

HOJAS DE ESTILO. CSS

© Luis Pilo Aceituno

luispiloaceituno@gmail.com

1.INTRODUCCIÓN

CSS no es un lenguaje de marcas sino un lenguaje de presentación por ello su estructura y sintaxis es muy distinta a la de HTML. No obstante CSS está muy ligado a los lenguajes de marcas

El objetivo de CSS es manejar el aspecto y formato de los documentos. liberando al HTML de la tarea de la presentación.

Esta separación entre contenidos y presentación supone muchas ventajas por ejemplo evita repetir código.

```
<body>
<h1><font face="Arial" size="5">UNIDAD 2. HTML y XHTML</font></h1>
<h2><font face="Verdana" size="4">2.5. Contenido del cuerpo</font>
</h2>
<h3><font face="Garamond" size="3">2.5.1. Manejo del texto</font>
</h3>
<h3><font face="Garamond" size="3">2.5.2. Listas</font></h3>
<h2><font face="Verdana" size="4">2.7. HTML 5</font></h2>
<h3><font face="Garamond" size="3">2.7.1. Elementos de sección
</font></h3>
</body>
```

Que se puede mejorar con este otro:

```
<html>
<head>
  <style type="text/css">
    h1 {font-family: Arial;}
    h2 {font-family: Verdana;}
    h3 {font-family: Garamond;}
  </style>
</head>
<body>
  <h1>UNIDAD 2. HTML y XHTML</h1>
  <h2>2.5. Contenido del cuerpo</h2>
  <h3>2.5.1. Manejo del texto</h3>
  <h3>2.5.2. Enlaces</h3>
  <h2>2.7. HTML 5</h2>
  <h3>2.7.1. Elementos de sección</h3>
</body>
</html>
```

- En el primer fragmento de código cada vez que introducimos una etiqueta hi debemos repetir su estilos, con la segunda opción no es necesario más que una vez (en la etiqueta style)

Ventajas en el uso del CSS

- Reduce el código a escribir
- Facilita la generación de código y su mantenimiento
- Mejora la legibilidad de los documentos.

En 1961 surgió la primera versión de CSS llamada CCS nivel I. El estándar oficial de CSS nivel 1 se publicó en 1998. Este estándar ha sufrido una revisión llamada CSS 2.1 que es la especificación actual. Desde 1998 hasta hoy W3C está trabajando para establecer la versión CSS 3.

Todos los navegadores tienen soporte amplio para CSS 2.1 pero no tan amplio para CSS 3. Para comprobar la compatibilidad de nuestro navegador con las versiones de CSS consultar en <http://www.css3.info/selectors-test/>

2. SINTAXIS

Disponemos de tres formas de introducir código CSS en nuestros documentos HTML

1. CSS en línea
2. CSS interno
3. CSS externo

1.CSS en línea

Se utiliza para especificar el estilo de una línea en particular dándole, por ejemplo, estilos a una etiqueta de la línea. Usamos el atributo style de la etiqueta a la que queremos darle estilos.

La sintaxis sería `<etiqueta style=""propiedad:valor; propiedad:valor;> texto </etiqueta>`

```
<body>
  <p style="font-family: Verdana; font-size: medium;">HOLA MUNDO</p>
</body>
```

2.-CSS interno

Se emplea para indicar los estilos de la página donde se definen. Para establecer los estilos se colocan entre las etiquetas `<style>` `</style>`

```

<html>
  <head>
    <style type="text/css">
      body {font-family: Courier New;}
      h1 {font-family: Arial; font-size: x-large;}
      p {font-family: Verdana; font-size: medium;}
    </style>
  </head>
  <body>
    <h1>Tipo de fuente Arial y tamaño grande</h1>
    <p>Tipo de fuente Verdana y tamaño medio</p>
  </body>
</html>

```

3.CSS externo

Se crea un archivo con los estilos. Una posible denominación para ese archivo es “**estilos.css**” en una carpeta, que se aconseja tenga como nombre /css. Después se inserta en cada documento mediante el enlace <link>. Se emplea para indicar los estilos globales que se van a compartir con todas las páginas web de un sitio

La etiqueta <link> tiene los siguientes atributos: type, href y rel. Sus valores son

<link rel="stylesheet" type="text/css" href="/css/estilos.css">

- En caso de colisión la prioridad de las reglas aumenta cuanto más particular es.
 1. css en línea
 2. css interno
 3. css externo
- Dentro de cada una de estas formas en caso de repetición es la última regla la que se aplica. En el siguiente ejemplo el tipo de fuente que se aplicará es Verdana

```

p {font-family: Arial;
   font-family: Verdana;}

```


ACTIVIDAD 1: Realiza los ejercicios nº 1, 2 y 3.

3. SELECTORES. TIPOS.

Cada regla CSS sirve para definir un estilo y consiste en la unión de un selector más una declaración entre llaves. Cada declaración se compone de una o más parejas de propiedades con su valor terminadas en punto y coma.

