

| | | | |
|--|--|--|-------------------|
| | | | Clase 8 |
|--|--|--|-------------------|

Tópico

HTML+ CSS

Metas de comprensión

- ✓ Los alumnos comprenderán el concepto de maquetar un sitio web, las técnicas y los beneficios que esto trae al momento de diseñarlo
- ✓ Los alumnos comprenderán el uso de las listas y sus diferentes categorías como se implementan estas en el desarrollo web.
- ✓ Los alumnos comprenderán de qué manera se definen los estilos y cuál es la ventaja de su uso
- ✓ Los estudiantes comprenderán como HTML implementa CSS3 para lograr sitios de calidad profesional
- ✓ Los estudiantes comprenderán como se establecen las distintas formas de definir estilos, las diferentes unidades de medidas que se definen en CSS3, y como se realiza la lectura de varias propiedades a la vez.

Desempeño de exploración

Debatimos sobre la importancia de maquetar un sitio previamente, la importancia que tienen los estilos en el embellecimiento del sitio. Se analizan distintos sitios para visualizar e interpretar el uso de los estilos desde el uso profesional.

Desempeño de investigación

Mediante la codificación de los ejercicios propuestos se harán nuevas propuestas de diseño y maquetación, lo que llevara a los alumnos a investigar nuevas definiciones o tags ya sea en forma grupal o individual. A su vez indagaremos, debatiremos participando en clase, foros o con mensajería cual es el uso de estilos en la actualidad.

Introducción

¡Cada clase según su contenido puede tener un numero variable de páginas, a leer, no te asustes!! La materia esta lo más actualizado posible al año 2022, de manera tal que tengas todos los temas en forma completa, de allí su extensión.

Algunas recomendaciones que te pueden ayudar a la hora de comprender el material de estudio:

- ✓ Lee varias veces la clase si fuera necesario.
- ✓ Subraya, destaca o resume los conceptos que creas principales o de importancia en cada tema.
- ✓ Puedes grabar la lectura de la clase (no necesaria puedes ser tú, puede ser un familiar, amigo etc.) para poder escucharla luego en el colectivo, recreo, relax o fin de semana etc.
- ✓ Muchas veces los números o medidas (velocidades, tamaños, resoluciones, latencia, potencia, capacidad, etc.) no son tan importantes ya que la informática evoluciona día a día y esos son bastantes cambiantes en la guerra de las empresas y fabricantes para sobresalir en el mercado, por eso no es necesario que los memorices a todos a menos que el tutor te lo indique.
- ✓ Puedes realizar grafica con cuadros,
- ✓ Puedes ampliar tu conocimiento con investigaciones adiciones en la web o viendo videos en youtube que refuercen los conceptos
- ✓ Puedes consultarle a tu tutor por cualquiera de las vías indicadas en el campus por el temario si tienes alguna duda, consulta o inquietud.
- ✓ Trata de organizar tu tiempo para la lectura y la comprensión del material
- ✓ Este texto va a acompañar a todas las clases para recordarte como leer, estudiar y comprender el material de la materia
- ✓ Cada clase en el campus podrá estar acompañada de material adicional como profundizar los temas para la mejor interpretación de los mismos con videos, encuestas, foros, actividades individuales y/o grupales.
- ✓ Busca algún compañero de estudio para poder compartir conocimiento, apuntes y metodología de estudio
- ✓ No esperes hasta último momento para realizar tus consultas, leer o ponerte al tanto con la materia

En muchas ocasiones queremos darle un toque personal a nuestra web con diseños en diferentes colores, tamaños de fuente y estilos.

El diseño de una página o sitio web, necesita un lenguaje común para que los navegadores puedan interpretarlo y mostrarlo en forma correcta. Pero también para que otros diseñadores hagan las modificaciones necesarias para su optimización. El conocimiento a fondo de CSS es una habilidad fundamental para el desarrollador web.

Es muy común frustrarse cuando usas alguna propiedad y los elementos no se ven. ¿Te ha pasado alguna vez? ¡No te preocupes!

Te tenemos todo lo que necesites en esta clase y en las restantes para aprovechar todo tu potencial y el de CSS.



Es por eso que no perdemos más tiempo y comenzamos con la lectura de la clase.

Listas

¿Qué más puede tener una página web? Además de títulos, párrafos, imágenes, un sitio web puede tener una forma de organizar la información en la página y esto se puede hacer a través de las listas o las tablas. Básicamente cuando se habla de lista, hablamos de cómo organizar objetos lineales (generalmente textos). Lo podemos hacer de dos maneras: ordenadas o no.

Al crear una lista deberemos indicar su *inicio* y su *fin*. Para empezar una etiqueta para cada tipo distinto de lista, por ejemplo, para crear una lista no ordenada será necesario usar la etiqueta **UL** (su instrucción de inicio será ****). Una vez hemos *abierto* la lista deberemos insertar, como hemos dicho antes, cada uno de los elementos de lista. Para ello usaremos una nueva etiqueta (o dos en algún caso) que dependerá del tipo de lista. En el caso de la lista no ordenada los elementos se introducen con la etiqueta de una única instrucción **LI**, por ejemplo:

```
<UL>
<LI>Analista de Sistemas
<LI>Webmaster
<LI>Informática Administrativa
<LI>etc.
</UL>
```

Listas ordenadas

Las *listas ordenadas* nos permiten mostrar un conjunto de elementos o sucesos con un orden estricto. Por ejemplo, cuando hacemos una descripción con un orden cronológico, en donde no se puede alterar el orden de la descripción. Para incluir tales elementos, **HTML** nos ofrece la etiqueta **...**, como resultado podemos encontrar un conjunto de líneas perfectamente enumeradas. La etiqueta **** posee una serie de atributos: por ejemplo **TYPE** que nos permite definir el estilo de numeración (A,B,C, 1,2,3 etc.), con el atributo **START** definimos el valor inicial de la lista. De la misma manera la etiqueta **** posee dos atributos **TYPE** y **VALUE**, que cumplen la misma función que para la etiqueta ****.

Pongamos un ejemplo:

| | |
|--|--|
| <pre><p>Reglas de comportamiento en el trabajo</p> El jefe siempre tiene la razón En caso de duda aplicar regla 1 </pre> | Reglas de comportamiento en el trabajo |
| | <ol style="list-style-type: none">1. El jefe siempre tiene la razón2. En caso de duda aplicar regla 1 |

Del mismo modo que para las listas desordenadas, las listas ordenadas ofrecen la posibilidad de modificar el estilo como lo dijimos anteriormente. En concreto nos es posible especificar el tipo de numeración empleado eligiendo entre números (1, 2, 3...), letras (a, b, c...) y sus mayúsculas (A, B, C,...) y números romanos en sus versiones mayúsculas (I, II, III,...) y minúsculas (i, ii, iii,...).

Para realizar dicha selección hemos de utilizar, como para el caso precedente, el atributo **type**, el cual será situado dentro de la etiqueta ****. Los valores que puede tomar el atributo en este caso son:

- ✓ **1** Para ordenar por números
- ✓ **a** Por letras del alfabeto
- ✓ **A** Por letras mayúsculas del alfabeto

- ✓ **i** Ordenación por números romanos en minúsculas
- ✓ **I** Ordenación por números romanos en mayúsculas

Puede que en algún caso deseemos comenzar nuestra enumeración por un número o letra que no tiene por qué ser necesariamente el primero de todos. Para solventar esta situación, podemos utilizar un segundo atributo, **start**, que tendrá como valor un número. Este número, que por defecto es 1, corresponde al valor a partir del cual comenzamos a definir nuestra lista. Para el caso de las letras o los números romanos, el navegador se encarga de hacer la traducción del número a la letra correspondiente.

