

## Fundamentos de Programación Web

### 1.1. Introducción a la programación web

En el desarrollo de proyectos Web, la generación del contenido de las páginas se lleva a cabo generalmente utilizando un conjunto de tecnologías en las que se integran tecnologías dedicadas, por un lado, a la generación del contenido estático, y por otro, tecnologías encaminadas a generar el contenido dinámico de las páginas Web.

Cuando se habla de contenido estático se hace referencia al contenido de las páginas Web, el cual se presenta de forma estática, invariante a las acciones que realiza el usuario, y cuyo objetivo primordial es presentar datos al usuario, los cuales normalmente no cambian con el paso del tiempo.

Por otro lado, cuando se habla de contenido dinámico se hace referencia a aquellas partes o elementos de las páginas Web que cambian con el tiempo, que sufren modificaciones y alteraciones por las acciones que realiza el usuario sobre el contenido.

### La WEB 2.0

El término Web 2.0 está asociado a aplicaciones web que facilitan la labor de compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual, a diferencia de sitios web donde los usuarios se limitan a la observación pasiva de los contenidos que se han creado para ellos.

Frente a las tradicionales páginas web estáticas (Web 1.0) donde sus visitantes solo pueden leer los contenidos ofrecidos por su autor o editor, en la Web 2.0 todos los cibernautas pueden elaborar contenidos y compartirlos, opinar,

etiquetar/clasificar... Esto supone una democratización de las herramientas de acceso a la información y de elaboración de contenidos, aunque como no todos los que escriben en Internet son especialistas, se mezclarán los conocimientos científicos con las simples opiniones y las falsedades.

### **Conceptos y terminología**

- **Bookmark o favorito:** en la World Wide Web, un bookmark es una dirección web que fue adicionada a una lista de favoritos.
- **Browser o navegador:** aplicación que permite al usuario navegar en la World Wide Web.
- **Computación móvil:** es un paradigma computacional derivado de la tecnología de redes sin cables que es utilizada por dispositivos móviles, tales como portátiles, tabletas, smartphones.
- **Cookie:** conjunto de información que es intercambiada entre el browser y el servidor web, y que es almacenada por el navegador del cliente.
- **Download o descargar:** transferir datos de otra máquina a la mía
- **Gadgets:** miniaplicaciones o bloques de código que son incorporados en las páginas web.
- **Plugin:** es un programa pequeño que sirve normalmente para adicionar funciones a otros programas mayores, adicionándoles alguna funcionalidad especial o muy específica.
- **Post:** mensaje publicado. Por ejemplo, entradas de texto cronológicas en blogs
- **Dominio:** Es un identificador de un conjunto de dispositivos que se encuentran publicados a través del internet.
- **Hosting:** También conocido como alojamiento, es el lugar físico en donde se encuentran los diferentes archivos (páginas web, imágenes) que serán visualizados a través de internet
- **URL:** Uniform Resource Locator, Localizador Universal de Recursos, es una cadena de caracteres la cual permite asignar una dirección única a cada recurso de información disponibles en el internet.

- **Página web:** Es un documento o archivo electrónico que es servido de forma remota o local y que es ejecutado o visualizado a través de un Browser por el cliente
- **Sitios web:** Es el conjunto de páginas web ordenadas que, relacionadas entre sí, brindan información específica sobre un tema, actividad o propósito específico.
- **Portales web:** Es la integración de diversos recursos o servicios sobre un tema, actividad o propósito específico, que permiten al usuario interactuar con la información que se presenta de una forma fácil y amigable.

