el carácter anterior se puede repetir en la cadena una o más veces.
?	**El carácter anterior una vez como máximo**. El símbolo interrogación indica que el carácter anterior se puede repetir en la cadena cero o una vez.
.	**Cualquier carácter individual**. El símbolo punto indica que puede haber cualquier carácter individual salvo el de salto de línea.
x|y	**x o y**. La barra vertical indica que puede ser el carácter x o el y.
{n}	**n veces el carácter anterior**. El carácter anterior a las llaves tiene que aparecer exactamente n veces.
{n,m}	**Entre n y m veces el carácter anterior.** El carácter anterior a las llaves tiene que aparecer como mínimo n y como máximo m veces.
[abc]	**Cualquier carácter de los corchetes.** En la cadena puede aparecer cualquier carácter que esté incluido en los corchetes. Además, podemos especificar rangos de caracteres que sigan un orden. Si se especifica el rango [a-z] equivaldría a incluir todas las letras minúsculas del abecedario.
[^abc]	**Un carácter que no esté en los corchetes**. En la cadena pueden aparecer todos los caracteres que no estén incluidos en los corchetes. También podemos especificar rangos de caracteres como en el punto anterior.
\.	**Punto**
\b	**Fin de palabra**. El símbolo \b indica que tiene que haber un fin de palabra o retorno de carro.
\B	**No fin de palabra**. El símbolo \B indica cualquiera que no sea un límite de palabra.
\d	**Cualquier carácter dígito.** El símbolo \d indica que puede haber cualquier carácter numérico, de 0 a 9.
\D	**Carácter que no es dígito.** El símbolo \D indica que puede haber cualquier carácter siempre que no sea numérico.
\f	**Salto de página.** El símbolo \f indica que tiene que haber un salto de página.

\n	**Salto de línea.** El símbolo \n indica que tiene que haber un salto de línea.
\r	**Retorno de carro.** El símbolo \r indica que tiene que haber un retorno de carro.
\s	**Cualquier espacio en blanco.** El símbolo \s indica que tiene que haber un carácter individual de espacio en blanco: espacios, tabulaciones, saltos de página o saltos de línea.
\S	**Carácter que no sea blanco.** El símbolo \S indica que tiene que haber cualquier carácter individual que no sea un espacio en blanco.
\t	**Tabulación**. El símbolo \ t indica que tiene que haber cualquier tabulación.
\w	**Carácter alfanumérico**. El símbolo \w indica que puede haber cualquier carácter alfanumérico, incluido el de subrayado.
\W	**Carácter que no sea alfanumérico.** El símbolo \W indica que puede haber cualquier carácter que no sea alfanumérico.

EJEMPLOS

- Expresión regular para un código postal: *^\d{5}$*
- Expresión regular para una fecha: *^([0][1-9]|[12][0-9]|3[01])[\/|-]([0][1-9]|[1][0-2])\/(\d{4})$*
- Expresión regular para un email: *^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,4})$*
- Expresión regular para una matrícula: *^\d{4}[A-Za-z]{3}$*
- Expresión regular para un número de cuenta (IBAN): *^ES\d{22}$*