<p>Ordenamos por números</p>

```
<ol type="1">
<li>Elemento 1
<li> Elemento 2
</ol>
```

Ordenamos por números

1. Elemento 1
2. Elemento 2

<p>Ordenamos por letras</p>

```
<ol type="a">
<li>Elemento a
<li> Elemento b
</ol>
```

Ordenamos por letras

- a. Elemento a
- b. Elemento b

<p>Ordenamos por números romanos empezando por el 10</p>

```
<ol type="i" start="10">
<li>Elemento x
<li> Elemento xi
</ol>
```

Ordenamos por números romanos empezando por el 10

- x. Elemento x
- xi. Elemento xi

Listas no ordenadas

Una lista no ordenada nos muestra un conjunto de elementos relacionados entre sí, pero sin ningún tipo de orden. La etiqueta que utilizamos en este caso es ** ... **. La implementación de este tipo de lista es similar a la ordenada. También utilizamos la etiqueta **** para identificar cada elemento de la lista. La etiqueta **** también tiene su atributo **TYPE** que puede tomar uno de los siguientes valores: “*circle*” (círculo sin relleno), “*square*” (cuadrado), “*disc*” (círculo con relleno).

Hay que tener en cuenta que unas listas pueden anidar otra lista de diferente tipo, o sea que podemos combinar listas ordenadas y no.



Ejemplo:

<p>Países del mundo</p>

```
<ul>
<li>Argentina
<li>Perú
<li>Chile
</ul>
```

Países del mundo

- Argentina
- Perú
- Chile

Podemos definir el tipo de viñeta empleada para cada elemento. Para ello debemos especificarlo por medio del atributo **type** incluido dentro de la etiqueta de apertura ****, si queremos que el estilo sea válido para toda la lista, o dentro de la etiqueta **** si queremos hacerlo específico de un solo elemento. La sintaxis es del siguiente tipo:

<ul type="tipo de viñeta">

Donde tipo de viñeta puede ser uno de los siguientes:

- ✓ **circle**: marcas en forma de círculo (por defecto)
- ✓ **square**: marcas en forma de cuadrado
- ✓ **disk**: marcas en forma de disco
- ✓ **none**: no utilizar ningún tipo de marca

Vamos a ver un ejemplo de lista con un cuadrado en lugar de un redondel, y en el último elemento colocaremos un círculo. Para ello vamos a colocar el atributo **type** en la etiqueta ****, con lo que afectará a todos los elementos de la lista.

```
<ul type="square">
<li>Elemento 1
<li>Elemento 2
<li>Elemento 3
<li type="circle">Elemento 4
</ul>
```

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

Listas de definición

Son aquellas en donde podemos presentar el texto de diferentes formas. Utilizamos éste tipo de lista cuando anidamos listas. Anidar es insertar dentro de una lista otra. Esta puede ser de distintos tipos o también podemos incluir una definición.

Para incluir listas de definiciones utilizamos la etiqueta **<DL>** ... **</DL>**, ésta marca el comienzo y el fin de la lista.

Estas listas constan de dos elementos: términos y definiciones. Para incluir un término se usa la etiqueta **<DT>... </DT>**, y para incluir una definición la etiqueta **<DD>... </DD>**. Como se puede apreciar las etiquetas tienen cierre, y este tipo de lista se utiliza para definir un término con su descripción.

Por ejemplo, aquí vamos a crear un archivo listas.html en donde se describirán por intermedio de listas, los pasos para tener una página Web en un servidor gratuito (en nuestro ejemplo utilizamos diferentes tipos de listas utilizando anidación):

Ejemplo:

```
<HTML>
<HEAD><TITLE>LISTAS ...</title></HEAD>
<BODY>
<H2 align="center"> LISTAS: (Ordenadas - No Ordenadas - Especiales) Anidación...</H2>
<OL TYPE=A>
<LI>Crear la página web <br>
<OL TYPE=I>
<LI> Hacer el estudio preliminar
<LI> Organizar la información en su web.
<OL TYPE=i start=3>
<LI> Crear la página de inicio (defaul.html o indeSi.html)
<LI> Crear las páginas de enlaces.
```

```

</OL>
<LI value=5> Crear los diferentes html.
</OL>
<LI>Registrarse en un servidor gratuito:
<UL TYPE="circle">
  <LI>freesever.com
  <DL>
    <DT><i> CON DOMINIO PROPIO</i></DT>
    <DD> <B> Podemos definir un dominio propio empresa.com,
      para facilitar el ingreso a nuestra página. </b> </DD>
  </DL>
  <LI>websamba.com
  <DL>
    <DT><i> SIN DOMINIO PROPIO</i></DT>
    <DD> <B> No podemos definir un dominio propio empresa.com, y utilizamos subdominio
      ejemplo www.websamba.com/empresa </b> </DD>
  </DL>
</UL>
<LI>Comprobar que el servidor maneja todos los elementos insertados en nuestra web
<LI>Cargar la web con un programa de ftp (configurando la aplicación)
<LI>Controlar el correcto funcionamiento de la web en el servidor.
<LI><b><i> Guardar nombre de registro y claves del servidor web.</i></b>
<LI>Proceso terminado.
</OL>
</BODY>
</HTML>

```

```

<p>Diccionario de la Real Academia</p>
<dl>
  <dt>Brújula
  <dd> acá se escribe su definición
  <dt>Oreja
  <dd> acá se escribe su definición
</dl>

```

```

Diccionario de la Real Academia

Brújula
      acá se escribe su definición

Oreja
      acá se escribe su definición

```

Fíjate en que cada línea **<dd>** esta desplazada hacia la izquierda. Este tipo de etiquetas son usadas a menudo con el propósito de crear textos más o menos desplazados hacia la izquierda.

```

<dl>
  <dd>Primer nivel de desplazamiento
  <dl>
    <dd>Segundo nivel de
    desplazamiento
    <dl>
      <dd>Tercer nivel de
      desplazamiento
    </dl>
  </dl>

```

```

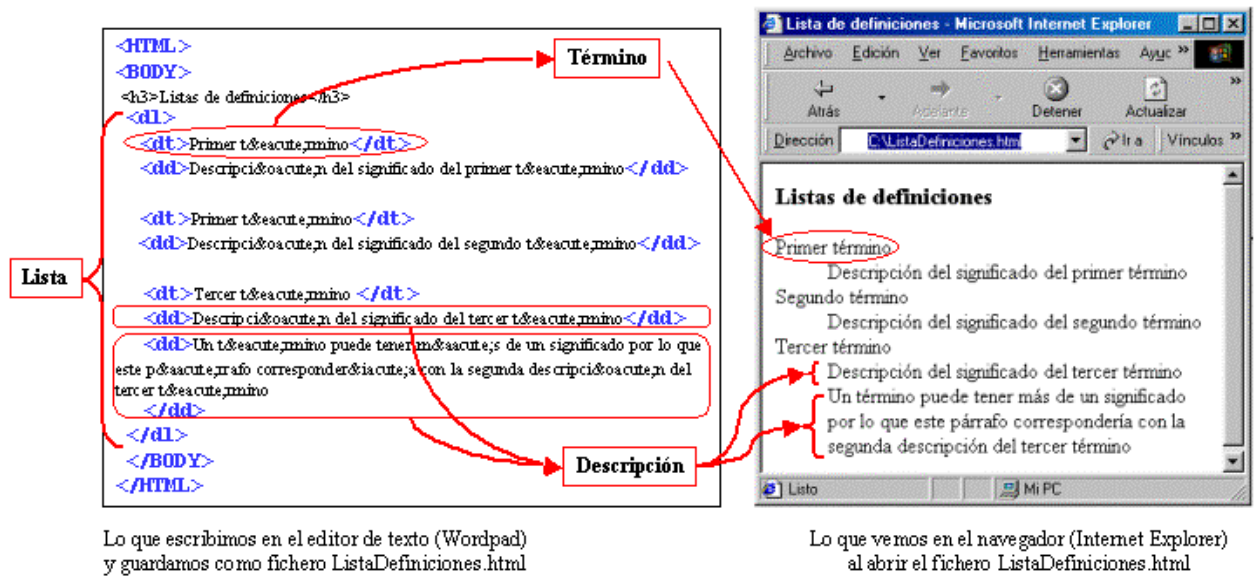
Primer nivel de desplazamiento

Segundo nivel de desplazamiento

Tercer nivel de desplazamiento