## **Tipos de dominio**

**1. Genéricos:** Son dominios que no se rigen a un lugar en específico y pueden ser creados desde cualquier parte del mundo. Ejemplo:

- **com:** Usado para entidades comerciales
- **net:** Proveedores de redes
- **org:** Organizaciones no lucrativas
- **edu:** Instituciones Educativas
- **gob:** Entidades Gubernamentales

**2. Territoriales:** También llamados dominios geográficos ccTLD (Country Code Top Level Domain), las extensiones territoriales indican el país o la región de cada web. Son utilizados por las empresas para proteger la identidad de su marca en un país concreto, así como para tener más cercanía con los consumidores del país al que se va a dirigir. Ejemplo:

- **mx:** México
- **dk:** Dinamarca

## **El futuro con la Web 3.0**

Web 3.0 es la web que facilita la accesibilidad de las personas a la información, sin depender de qué dispositivo use para el acceso a ella, una web con la que interactuar para conseguir resultados más allá del hecho de compartir "información", que esta información sea compartida por cada persona de una forma inteligible y de provecho para ella y sus necesidades en cada circunstancia.

## **Frontend**

FrontEnd es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas, los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales... y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario.

Como hemos dicho, el desarrollador frontend se encarga de la experiencia del usuario, es decir, en el momento en el que este entra a una página web, debe ser capaz de navegar por ella, por lo que el usuario verá una interface sencilla de usar, atractiva y funcional.

Un desarrollador frontend debe conocer los siguientes lenguajes de programación: HTML5, CSS3, JavaScript, JQuery, Ajax.

## **Backend**

Cuando hablamos de "Backend" nos referimos al interior de las aplicaciones que viven en el servidor y al que a menudo se le denomina "el lado del servidor".

El backend del sitio web consiste en un servidor, una aplicación y una base de datos. Se toman los datos, se procesa la información y se envía al usuario. Los

desarrolladores de Frontend y Backend suelen trabajar juntos para que todo funcione correctamente.

Un desarrollador Backend debe tener amplios conocimientos de los siguientes lenguajes: frameworks y los tipos de base de datos. No siendo necesario conocer todos los lenguajes, pero sí entender y saber trabajar con algunos de ellos: ASP.NET, PHP, Python, Ruby, Node.js, Java, MySQL, SQL Server, PostgreSQL, Oracle, MongoDB.

### **Programas más populares para la creación de páginas web**

Los productos de software para la creación de páginas web permiten crear rápidamente sitios web propios con herramientas de diseño, plantillas, alojamiento de dominio y opciones de programación. También pueden incluir widgets para integración de redes sociales, procesamiento de tarjetas de crédito y funciones multimedia. Varias de estas aplicaciones también incluyen herramientas SEO integradas que mejoran el posicionamiento en los resultados de búsqueda.

- ☐ WordPress
- ☐ Joomla
- ☐ Dreamweaver
- ☐ Drupal
- ☐ Prestashop
- ☐ Google Sites
- ☐ Blogger
- ☐ Wix

### **CMS**

Un CMS (Content Management System) o Sistema de Gestión de Contenidos es un sistema online que permite poner en marcha una página web de forma sencilla y rápida. Se trata de un software que te ayuda a administrar contenidos

dinámicos, por ejemplo, un blog, un ecommerce o cualquier tipo de página web. Especialmente es para aquellos que necesitan una actualización constante. Gracias a los CMS cualquier usuario sin ningún tipo de conocimiento en programación puede administrar a través de una interfaz gráfica todos los aspectos de una página web. Desde crear y editar contenido hasta agregar imágenes y vídeos.

Los gestores de contenidos más populares en la actualidad son: WordPress y Joomla

Editores de código para crear páginas web

Los editores de código clásicos vienen a ser una solución intermedia entre los sencillos editores de texto y los sofisticados IDE (del inglés «Integrated Development Environment», o entorno de desarrollo integrado). Los primeros destacan por su amplio rango de funcionalidad, mientras que los segundos abarcan e integran varias herramientas para el desarrollo de software en una sola aplicación. Por lo general, los complejos entornos de desarrollo integrados también tienen un editor de código incorporado, aunque este solo es uno de sus muchos componentes.

- ☐ Visual Studio Code
- ☐ Brackets
- ☐ Sublime Text
- ☐ Atom Editor
- ☐ Notepad++
- ☐ También se puede utilizar el bloc de notas de Windows.

## 1.2. Lenguaje HTML

HTML son las siglas de HyperText Markup Language (Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de “etiquetas” o “marcas”, rodeadas por corchetes angulares (< >).

HTML fue desarrollado originalmente por Tim Berners-Lee mientras estaba en el CERN, y popularizado por el navegador Mosaic desarrollado en NCSA. Durante el transcurso de la década de 1990 proliferó con el crecimiento explosivo de la Web. Durante este tiempo, se añadieron etiquetas al lenguaje HTML. La web depende de los autores de páginas web y de que las compañías compartan las mismas convenciones de HTML. Esto ha motivado el trabajo conjunto sobre las especificaciones de HTML.

HTML no es propiamente un lenguaje de programación como puede serlo Java o C, sino que se basa en la utilización de un sistema de etiquetas restringido que es aplicado a un documento de texto. Otra característica es que HTML no necesita ser compilado, sino que es un lenguaje interpretado, es decir, es ejecutado a medida que se avanza por el documento HTML y su interprete es el navegador Web.

## Estructura de un documento HTML

Toda página HTML debe incluir las etiquetas <HTML> y </HTML>, estas etiquetas nos están indicando que el código contenido entre ellas va a ser HTML.

Los documentos escritos en HTML están estructurados en dos partes diferenciadas:

1. **HEAD (cabecera):** Es la primera de las dos partes en que se estructura un documento HTML. En esta zona reside información acerca del documento, y generalmente no se ve cuando se navega por él. En la zona <HEAD> se pone el elemento <TITLE></TITLE> que es una breve descripción que identifica el documento. Es lo que veremos cómo título de la ventana en los navegadores que lo permitan. Es como se conocerá nuestra página en algunos buscadores y en la agenda de direcciones (bookmarks) de los usuarios. Por tanto, parece importante pensarnos bien

cómo llamarla. Este título aparecerá en la barra superior del navegador junto con el nombre de dicho navegador.

2. **BODY (cuerpo):** El cuerpo `<BODY>` y `</BODY>` es la segunda y última de las dos partes en que se estructura un documento HTML. Esta parte al contrario que `<HEAD>` es obligatoria, ya que es aquí donde reside el verdadero contenido de la página.

## HTML5

Lo que hace el estándar HTML5 es especificar qué marcas deben utilizarse para escribir un documento web y qué significa cada una de ellas. De esta forma, cualquier navegador sabrá cómo interpretarlas. Por ejemplo, todo documento HTML ha de ir delimitado por la etiqueta de inicio y cierre `<html>`. Dentro de estas dos etiquetas tenemos dos partes bien diferenciadas: Cabecera (elemento `<head>`) y Cuerpo (elemento `<body>`).