```

</dl>
</dl>



Anidando listas

Nada nos impide utilizar todas estas etiquetas de forma anidada como hemos visto en otros casos. De esta forma, podemos conseguir listas mixtas como, por ejemplo:

<p>Ciudades del mundo</p>

Argentina

Buenos Aires

Bariloche

Uruguay

Montevideo

Punta del Este

Ciudades del mundo

- Argentina
 1. Buenos Aires
 2. Bariloche
- Uruguay
 1. Montevideo
 2. Punta del Este

Mas ejemplo:

<HTML>

<HEAD>

<TITLE>Ejemplo Sobre Listas</TITLE>

</HEAD>

<BODY>

Una lista ordenada

Uno

Dos


```
<li>Tres</li>
```

```
</ol>
```

Una lista sin ordenar

```
<ul>
```

```
<li>Uno</li>
```

```
<li>Dos</li>
```

```
<li>Tres</li>
```

```
</ul>
```

Una lista de glosario

```
<dl>
```

```
<dt>T&eacute;mino 1</dt>
```

```
<dd>Definici&oacute;n 1</dd>
```

```
<dt>T&eacute;mino 2</dt>
```

```
<dd>Definici&oacute;n 2</dd>
```

```
</dl>
```

Listas anidadas

```
<ul>
```

```
<li>Uno
```

```
    <ul>
```

```
        <li>Uno</li>
```

```
        <li>Dos</li>
```

```
        <li>Tres</li>
```

```
    </ul>
```

```
</li>
```

```
<li>Dos</li>
```

```
<li>Tres</li>
```

```
</ul>
```

```
</BODY>
```

```
</HTML>
```

Ejemplo:

```
<ol type="I">
```

```
<li> Primer elemento romano
```

```
<ol type="I">
```

```
<li> Mejor en la lista interna
```

```
<li> Su segundo Valor
```

```
</ol>
```

```
<li> Seguimos con el segundo romano
```

```
<ol type="a">
```

```
<li> Empezamos lista con letra para terminar
```

```
</ol>
```

```
</ol>
```



¿Qué es meta charset?

Es un set de caracteres que reconoce la Pc de la misma manera que una calculadora identifica números. Cada uno de los caracteres se representa con un número conocido como punto de código y esto crea un canal de comunicación para codificar y decodificar contenido.

La computadora almacena los caracteres como uno o más bytes. Un ejemplo es el conjunto de caracteres *ASCII* que representa todos los caracteres en inglés y los caracteres de control especiales con números del 0 al 127.

Sin embargo, la mayoría de los charset solo funcionan para idiomas específicos y reconocen caracteres limitados, lo que dificulta o imposibilita la codificación. Sin embargo, en la actualidad, *Unicode* es el conjunto de caracteres más confiable y universalmente aceptado debido a su facilidad para traducir códigos y números. Puedes ver el meta charset en el encabezado de tu código html

`<meta charset="utf-8">`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mi página de prueba</title>
  </head>
  <body>
    <p>Esta es mi página</p>
  </body>
</html>
```

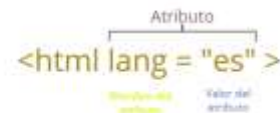
Sin la codificación de caracteres adecuada, un navegador mostrará texto basura porque simplemente no comprende lo que se le está añadiendo y tiene que hacer una suposición rápida y desinformada (es por eso que a veces no nos aparece la letra ñ o los acentos).



Es el código dominante de internet. **UTF-8** se utiliza en el 94% de los sitios web. Codifica los caracteres más comunes, números básicos e inglés con 8 bits. UTF-8 usa un mínimo de 1 byte. UTF-8 también es idéntico a ASCII para inglés. Esto significa que cualquier texto ASCII también es un texto UTF-8.

Lang

El atributo de HTML lang especifica el idioma principal del contenido de un documento; para establecer el idioma que se quiere usar, se debe agregar el atributo a la etiqueta <html>:



```
<html lang="es">
```

```
...
```

```
</html>
```

```
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <meta charset="utf-8">
5    </head>
6    <body>
7      Contenido del código del cuerpo
8    </body>
9  </html>
```

CSS

¿Qué es CSS?

CSS son las siglas de Cascading Style Sheets (en español, hojas de estilo en cascada), y es un lenguaje que define el estilo de la presentación de páginas de Internet. Se usa comúnmente en conjunto con HTML, donde la parte de HTML define la estructura de la página, y la parte de CSS define la presentación de la página, esto es, aspectos como colores, tipos de letra, márgenes, dimensiones, entre otros.

¿Para qué sirve?

CSS se utiliza para dar formato a documentos HTML y XML, separando el contenido de la presentación. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.



Comenzando a Definir Estilos

Una vez que aprendimos a crear la estructura en HTML5, haremos hincapié en el formato del sitio, para ello comenzaremos a estudiar CSS en su versión 3. A continuación te indico tres conceptos que debes manejar de ahora en más:

- ✓ **Atributos**: Son las palabras que usaremos para indicar cual estilo queremos modificar, por ejemplo, si queremos cambiar el tipo de letra, usamos el atributo "font", si es el fondo, el atributo "background", etc.
- ✓ **Valores**: Son para definir cómo vamos a modificar el atributo, o la propiedad que le daremos. Por ejemplo, si queremos que un tipo de letra sea rojo, usamos el atributo "font" y el valor "red".
- ✓ **Selectores**: Se usan para definir sobre cuales elementos HTML vamos a aplicar los estilos, si queremos definir un estilo para toda la página, debemos usar el selector "body" que se refiere a la etiqueta <body> del documento HTML.

Hay tres tipos de selectores:

- ✓ Los *selectores de etiquetas HTML*, se utilizan escribiendo el nombre de la etiqueta a la que le aplicaremos el estilo.
- ✓ Los *selectores de identificador*, se usan para aplicar estilos solo a las etiquetas identificadas con un nombre.
- ✓ El tercer *selector es el de clase*, se escribe en el documento CSS comenzando con un punto "." seguido del nombre que le queramos poner a la clase, de esta forma: .mi_clase.

Obviamente que veremos cada una de estas formas a lo largo de la materia. Comenzaremos por la parte más fácil incrementando el grado de dificultad, para ello te pido que desarrolles los ejemplos y ejercicios que se muestran en cada una de las clases, esto te ayudará a tomar destreza y experiencia en el código y sobre todo a la hora de depurar errores.





Veamos con mayor precisión la declaración de los estilos, los mismos están formados por:

- ✓ **Selector:** El selector le dice al navegador web que directiva o directivas de una página tienen que someterse al estilo.
- ✓ **Bloque de declaración:** El bloque comienza con una llave de apertura ({) y termina con una llave de cierre (}).
- ✓ **Declaración:** Entre las llaves de apertura y de cierre de la declaración, se añade una o más declaraciones o instrucciones de formato. Cada declaración tiene dos partes: una propiedad y un valor, y además acaban en un punto y coma.
- ✓ **Propiedad:** CSS ofrece un amplio rango de opciones de formato, que se llaman propiedades. Una propiedad es una palabra que indican un cierto estilo de efecto.
- ✓ **Valor:** Finalmente se puede expresar la creatividad asignando un valor a una propiedad CSS. Las diferentes propiedades CSS requieren tipos específicos de valores: un color, una longitud (como 16px, o 5em), o una clave específica (como top, center o bottom)

```
p {
  color:blue;
  font-size:20;
  font-weight:bold;
  font-family:Arial;
}
```

Podemos aplicar el mismo estilo en un conjunto de etiquetas. Para ello, indicamos las etiquetas seguidas por comas y luego, entre llaves, los atributos que queramos definir.