### Estructura HTML

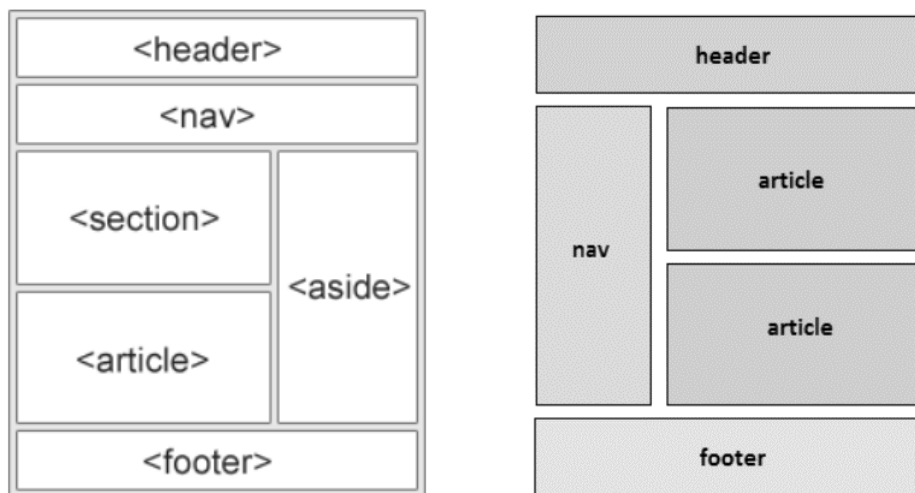
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <h1>Hola mundo</h1>
</body>
</html>
```

### Estructura HTML5

Elementos como `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>` y `<footer>` actúan más o menos como elementos `<div>`. Agrupan otros elementos en secciones de página. Sin embargo, cuando una etiqueta `<div>` puede contener cualquier tipo de información, es fácil identificar qué tipo de información iría en una región `<header>` semántica.



Ejemplos:



### Etiquetas comunes

- `<TITLE>`: Establece el título de la página web.
- `<DIV>`: Crea un contenedor dentro de una página web.
- `<TABLE>`: Es la etiqueta principal para crear una tabla
- `<TH>`: Define una columna de encabezado dentro de una tabla.
- `<TR>`: Define una fila dentro de una tabla.
- `<TD>`: Define una columna sin encabezado dentro de una tabla.
- `<P>`: Define un párrafo dentro de una página web.
- `<FORM>`: Define un formulario dentro de una página web.
- `<INPUT>`: Define un campo de ingreso de datos dentro de una página web
- `<LABEL>`: Crea un rótulo de un input
- `<IMG>`: Permite la visualización de imágenes dentro de una página web.
- `<LINK>`: Crea un vínculo que hace referencia a otro recurso de la página web
- `<A>`: Crea un hipervínculo o vínculo que direcciona a otro recurso o página web
- `<BR>`: Crea un salto de línea página web.
- `<HR>`: Crea una línea horizontal en la página web.

- **<CENTER>**: Permite centrar elementos en una página web.
- **<H1,2,3,4,5,6>**: Permite establecer texto de encabezado (Títulos) dentro de una página web.

## La cabecera HTML

La cabecera está delimitada por el elemento **<head>** (siempre contenida dentro del elemento **html**) y en ella se describe información del documento y su configuración (título, configuración, scripts y estilos).

- La cabecera está delimitada por el elemento **<head>** (siempre contenida dentro del elemento **html**) y en ella se describe información del documento y su configuración (título, configuración, scripts y estilos).
- **<TITLE>**: Establece el título de la página web que se mostrará en la pestaña del navegador.
- Icono de la página web se puede agregar con la etiqueta **link**.

```
<link rel="icon" href="http://example.com/favicon.png">
```

- Los metadatos (elementos **<meta>**) proporcionan metainformación sobre el documento (información procesable automáticamente por programas que analicen la página). Normalmente usan los atributos **name** (para definir un tipo de metadato) y **content** (para definir el valor),

```
<meta charset="utf-8">
<meta name="author" content="Juan Antonio Recio" />
<meta name="keywords" content="desarrollo web, html, css, javascript" />
<meta name="description" content="El mejor libro sobre desarrollo web." />
```

- **charset** indica la codificación en la que se ha escrito el documento de la página. Es importante para que podamos ver correctamente las tildes en nuestra página. Como nuestro editor de texto guarda en codificación UTF-8 eso es lo que hemos indicado.
- **author** indica el autor de la página.

- **keywords y description** hacen referencia al contenido de la página y suelen ser usados por los motores de búsqueda como Google o Bing para indexar las páginas web. De esta forma, el buscador sabe qué términos de búsqueda son los que mejor describen a nuestra web para así mostrársela a sus usuarios.

## El cuerpo del documento HTML

- Lo primero que debemos hacer al diseñar una página web es decidir cuál es su estructura en bloques. Normalmente toda web tiene una cabecera de título, algún tipo de barra lateral de navegación, un contenido principal y quizás un pie de página. Para hacer esta estructuración suele utilizarse el elemento `<div>` o división.
- **Body:** Las etiquetas `<body>` y `</body>` rodean el contenido visible de la página. También puede llevar incluida información sobre las propiedades de la página, por ejemplo `<body bgcolor="#RRGGBB">` define el color de fondo de la página.

## Etiquetas HTML

- **Comentarios:** `<!-- comentarios -->`: Para escribir anotaciones que sirven para ayudar a la comprensión del código. Lo que se escribe dentro de esta etiqueta es ignorado por el navegador y no se muestra en la página.

## Formato de Textos

- `<b>...</b>` : Aplica negrita al texto incluido entre las etiquetas (es equivalente usar `<strong>... </strong>`)
- `<i>...</i>`: Aplica cursiva al texto incluido entre las etiquetas (es equivalente usar `<em>...</em>`)
- `<u> </u>`: Aplica subrayado

- `<font>...</font>`: Delimita un texto con un formato de fuente determinado definido por sus propiedades:
  - `<font color="#RRGGBB">... </font>`: Define el color del texto, donde cada letra RRGGBB es un valor hexadecimal (de 0 a F) que indica el color.
  - `<font face="arial">... </font>`: Determina el tipo de fuente de texto, es decir, la tipografía. La etiqueta `<font>` puede incluir los tres parámetros (tamaño, fuente y color): `<font size=X color=#XXYYZZ face=fuente escogida> ..... </font>`
- `h1, h2, h3, h4, h5, h6`: Indican 6 niveles de formato de encabezados, en los que `<h1>...</h1>` delimitaría el tipo de fuente de mayor tamaño.

## Formato de Párrafos

- `<br>`: Introduce un salto de línea.
- `<p>... </p>`: Delimita un párrafo de texto.
  - `<p align='center'>`: Texto del párrafo con alineación centrada.
  - `<p align='left'>`: Párrafo alineado a la izquierda.
  - `<p align='right'>`: Párrafo alineado a la derecha. `<p align='justify'>`: Texto del párrafo con alineación justificada.
- `<hr>`: Inserta una línea horizontal.
- `<div>`: Se utiliza para delimitar una sección dentro del documento, de forma que agrupe un número de elementos para luego añadirle un estilo determinado o realizar operaciones sobre ese bloque específico.

## Creación de Listas

- **Lista no numerada:**

```
<ul> Inicio de la lista
  <li>primer elemento de la lista</li>
  <li>segundo elemento de la lista</li>
  <li>tercer elemento de la lista</li>
</ul> Cierra la lista
```

- **Lista numerada:**

```
<ol> Inicio de la lista
    <li>primer elemento de la lista</li>
    <li>segundo elemento de la lista</li>
    <li>tercer elemento de la lista</li>
</ol> cierra lista.
```

## Creación de Enlaces

- `<a href="http://www.ejemplo.com">Nombre del enlace</a>`: Cuando se pulsa sobre el texto "Nombre del enlace" en la web, se va al vínculo indicado en la dirección de href.
- `<a href="mailto:ejemplo@ejemplo.com">Nombre del enlace</a>`: Cuando se pulsa sobre el texto "Nombre del enlace" en la web, se abre una ventana para enviar un correo electrónico a la dirección indicada.

## Imágenes

- ``: Inserta una imagen que se encuentra en la ruta indicada por "dirección de la imagen". También se pueden incluir una serie de propiedades:
- `<img ... border="X">`: Establece un borde de X pixels en torno a la imagen.
- `<img ... height="XX" width="YY">`: Establece un tamaño de la imagen, donde XX e YY son al altura y anchura en pixels respectivamente.
- `<img ... title="texto explicativo">`: Se muestra un texto al pasar el ratón sobre la imagen.
- `<img ... align="bottom">`: Alineación inferior de la imagen respecto al texto.
- `<img ... align="middle">`: Alineación de la imagen en medio del texto.
- `<img ... align="top">`: Alineación superior de la imagen respecto al texto.
- `<img ... align="left">`: Alineación izquierda de la imagen en el párrafo.
- `<img ... align="right">`: Alineación derecha de la imagen en el párrafo.
- `<img ... hspace="x">`: Espacio horizontal en pixeles entre la imagen y el texto.

- `<img ... vspace="y">`: Espacio vertical en píxeles entre la imagen y el texto.

## Tablas

- `<table>...</table>`: Define dónde comienza y termina la tabla.
- `<table width="X">`: Determina la anchura de la tabla. Puede darse en píxeles o en porcentaje.
- `<table height="X">` Determina la altura de la tabla en píxeles.
- `<table border="X">`: Establece el grosor en píxeles del borde de la tabla
- `<table cellpadding="X">`: Define el espacio en píxeles entre las celdas.
- `<table cellspacing="X">`: Define el espacio en píxeles entre el borde y el texto.
- `<tr>...</tr>`: Indica el comienzo y el fin de cada una de las filas de la tabla.
- `<td>...</td>`: Indica el comienzo y el fin de cada una de las columnas o celdas dentro de las filas.

Ejemplo:

```
<table>
  <tr>
    <th>Nombre</th>
    <th>Edad</th>
    <th>Teléfono</th>
  </tr>
  <tr>
    <td>Juan Perez</td>
    <td>32</td>
    <td>0997445142</td>
  </tr>
</table>
```

## Archivos multimedia

HTML5 incluye en el propio lenguaje dos elementos que sirven para añadir audio y vídeo a nuestras páginas web:

- `<video>`: incrusta un vídeo en la página.
- `<audio>`: incrusta un elemento de audio.

Ejemplo:

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

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

## El elemento Inline Frame

`<iframe>` representa un contexto de navegación anidado, el cual permite incrustar otra página HTML en la página actual:

```
<iframe src="default.asp" width="100%" height="300" ></iframe>
```

## Formularios

Un formulario es un área especial de interactividad, dentro de la cual es posible insertar un conjunto de elementos que permiten recibir la respuesta a una o varias preguntas formuladas al usuario. Ejemplo de estos elementos son una caja de texto, un botón de opción, una lista desplegable, etc. La etiqueta necesaria para la creación de un formulario es `<form></form>`, que engloba el comienzo y el final del formulario. Por tanto, los elementos que reciban los datos deberán estar definidos dentro de la apertura y cierre de dicha etiqueta.

## Procesamiento de formularios

La etiqueta necesita de dos atributos para que el formulario sea funcional. El primer atributo es `action`. Este atributo debe contener la URL a la que se redirigirán los datos del formulario. Es decir, es un hipervínculo a otra página que será la que extraiga los datos del formulario y los procese.

El segundo atributo importante es `method`, e indica cuál es el método por el cual el formulario va a enviar los datos. Las opciones de esta etiqueta son POST o

GET. En el caso de POST, los datos se mandan de forma oculta, mientras que en el método GET los datos son enviados de forma directa dentro de la URL.

Un formulario es una sección dentro de un documento HTML que sirve para que los usuarios introduzcan una serie de datos y los envíen a un servidor web o a un servidor de correo. Las etiquetas que se usan principalmente en los formularios HTML son:

- `<form>...</form>`: Indican el comienzo y el fin del formulario.
- `<label>...</label>`: Delimitan una etiqueta de texto dentro del formulario.
- `<input>...</input>`: Indican el principio y final de algún control de entrada de datos por parte del usuario. Pueden ser de varios tipos, según se defina su atributo type:
  - text: Entrada de texto.
  - password: Igual que text pero el texto introducido se presentan de tal modo que se oculten los caracteres, normalmente mediante un serie de asteriscos. Este tipo de control suele utilizarse para introducir contraseñas.
  - checkbox: Casilla de verificación o checkbox.
  - radio: Elemento de selección entre varias opciones de tipo radio.
  - submit: Botón para enviar el formulario.
  - image: Botón para enviar el formulario pero con una imagen de fondo.
  - reset: Botón para borrar todo el contenido de datos del formulario.
  - hidden: campo oculto, envía una información que no se muestra en el formulario.
  - file: Permite adjuntar un archivo desde el ordenador.
- `<select>...</select>`: Delimita el principio y el fin de una lista desplegable de opciones. Cada opción se indica mediante la etiqueta `<option>`.
- `<textarea>...</textarea>`: Permite introducir un texto largo.

Ejemplo de un formulario web

```
<form action="ejemplo.php" method="get">  
  <p>Nombre: <input type="text" name="nombre" size="40"></p>
```