```
p, h1, h5 {
  color:#ffcc03;
  font-family:Arial;
}
```

Propiedades del Texto

font-family y se utiliza para indicar el tipo de letra con el que se muestra el texto. Mediante el nombre genérico de una familia tipográfica: los nombres genéricos no se refieren a ninguna fuente en concreto, sino que hacen referencia al estilo del tipo de letra. Las familias genéricas definidas son serif (tipo de letra similar a Times New Roman), sans-serif (tipo Arial), cursive (tipo Comic Sans), fantasy (tipo Impact) y monospace (tipo Courier New).

Ejemplos:

font-family:{ Georgia, "Times New Roman", Times, serif};

font-size. Además de todas las unidades de medida relativa y absoluta y el uso de porcentajes, CSS permite utilizar una serie de palabras clave para indicar el tamaño de letra del texto:

- ✓ **tamaño_absoluto:** indica el tamaño de letra de forma absoluta mediante alguna de las siguientes palabras clave: *xx-small*, *x-small*, *small*, *medium*, *large*, *xx-large*.
- ✓ **tamaño_relativo:** indica de forma relativa el tamaño de letra del texto mediante dos palabras clave (*larger*, *smaller*) que toman como referencia el tamaño de letra del elemento padre.

```

.parrafo1 {
  font-size: 1cm;
}
.parrafo2 {
  font-size: 20px;
}
.parrafo3 {
  font-size: 150%;
}
.parrafo4 {
  font-size: 2em;
}

.parrafo1 {
  font-family: "Palatino Linotype", "Book Antiqua", "arial";
}
.parrafo2 {
  font-family: Tahoma, Geneva, sans-serif, "Times New Roman",
Times, serif;
}
.parrafo3 {
  font-family: "Comic Sans MS", cursive;
}

```

| | | |
|-------|------|------|
| .5em | 50% | 8px |
| .7em | 70% | 12px |
| .9em | 90% | 14px |
| 1em | 100% | 18px |
| 1.5em | 150% | 24px |
| 2em | 200% | 36px |
| 3em | 300% | 48px |
| 3.5em | 350% | |

CSS recomienda indicar el tamaño del texto en la unidad *em* o en porcentaje (%). Además, es habitual indicar el tamaño del texto en puntos (*pt*) cuando el documento está específicamente diseñado para imprimirlo.

Por defecto los navegadores asignan los siguientes tamaños a los títulos de sección: `<h1>` = *xx-large*, `<h2>` = *x-large*, `<h3>` = *large*, `<h4>` = *medium*, `<h5>` = *small*, `<h6>` = *xx-small*.

Una vez indicado el tipo y el tamaño de letra, es habitual modificar otras características como su grosor (texto en negrita) y su estilo (texto en cursiva). La propiedad que controla la anchura de la letra es *font-weight*.

Los valores que normalmente se utilizan son *normal* (el valor por defecto) y *bold* para los textos en negrita. El valor *normal* equivale al valor numérico **400** y el valor *bold* al valor numérico **700**. Además de la anchura de la letra, CSS permite variar su estilo mediante la propiedad *font-style*.

Normalmente la propiedad *font-style* se emplea para mostrar un texto en cursiva mediante el valor *italic*.

```

.parrafo1 {
  font-style:normal;
}
.parrafo2 {
  font-style:italic;
}
.parrafo3 {
  font-style: oblique;
}
.parrafo4 {
  font-style: inherit;
}

```

La propiedad que define la alineación del texto se denomina *text-align*.

Los valores definidos por CSS permiten alinear el texto según los valores tradicionales: a la izquierda (*left*), a la derecha (*right*), centrado (*center*) y justificado (*justify*).

El interlineado de un texto se controla mediante la propiedad *line-height*, que permite controlar la altura ocupada por cada línea de texto.

```

p { line-height: 1.2; font-size: 1em; }
p { line-height: 1.2em; font-size: 1em; }
p { line-height: 120%; font-size: 1em; }

```

Diagram illustrating text alignment and line-height properties:

- `text-indent: 1em` (indicated by a horizontal arrow)
- `line-height: 800%` (indicated by a vertical arrow)
- `letter-spacing: 2px` (indicated by a horizontal arrow)

The diagram shows two lines of text: "línea 1" and "línea 2".

Además de la decoración que se puede aplicar a la tipografía que utilizan los textos, CSS define otros estilos y decoraciones para el texto en su conjunto. La propiedad que decora el texto se denomina ***text-decoration***.

El valor ***underline*** subraya el texto, por lo que puede confundir a los usuarios haciéndoles creer que se trata de un enlace. El valor ***overline*** añade una línea en la parte superior del texto, un aspecto que raramente es deseable. El valor ***line-through*** muestra el texto tachado con una línea continua, por lo que su uso tampoco es muy habitual. Por último, el valor ***blink*** muestra el texto parpadeante y se recomienda evitar su uso por las molestias que genera a la mayoría de usuarios.

Una de las propiedades de CSS más desconocidas y que puede ser de gran utilidad en algunas circunstancias es la propiedad ***text-transform***, que puede variar de forma sustancial el aspecto del texto.

La propiedad ***text-transform*** permite mostrar el texto original transformado en un texto completamente en mayúsculas (***uppercase***), en minúsculas (***lowercase***) o con la primera letra de cada palabra en mayúscula (***capitalize***).

```
.p1{
    text-transform:capitalize;
}
.p2{
    text-transform:lowercase;
}
.p3{
    text-transform:uppercase;
}
```

Uno de los principales problemas del diseño de documentos y páginas mediante CSS consiste en la alineación vertical en una misma línea de varios elementos diferentes como imágenes y texto. Para controlar esta alineación, CSS define la propiedad ***vertical-align***.

El valor por defecto es ***baseline*** y el valor más utilizado cuando se establece la propiedad ***vertical-align*** es ***middle***.

En muchas publicaciones impresas suele ser habitual tabular la primera línea de cada párrafo para facilitar su lectura. CSS permite controlar esta tabulación mediante la propiedad ***text-indent***.

CSS también permite controlar la separación entre las letras que forman las palabras y la separación entre las palabras que forman los textos. La propiedad que controla la separación entre letras se llama ***letter-spacing*** y la separación entre palabras se controla mediante ***word-spacing***.

Cuando se utiliza un valor numérico en las propiedades ***letter-spacing*** y ***word-spacing***, se interpreta como la separación adicional que se añade (si el valor es positivo) o se quita (si el valor es negativo) a la separación por defecto entre letras y palabras respectivamente.

The image shows two examples of the text 'letter-spacing'. The top example, labeled 'Sin espaciado', shows the letters of the word 'letter-spacing' tightly packed together. The bottom example, labeled 'Espaciado de 10 em', shows the same text with significant extra space between each letter, making it much more spread out.

CSS también permite controlar el tratamiento de los espacios en blanco de los textos mediante la propiedad ***white-space***.

El significado de cada uno de los valores de ***white-space*** es:

- ✓ ***normal***: comportamiento por defecto de HTML.
- ✓ ***pre***: se respetan los espacios en blanco y las nuevas líneas (exactamente igual que la etiqueta `<pre>`). Si la línea es muy larga, *se sale* del espacio asignado para ese contenido.
- ✓ ***nowrap***: elimina los espacios en blanco y las nuevas líneas. Si la línea es muy larga, *se sale* del espacio asignado para ese contenido.
- ✓ ***pre-wrap***: se respetan los espacios en blanco y las nuevas líneas, pero ajustando cada línea al espacio asignado para ese contenido.

- ✓ **pre-line**: elimina los espacios en blanco y respeta las nuevas líneas, pero ajustando cada línea al espacio asignado para ese contenido.

Por último, CSS define unos elementos especiales llamados "**pseudo-elementos**" que permiten aplicar estilos a ciertas partes de un texto. En concreto, CSS permite definir estilos especiales a la primera frase de un texto y a la primera letra de un texto.

El **pseudo-elemento** **:first-line** permite aplicar estilos a la primera línea de un texto. Las palabras que forman la primera línea de un texto dependen del espacio reservado para mostrar el texto o del tamaño de la ventana del navegador, por lo que CSS calcula de forma automática las palabras que forman la primera línea de texto en cada momento.

p:first-line { text-transform: uppercase; }

De la misma forma, CSS permite aplicar estilos a la primera letra del texto mediante el pseudo-elemento **:first-letter**.

Ejemplos:

p:first-letter { font-size: 2.5em; font-weight: bold; line-height: .9em; float: left; margin: .1em; }

Otros pseudo-elementos pueden ser:

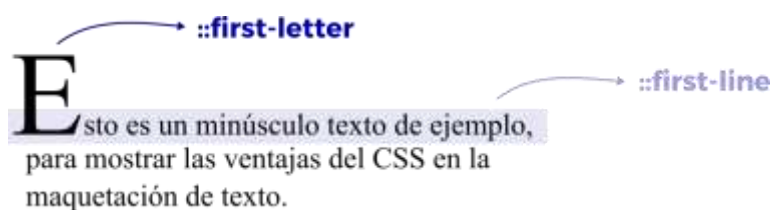
- ✓ **:before**. Nos posiciona al inicio del contenido de un elemento.
- ✓ **:after**. Nos posiciona al final del contenido de un elemento.



```
<head>
< style type="text/css">
p:first-line { color: #006699;}
p:first-letter {font-size: xx-large;}
</style>
</head>
```

```
< body>
<p>Este es un ejemplo para la clase 4 de DAI1.
Recuerda, no dejes de practicar los
ejemplos.<p>
</body>
```

Este es un ejemplo para la clase 4 de DAI1.
Recuerda, no dejes de practicar los ejemplos.



Propiedades para los hipervínculos

Es posible utilizar propiedades para definir distintos formatos para un hipervínculo dependiendo del estado en que se encuentre. Un hipervínculo se puede clasificar en cuatro estados:

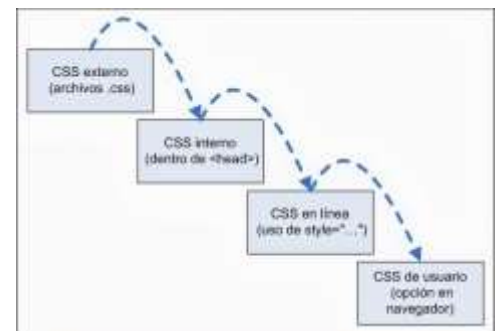
- ✓ **Visitado (visited):** El usuario ya visitó el enlace
- ✓ **Flotante (hover):** El usuario ha situado el curso sobre el enlace, pero no lo ha visitado
- ✓ **Activo (active):** El usuario pincha en el hipervínculo
- ✓ **Normal (link):** El usuario no interacciona de ninguna de las anteriores maneras con el hipervínculo

```
<html>
<head>
  <style type="text/css">
    a:link {color:#FF0000;}
    a:visited {color:#00FF00;}
    a:hover {color:#FF00FF;}
    a:active {color:#0000FF;}
  </style>
</head>
<body>
  <a href="http://www.unican.es">
    Universidad</a>
</body>
</html>
```

Formas de implementación CSS

Hay varias formas de implementar código CSS a un documento HTML:

1. **Hoja de estilos externa:** Esta es la forma más versátil y la única que recomiendo; un fichero externo contiene todo nuestro código CSS de manera que es completamente independiente el código propio de la página web de su código de formato. Esta última metodología es la que dejaremos para el final de todo.
2. **Hoja de estilos interna (la etiqueta style):** Esta forma es a veces utilizada; consiste en incluir el código CSS en la cabecera del sitio (etiquetas head del documento HTML), concretamente dentro de las etiquetas `<style>` y `</style>`



```
<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. **Estilos en línea (el atributo style):** Esta es, quizá, la forma menos adecuada, versátil y recomendable de todas ya que vincula directamente los estilos de formato con el código HTML. Su implementación se hace directamente en las etiquetas HTML con la propiedad `style`.

```
<html>
  <head>
    <title>Ejemplo</title>
  </head>
```

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



Una duda que puede surgir al hablar de estos tres tipos de implementación para CSS, en caso de usar los tres simultáneamente, es ¿cuál tiene más prioridad? La respuesta es que el navegador interpreta nuestro código (tanto HTML como CSS) linealmente de izquierda a derecha y de arriba a abajo (tal y como leemos nosotros); por lo que si lo primero que hacemos es incluir una hoja de estilos externa, luego implementamos una hoja de estilos interna y posteriormente le damos un estilo a una etiqueta HTML concreta, el estilo que visualizaremos será el de la etiqueta HTML por ser el último que el navegador a leído.

Estilo en línea

Una de las formas más simples e intuitivas de dotar de estilos al código HTML es usando el atributo *style* que admiten la mayoría de las etiquetas HTML.

Supón que queremos que el menú se muestre de color verde y el texto de los párrafos de color azul. Para aplicar un estilo utilizaremos esta sintaxis:

```
<nombreDeEtiqueta style = "propiedadCSS : valorEstablecido;" > ... </nombreDeEtiqueta>
```

En nuestro caso para un párrafo usaríamos `<p style = "color: blue;"> ... </p>`

Para aplicar el color verde a los elementos del menú, que están en una lista, podemos probar a aplicarle el atributo *style* y la propiedad *color* a la etiqueta ` ... `. El código quedaría como sigue:

```
<ul style = "color: green;">
  <li><a href="#" >Inicio</a></li>
  <li> <a href="libros.html">Libros de programación</a> </li>
  <li> <a href="cursos.html">Cursos de programación</a> </li>
  <li> <a href="humor.html">Humor informático</a> </li>
</ul>
```

Nos planteamos ahora dejar las viñetas en color rojo y el texto del menú en color verde, pero sin subrayado. Para ello tendremos que añadir una propiedad al link que elimine el estilo subrayado que por defecto incorpora el navegador. Esta propiedad será *text-decoration* cuyos posibles valores son *none* (ninguno), *underline* (subrayado), *overline* (línea superior), *line-through* (tachado). El código será el siguiente:

```
<ul style = "color: red;">
  <li><a href="#" style = "color: green; text-decoration: none;">Inicio</a></li>
  <li> <a href="libros.html" style = "color: green; text-decoration: none;">Libros de
programación</a> </li>
  <li> <a href="cursos.html" style = "color: green; text-decoration: none;">Cursos de
programación</a> </li>
  <li> <a href="humor.html" style = "color: green; text-decoration: none;">Humor
```

```
informático</a> </li>
</ul>
```

Y el resultado:



Otro ejemplo:

```
<p style="color:#990000"> Esto es un párrafo de color rojo.</p>
<p style="color:#000099">Esto es un párrafo de color azul. </p>
```

```
<HTML>
<HEAD>
<TITLE>Ejemplo de CSS - DIRECTOS</title>
</HEAD>
<BODY>
<H1 STYLE="font-family:Verdana,sans-serif;font-size:14pt;color:red"> Prueba de definición
de estilo directo en una etiqueta</H1>
</BODY>
</HTML>
```

Como se puede apreciar en este ejemplo solamente aplicamos el estilo sobre la etiqueta **<H1>**, en donde tenemos definido tipo de letra, tamaño y color para el texto que se encuentra encerrado dentro de esta etiqueta.