```
<p>Año de nacimiento: <input type="number" name="nacido"
min="1900"></p>
<p>Sexo:
  <input type="radio" name="hm" value="h"> Hombre
  <input type="radio" name="hm" value="m"> Mujer
</p>
<p>
  <input type="submit" value="Enviar">
  <input type="reset" value="Borrar">
</p>
</form>
```

## Nuevos Tipos de Controles de Entradas

- `<input type="email" />`: Para introducir una dirección de correo electrónico. El sistema validará que la sintaxis es correcta.
- `<input type="url" />`: Para introducir una dirección web o url. Muestra un mensaje de error por ejemplo si se introducen caracteres no válidos.
- `<input type="date" />`: Permite introducir una fecha, incorpora un calendario desplegable para elegirla.
- `<input type="number" />`: Para introducir un número, se puede indicar un rango máximo y mínimo.
- `<input type="range" />`: Presenta una barra con un rango para arrastrar con el ratón.
- `<input type="search" />`: Para introducir un término de búsqueda en la web
- `<input type="color" />`: Para elegir un color. Despliega una carta de colores.

## Nuevos Atributos para las Etiquetas

Entre los nuevos atributos para las etiquetas de tipo `<input>` se encuentran:

- **autocomplete**: La función de autocompletar el texto cuando se rellena un formulario es una característica del navegador. Sin embargo, HTML5 ofrece la posibilidad de activar o desactivar esta característica, dando el valor "on" u "off" a este atributo.

- autofocus: Indica si se colocará el cursor del ratón por defecto en el formulario al entrar en la página. □ min: Permite indicar un valor mínimo para un rango, por ejemplo en el input de tipo number.
- max: Permite indicar un valor máximo para un rango, por ejemplo en el input de tipo number.
- placeholder: Permite definir un texto que se desea que aparezca dentro del campo del formulario a modo de ayuda. Este texto se resalta en un tono diferente, de forma que, cuando se pone el cursor sobre el campo, el texto desaparece y el campo vuelve a su color habitual.
- required: Si se coloca este atributo dentro de una etiqueta input, el sistema comprobará que el campo ha sido rellenado antes incluso de pulsar el botón de envío. Si el usuario deja este campo en blanco, se mostrará un mensaje de error o simplemente se colocará el cursor de escritura sobre el primer campo vacío.

### 1.3. CSS

CSS (Cascading Style Sheets u Hojas de Estilo en Cascada) es un lenguaje de programación muy parecido a HTML que permite aplicar estilos a los distintos elementos de las páginas web, de modo que los títulos, listas y párrafos pueden verse igual en todas y cada una de las páginas.

Se puede hacer de tres formas diferentes, pero la más recomendada es la tercera.

1. **Mediante el atributo style:** El atributo style es de CSS pero se inserta dentro de las etiquetas HTML directamente. Por ejemplo, para aplicar el color rojo al fondo de la página, en el documento HTML se escribiría, dentro de la etiqueta body:

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>Esta es una página con fondo rojo</p>
  </body>
</html>
```

2. **Mediante la etiqueta style:** También de esta forma se insertaría código CSS dentro del archivo HTML, en esta ocasión mediante una etiqueta HTML llamada style. Por ejemplo, para conseguir el mismo efecto que en el caso anterior, se escribiría:

```
<html>
  <head>
    <title>Ejemplo</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>Esta es una página con fondo rojo</p>
  </body>
</html>
```

3. **Mediante un archivo CSS externo.**

```
<html>
  <head>
    <title>Mi documento</title>
    <link rel="stylesheet" type="text/css" href="estilos.css" />
  </head>
  <body>
    ...
  </body>
```

## Agrupación de elementos con el atributo class

Se pueden definir estilos diferentes para un mismo tipo de etiqueta HTML, creando clases mediante el atributo class y luego indicando en CSS las propiedades de cada clase particular.

```
<p>Ropa:</p>
  <ul>
    <li><a href="camisetas.htm" class="ropa">Camisetas</a></li>
    <li><a href="pantalones.htm" class="ropa">Pantalones</a></li>
    <li><a href="camisas.htm" class="ropa">Camisas</a></li>
  </ul>

<p>Complementos:</p>
  <ul>
    <li><a href="sombreros.htm" class="complementos">Sombreros</a></li>
    <li><a href="collares.htm" class="complementos">Collares</a></li>
    <li><a href="pendientes.htm" class="complementos">Pendientes</a></li>
  </ul>
```