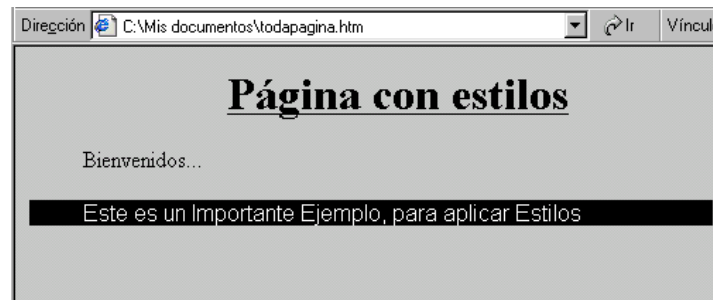
Estilo definido para toda una página

Podemos definir, en la cabecera del documento, estilos para que sean aplicados a toda la página. Es una manera muy cómoda de darle forma al documento y muy potente, ya que estos estilos serán seguidos en toda la página y nos ahorraremos así muchas etiquetas **HTML** que apliquen forma al documento. Además, si deseamos cambiar los estilos de la página lo haremos de una sola vez. Este ejemplo es más complicado, puesto que se utiliza la sintaxis CSS de manera más avanzada.

```
<html>
<title>Ejemplo de estilos para toda una página</title>
<STYLE type="text/css">
  H1 {text-decoration: underline; text-align:center;}
  P {font-Family:arial,verdana; color: white; background-color: black;}
  BODY {color:black; background-color: #cccccc; text-indent:10px;}
</STYLE>
<body>
<h1>Página con estilos</h1>
Bienvenidos...
```

```
<p>Este es un Importante Ejemplo, para aplicar Estilo</p>
</body>
</html>
```

En el ejemplo vemos que se utiliza la etiqueta **<STYLE>** colocada en la cabecera de la página para definir los distintos estilos del documento. A grandes rasgos, entre de **<STYLE>** y **</STYLE>**, se coloca el nombre de la etiqueta que queremos definir los estilos y entre llaves - {} - colocamos en sintaxis CSS las características de estilos.



Como se puede apreciar en el código, hemos definido que la etiqueta **<H1>** se presentará

- ✓ Subrayado (**text-decoration: underline**)
- ✓ Centrada (**text-align:center**)

También, por ejemplo, hemos definido que el cuerpo entero de la página (etiqueta **<BODY>**) se le aplique los estilos siguientes:

- ✓ Color del texto negro (**color:black**)
- ✓ Color del fondo grisáceo (**background-color: #cccccc**)
- ✓ Margen lateral de 1 centímetro (**text-indent:1cm**)

Cabe destacar que si aplicamos estilos a la etiqueta **<BODY>**, estos serán heredados por el resto de las etiquetas del documento. Esto es así siempre y cuando no se vuelvan a definir esos estilos en las siguientes etiquetas, en cuyo caso el estilo de la etiqueta más concreta será el que mande. Puede verse este detalle en la etiqueta **<P>**, que tiene definidos estilos que ya fueron definidos para **<BODY>**. Los estilos que se tienen en cuenta son los de la etiqueta **<P>**, que es más concreta.

Unidades de medida

Unidades Absolutas

Son medidas de un tamaño fijo, como, por ejemplo, un centímetro. Solo deben usarse cuando sepamos con certeza las dimensiones del medio de salida (monitor, impresora, etc) lo cual no siempre es posible. Las distintas unidades Absolutas que podemos usar son las siguientes:

- ✓ **in**: son pulgadas (en inglés "inches") y equivalen a 2.5 centímetros aproximadamente.
- ✓ **Cm y mm**
- ✓ **pt**: el Punto equivale a 0.014 pulgadas
- ✓ **pc**: Las picas, equivalen a 12 puntos.

Unidades absolutas



Unidades Relativas

Adoptan el valor relativo al tamaño del documento, del monitor, impresora, etc, según el caso y son las siguientes:

- ✓ **px:** la más utilizada es el Pixel.
- ✓ **ex:** se refiere a la altura de la "x"
- ✓ **em:** también llamada "cuadratín" es relativa al tamaño de letra definido por font-size.

Ahora bien, una vez que ya tenemos todos los conceptos principales vistos vamos a ver como se definen y en qué zona de la página se implementan los estilos con un ejemplo completo.

Unidades relativas



Selectores

Selector Universal

Este selector lo usamos cuando queremos que todos los elementos tengan un estilo definido.

Por ejemplo, si quisiéramos que todos los elementos que tengan texto o no tuvieran el mismo tipo de fuente usaríamos * para definir la fuente, y de esa forma todos los textos que tengamos en nuestro sitio tendrán la misma fuente.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

```
* { margin: 0; padding: 0;}
```

Etiqueta

Este es el más usado, y por Elemento me refiero a cualquier etiqueta HTML (span, div, img, etc). Si quisiéramos definir un estilo para todas las imágenes definiríamos un estilo `img{ }` para que afecte a las imágenes, un `p{ }` para los párrafos, y así con cada etiqueta que quisiéramos.

h1 { color: red;}

En vez de definir dos veces los mismos estilos para dos etiquetas diferentes podemos agruparlas usando un espacio entre cada nombre de la etiqueta, de esa forma ahorramos espacio y no tenemos que modificar estilos dos veces. Por ejemplo, si quisiéramos que los H1, H2, y H3 tuvieran el mismo color de texto lo lógico sería hacer un estilo para H1 {}, luego para H2 {}, y para H3 {}, pero para hacernos la vida más fácil podemos definir un sólo estilo que afecte a las tres etiquetas quedando como H1, H2, H3 {}

```

<html>
<head>
  <style type="text/css">
    p {background-color:#e0ffff;}
    body {background-image:
      url('paper.jpg');}
    h1 {color:#00ff00;}
    h1 {text-align:center;}
    p {text-indent:50px;}
  </style>
</head>
<body>
  <h1>Titulo</h1>
  <p>Primer parrafo.</p>
</body>
</html>

```

h1, h2, h3 {color: #8A8E27; font-weight: normal; font-family: Arial;}

```

p a { color: red; }

<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>

```

Selectores de Clases

También podemos usar el “.” para definir un estilo y que éste aplique a todas las etiquetas HTML que tengan la propiedad ‘class=’nombreClase’’. Este tipo de estilo se utiliza de manera tal que el programador pueda crear su propio estilo y aplicarlo cuantas veces quiera.

.aviso { padding: 0.5em; border: 1px solid #98be10; background: #f6feda;}

.error { color: #930; font-weight: bold;}

<h1 class="error">...</h1>

<p class="aviso">...</p>

Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de ID. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo *class*.

Los selectores de clase son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes.

```

8 <style type="text/css">
9 .DeClase{
10 color:aqua;
11 font-size:x-large
12 }
13 </style>
14 </head>
15
16 <body>
17 <p class="DeClase">Hola este es el parrafo numero 1</p>
18 </body>
19
20 </html>

```

Selectores ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso. El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página. La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.).

```
#destacado { color: red; }
```

```
<p>Primer párrafo</p>
```

```
<p id="destacado">Segundo párrafo</p>
```

```
<p>Tercer párrafo</p>
```

En una misma página, el valor del atributo id debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id. Sin embargo, el atributo class no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.

De esta forma, la recomendación general es la de utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes de la página HTML.

```
8 <style type="text/css">
9 #DeId{
10 color:darkcyan;
11 font-size:x-large
12 }
13 .Estilo
14 {
15     color:crimson;
16 }
17 </style>
18 </head>
19
20 <body>
21 <h1 class="Estilo">Titulo 1</h1>
22 <h3 class="Estilo" id="DeId">Titulo 3</h3>
23 <p class="Estilo">Hola este es el parrafo numero 1</p>
24
25 </body>
```

Selector descendiente

El selector descendiente se utiliza para encontrar objetos dentro de otros objetos. Por ejemplo, si quisiéramos darle un estilo a cualquier link dentro de algún párrafo, usaríamos lo siguiente:

```
p a {color:black;}
```

```
p, li {font-size:25px;}

ul li span {font-weight:bold;
            color:red;
            font-size:25px}

p span {color:green;}
```



```

1. article p {
2.   text-align: justify;
3. }
4. /* En esta regla utilizamos un selector de etiquetas para indicar que el texto dentro
   de la etiqueta HTML p ubicada dentro de un article se debe pintar justificado */
5. aside p {
6.   text-align: right;
7. }
8. /* En esta regla utilizamos un selector de etiquetas para indicar que el texto dentro
   de la etiqueta HTML p ubicada dentro de un aside se debe pintar alineado a la derecha
   */

```

Selector de hijos (>)

Este selector funciona de manera similar al selector descendiente, sin embargo, dista mucho en funcionamiento. La diferencia recae en que, el selector descendiente no tiene en cuenta si es que el objeto a seleccionar está directamente dentro del objeto padre, o si –de hecho– está dentro de un objeto, que está dentro de otro objeto, que está dentro del objeto padre. En el siguiente código, podemos ver que todos los links que estén dentro del h1, cambiarán a un color negro, incluyendo el que está dentro del elemento span:

```

<h1 >
  <a href="#" >
    <span >
      <a href="#" >
    </span >
  </a>
</h1 >

h1 a {color:#000;}

```

Sin embargo, en el código de ejemplo de más abajo, podemos ver el uso del selector de hijos. En este caso, el selector solo dará el color negro a los links que estén directamente relacionados con el h1. Es decir, el link que está dentro de un elemento span NO cambiará de color, pues no es un hijo directo de h1.

h1 > a {color:#000;}

```

ul > li {
  list-style: square;
}

```

Es realmente importante entender y practicar el buen uso de esta regla, pues puede ser de mucha utilidad para evitar la sobre-utilización de clases e ids.

| | | |
|---|--|---|
| <pre> <div> <p>Este párrafo es hijo en primer nivel de un div</p> </div> <div> <p>Este párrafo está dentro de un span por lo que no es hijo de primer nivel de un div</p> </div> </pre> | <pre> <style> div > p { background-color: #1258a3; color: #ffffff; font-size: 20px; } </style> </pre> | <div style="background-color: #1258a3; color: white; padding: 5px; text-align: center;"> Este párrafo es hijo en primer nivel de un div </div> <p>Este párrafo está dentro de un span por lo que no es hijo de primer nivel de un div</p> |
|---|--|---|

Selector adyacente +

Este selector indica al navegador que otorgue estilos al primer hermano que esté a continuación del elemento indicado. Su sintaxis es:

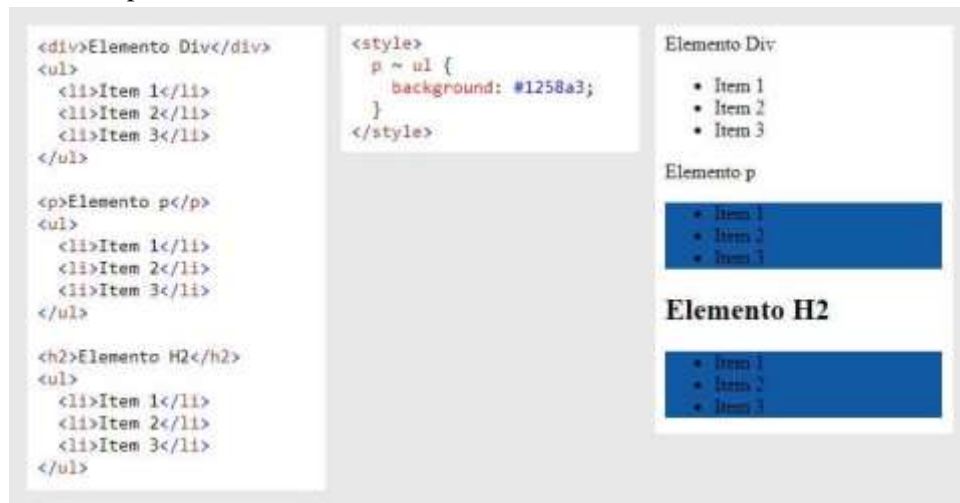
elemento1 + elemento2 { ... }

h1 + p {
color: coral;
font-size: 18px;
}



Selector de hermano predecesor ~

Este selector es útil para darle estilos a todos los elementos hermanos que vengan a continuación del predecesor indicado:



Veremos un ejercicio integrador para que veas la implementación de los estilos vistos en la clase:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Ejercicio</title>
<meta charset="utf-8" />
```

```
<style type="text/css">
```

```
* {margin:0; padding:0; border:0;}
```

```
html, body { width: auto; height: 100%; font-family: Verdana, Geneva, sans-serif;
font-weight: bold; background-color: #333; }
```

```
header{ position: relative; background-color: #9CC; color: #FFF; width: 98%;
height: 13%; margin: 0px auto; }
```

```
nav{ position: relative; background-color: #CC9; color: #FFF; width: 98%;
height: 13%; margin: 0px auto; }
```

```
section{ position:relative; float:left; background-color:#777; width:78%; height:55%; color:#FFF; }
```

```
article{ background-color: #0CC; width:98%; height:58%; color:#FFF;}
```

```
aside{ position:relative; background-color:#7A7; width:18%; height:55%; color:#FFF; float:left;}
```

```
footer{ position:relative; background-color:#AA7; width:98%;height:5%; color:#FFF;clear:both;}
```

```
header,section,footer,aside,article,nav{-webkit-border-radius: 10px; -moz-border-radius: 10px; border-radius: 10px; margin:1%;}
```