Para asignar las propiedades a cada clase, se utiliza la sintaxis css:

```
<style>
  a{color: blue;}
  a.ropa{color: green;}
  a.complementos{color: red;}
</style>
```

## Selectores

Los selectores sirven para identificar los elementos a los que se aplicará el formato definido por la lista de declaraciones.

### Selector universal

Se define con \* y se aplica a todos los elementos de la página:

```
* {
  margin: 0;
  font-family: Verdana;
}
```

## Selector de etiqueta HTML

Indica el estilo que se va a un elemento o etiqueta concreto.

```
h1, h2, h3, h4, h5 {  
    font-size: large;  
}
```

## Selector de clase

Desde el punto de vista de las CSS, podemos aplicar una regla a todos los elementos que sean de una determinada clase:

```
.cabecera {  
    color: blue;  
}
```

## Selector de identificador

Permiten aplicar un estilo único al único elemento de la página que tiene un determinado identificador (especificado con el atributo id).

```
p#destacado {  
    background-color:yellow;  
}
```

## Propiedades de texto

- color: indica el color de la fuente. Su valor es un color.
- font-size: tamaño de la fuente.
- font-style: para letras en cursiva: normal, italic.
- font-weight: para letras en negrita o letras delgadas: normal, bold, bolder, lighter...

- text-decoration: para añadir subrayados y tachados: none, underline, line-through, overline...
- text-transform: permite pasar a mayúsculas (uppercase) o a minúsculas (lowercase).
- letter-spacing y word-spacing: permiten indicar el espacio entre las letras y palabras.
- line-height: espacio entre las líneas.
- text-align: alineación horizontal del texto en un elemento que lo contiene: left, right, center, justify.
- text-indent: para añadir indentación a un bloque de texto. El valor es una unidad de medida.

## Propiedades de fondo

Para establecer el fondo de un elemento utilizamos la propiedad background-color.

Otra opción muy utilizada es desactivar la repetición con background-repeat: no-repeat; y establecer una posición fija de la imagen con background-position. Por ejemplo, para situarla en la esquina inferior derecha utilizaríamos: background-position: right bottom;. Si además queremos que la imagen esté fija y no desaparezca al hacer scroll en la página, entonces utilizamos background-attachment: fixed;. Juntándolo todo quedaría así (pruébalo por ti mismo):

```
body {
    background-image: url("fondo.png");
    background-repeat: no-repeat;
    background-position: right bottom;
    background-attachment: fixed;
}
```

## Dimensiones y bordes

Para indicar las dimensiones de un elemento utilizamos las propiedades width y height (ancho y alto).

Podemos asignar un margen externo para los cuatro lados con la propiedad `margin`. O también podemos indicar un margen externo particular para cada lado con `margin-top`, `margin-bottom`, `margin-left` y `margin-right`. De forma análoga, tenemos los márgenes internos `padding-top`, `padding-bottom`, `padding-left` y `padding-right`.

Ejemplo

```
#lateral {
    float:right;
    width:300px;
    background-color:#FFF599;
    padding:0px 20px 0px 20px;
}

#principal {
    float:left;
    width:600px;
    background-color:#BFCFFF;
    padding:0px 20px 0px 20px;
}
```

### Otras propiedades

- `visibility`: indica si el elemento está o no visible: `visible`, `hidden`, `collapse` (usado solo para elementos de tablas para no visualizar filas o columnas).
- `display`: cambia el tipo de caja del elemento: `block`, `inline` o `none`. Como veremos en la próxima sección, esto afectará a la forma en la que se posiciona un elemento en dentro de la página. El valor `none` suele utilizarse para hacer desaparecer elementos.