```
</style>
</head>
```

```
<body>
<header>
<h1>Cabecera</h1>
<p>Sitio web creado en html5</p>
</header>
<nav>
<h1>Zona de Menú</h1>
</nav>
<section>
<h2>Área de Sección</h2>
<article>
<h2>Descripción de un artículo en particular</h2>
<p> Contenido (además de imágenes, citas, videos etc.) </p>
</article>
</section>
<aside>
<h3>Zona Lateral</h3>
</aside>
<footer>Pie</footer>
</body>
</html>
```



Como veras en el ejemplo están codificadas todas las secciones de html5 y los estilos se han definido en la zona de *<Style>*, muchos atributos aun no los hemos visto, pero en la clase subsiguiente serán analizados.



Conclusión

CSS es la tecnología creada con el fin de separar la estructura de la presentación, y mejorar la usabilidad y experiencia de usuario de un sitio web, permitiendo el desarrollo web responsive y disminuyendo el procesamiento y el tiempo de carga, factores que pueden marcar la diferencia entre triunfar o morir en el intento. Es por eso que podemos concluir después de esta clase que:

- ✓ Las CSS constituyen una valiosa herramienta de diseño
- ✓ Facilitan la tarea de edición y mantenimiento de los sitios en la Web
- ✓ Se complementan con otros lenguajes para lograr una mejor comunicación

Hoy en día encontrarás CSS en todas las páginas web, ya que es un requisito tan importante como el propio lenguaje de marcado, ¡no dejes de implementarlo en tu sitio web!

¡Nos vemos en la próxima clase, y recuerda mientras más codifiques mejor comprenderás el lenguaje!



Ejercitación

1. Codifica los siguientes ejemplos, estos te permitirán adquirir habilidad en maquetado.

Ejercicio 1

1. Argentina
 - o [La Nación](#)
 - o [Clarín](#)
 - o [Página 12](#)
2. España
 - o [El País Digital](#)
 - o [ABC](#)
 - o [El Mundo](#)
3. México
 - o [La Jornada](#)
 - o [El Universal](#)

Ejercicio 2

PASTEL

Ingredientes

- 100 g de harina
- 10 g de azúcar
- 1 taza de agua
- 2 huevos
- sal y pimienta al gusto

Pasos a seguir

1. Mezcle bien los ingredientes secos.
2. Vierta los ingredientes líquidos.
3. Remueva durante 10 minutos.
4. Hornear durante una hora a 300 grados.

Notas

Puede añadir pasas y almendras picadas para mejorar la receta.

Ejercicio 3

Jugadores de fútbol de leyenda

- España
 - Casillas
 - Xavi
 - Iniesta
- Francia
 - Zidane
 - Henry
 - Barthez
- Brasil
 - o Ronaldo
 - o Pelé
 - o Ronaldinho

Ejercicio 5

1. Andalucía
 - a. Córdoba
 - I. Santaella
 - II. La Carlota
 - b. Huelva
 - I. Ayamonte
 - II. Isla Cristina
2. País Vasco
 - a. Vizcaya
 - I. Bilbao
 - II. Durango
 - b. Guipúzcoa
 - I. San Sebastián
 - II. Irún

Ejercicio 4

- Inicio
- Sobre mí
- servicios
 - o Fontanería
 - o Electricidad
 - o Pintura
 - Parkings
 - Trabajos en altura
 - Domicilios privados
- Testimonios
- Contacto

Ejercicio 6

Curriculum Vitae de Bruce Wayne

Datos personales

- Nombre completo: Bruce Wayne
- Fecha de nacimiento: 1-5-1939
- Lugar de nacimiento: Gotham City

Formación académica

- 1956-1961: Universidad del Esquapajaro
- 1952-1956: Instituto de Dos Caras
- 1944-1952: Escuela Primaria del Joker

Experiencia laboral

- 1975-1985: En el paro
- 1965-1975: Cazavillanos y demás cosas
- 1962-1965: Aprendiz de superhéroe

Ejercicio 7

Internet

Contenido

1. Principales protocolos
2. Formas de conectarse

1. Principales protocolos

- Transmisión de ficheros:
 - FTP
- Correo electrónico:
 - IMAP
 - POP
 - SMTP

2. Formas de conectarse

Red Telefónica Conmutada (RTC)

La línea telefónica de toda la vida. Para acceder a Internet es necesario un módem.

Red Digital de Servicios Integrados (RDSI)

Una línea telefónica especial. Para acceder a Internet es necesario un módem RDSI.

Línea de Abonado Digital Asimétrica (ADSL)

Se basa en la conversión de una línea RTC en una línea de alta velocidad. Para acceder a Internet es necesario un módem ADSL.

Fibra Óptica

Una línea de fibra óptica. Normalmente la fibra óptica no llega hasta el usuario final, por lo que el término más apropiado es Fibra híbrida coaxial.

Comprar el libro *Portada del libro 'HTML & CSS: Curso práctico avanzado'* por Sergio Luján Mora HTML & CSS: Curso práctico avanzado en el sitio web de Publicaciones Altaria

2. Aplicar CSS a párrafos, títulos, listas y links, implementa selectores

Ejercicio 1



Ejercicio 2



Ejercicio 3

Como en la hoja de estilo hemos dicho que el body fuera con fondo blanco, con un texto de tamaño 10 y centrado, ésta será la apariencia principal de la página.

Aquí tenemos un párrafo. Como también le hemos aplicado un estilo, podemos ver el texto del párrafo de color azul y con un tamaño ligeramente superior.

A este párrafo le hemos aplicado el "estilo1", por lo que el texto tiene un fondo negro, un texto en blanco y un tamaño de 15 px.

Este texto posee unas características propias de la clase 1. No es que sea un estilo muy bonito, pero por lo menos se entiende bien, ¿no??

Y aquí tenemos un último estilo.

Pues estos han sido todos los estilos que teníamos preparados para ti. Este tiene otra vez el estilo predeterminado para body.

Ejercicio 4

Datos del libro

- Título: HTML & CSS: Curso práctico avanzado
- Autor: Sergio Lojón Mora
- Editorial: Publicaciones Altaria
- Año de publicación: 2015
- ISBN: 978-84-944049-4-8

Descripción del libro

Aunque los inicios de **Internet** se remontan a los años sesenta, no ha sido hasta los años noventa cuando, gracias a la **Web**, se ha extendido su uso por todo el mundo. En pocos años, la **Web** ha evolucionado enormemente: se ha pasado de páginas sencillas, con pocas imágenes y contenidos estáticos que eran visitadas por unos pocos usuarios a páginas complejas, con contenidos dinámicos que provienen de bases de datos y que son visitadas por miles de usuarios al mismo tiempo.

Todas las páginas están internamente construidas con la misma tecnología, con el Lenguaje de marcado de hipertexto (**Hypertext Markup Language, HTML**) y con las Hojas de estilo en cascada (**Cascading Style Sheets, CSS**).

Este libro es adecuado para cualquiera que tenga interés en aprender a desarrollar sus propias páginas web. No son necesarios conocimientos previos para aprender con este libro, lo único que es necesario es saber utilizar un ordenador y saber navegar por la **Web**.

Contenido del libro

El contenido de este libro se estructura en tres apartados bien diferenciados:

- En la primera parte del libro se trabajan conceptos generales que son necesarios para poder desarrollar páginas web; se explican conceptos de estructura física y estructura lógica (o estructura de navegación) de un sitio web. Se detalla cómo influye la estructura física en las **URL** o direcciones que se emplean a la hora de crear los enlaces de un sitio web. Pasando por el concepto de "estándar web", un término general que se emplea para referirse a los estándares que definen su funcionamiento como **HTML** y **CSS**, empleados para el desarrollo de las páginas web en el lado del cliente.
- En la segunda parte se trabaja **HTML**. Partiendo de la estructura básica de una página web, se explican las etiquetas de **HTML** que se utilizan para definir el texto, los enlaces, las listas, las tablas, los formularios y los elementos multimedia.
- En la tercera y última parte se explica **CSS**, el lenguaje que se emplea para definir el formato y la presentación de una página web. Se explica cómo utilizar el color, cómo definir la presentación del texto, de las tablas y de los formularios; cómo realizar transformaciones y transiciones con el fin de diseñar una página web.



Autoevaluación

En base a los conocimientos adquiridos, con sus propias palabras responde las siguientes preguntas:

1. ¿Cuál es la función de crear una lista en una página web?
2. Lista los atributos de las listas
3. ¿Cuál es el uso que se le pueden dar a las listas?
4. ¿Cuántos tipos de UTF existen en la actualidad?
5. ¿Qué otras unidades relativas existen?
6. Indica la diferencia de los selectores de clases y los de identificación
7. ¿Para que se utiliza el selector nth-?
8. ¿Cuál es la diferencia entre pseudo-clase y pseudo-elemento?