Regla CSS

```
selector { propiedad: valor; propiedad: valor; ...; }
```



```
p {font-family: Verdana; font-size: medium;}
```

- El selector especifica los elementos sobre los que se define un estilo y la propiedad es una característica del selector
- El listado de propiedades y valores posibles se encuentra en la especificación oficial del W3C: <http://www.w3.org/TR/CSS2/propidx.html>

Las reglas pueden agruparse en caso de compartir la declaración o el selector

```
h1 {font-family: Verdana; color: red;}  
h2 {font-family: Verdana; color: red;}
```

Los comentarios se establecen igual que en el C. /*comentarios*/

3.1.-Selectores

1. En CSS 2 tenemos los siguientes tipos de selectores.
2. Selector universal
3. Selector de tipo
4. Selector descendente
5. Selector hijo
6. Selector adyacente
7. Selector de atributos
8. Selector de clase
9. Selector de id
10. Pseudo-clase
11. Pseudo-elemento

1.Selector universal

Sirve para seleccionar todos los elementos de la página. Se indica mediante el símbolo asterisco * { selector:propiedad; selector:propiedad;}

Ejemplo:

```
* { margin: 0px }
```

2.-Selector de tipo

Selecciona todos los elementos de la página cuya etiqueta coincida con el selector.

etiqueta {propiedad:valor; propiedad:valor;}

Ejemplo:

```
p {font-family: Verdana; color: red;}
```

3.-Selector descendente

Decimos que D es un elemento descendente de E, si D está contenido entre las etiquetas de apertura y cierre de E

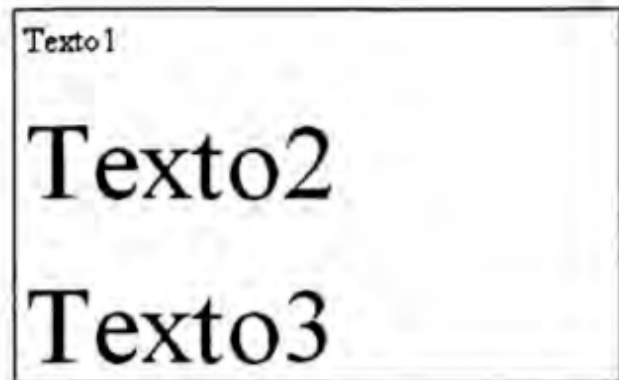
Un selector de la forma ***E M {declaración}*** sirve para seleccionar todos los elementos M que son descendientes de E a cualquier nivel

p a {font-size: 50px}

Aplicada al código:

```
<p>Textol</p>
<p><a>Texto2</a></p>
<p><span class="nieto"><a>Texto3</a></span></p>
```

Visualizaría el Texto2 y el Texto3 a tamaño 50.



Observaciones

- Si queremos excluir de la selección a los hijos y seleccionar a partir de los nietos en adelante se combina con el selector universal

etiqueta padre * etiqueta nieto {declaración}

Ejemplo: Asignamos estilos a todos los enlaces que estén contenidos en una etiqueta que a su vez está contenida dentro de una etiqueta de párrafo. Si está directamente contenida en una etiqueta de párrafo (etiqueta hijo) no se le aplica el estilo

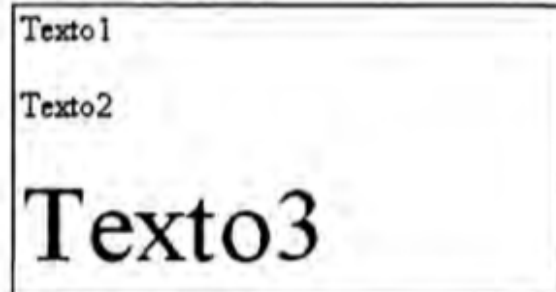
Aplicada al siguiente texto

```
<p>Textol</p>
<p><a>Texto2</a></p>
<p><span class="nieto"><a>Texto3</a></span></p>
```

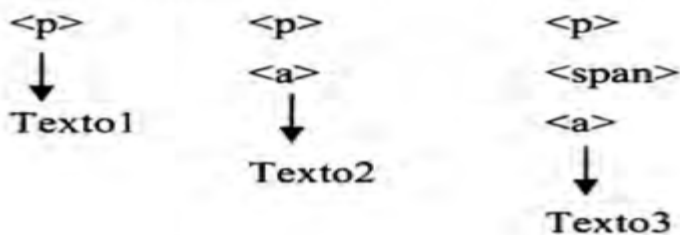
La declaración

p * a {font-size: 50px}

Visualizaría solamente el Texto3 a tamaño 50.



El documento HTML genera una estructura de árbol y podemos hablar de elementos padre, hijo, nieto, ..



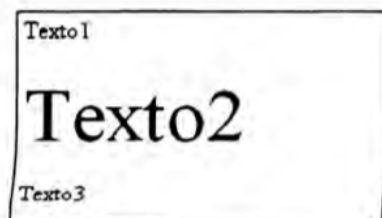
4.-Selector hijo

Para seleccionar al primer descendiente de un elemento (el hijo) excluyendo al resto usamos el signo los selectores hijos, que se construyen con >

etiqueta padre > etiqueta hijo { declaración}

p > a {font-size: 50px}

Visualizaría solamente el Texto2 a tamaño 50, ya que Texto3 sería descendiente de 2º nivel, pero no sería hijo (descendiente de 1º nivel).



OBSERVACIONES

1. Para seleccionar todas las etiquetas hijas de otra (sin distinguir si son hijos directos o no) es decir: para seleccionar todas las etiquetas descendientes de otra usamos el selector descendente

selector padre selector hijo {declaración;}

2. Para seleccionar todas las etiquetas descendientes de un descendiente directo de otra: Es decir para dar estilos a los nietos debemos combinar un selector dado con el selector universal

selector padre * selector nieto {declaración;}

3. Para seleccionar los descendientes directos de una etiqueta: es decir para dar estilos a los hijos directos usaremos la etiqueta

selector padre > hijo {declaración;}

5. Selector adyacente.

Se emplea para seleccionar elementos que son hermanos (tienen el mismo padre) y son adyacentes (consecutivos) en el código.

E+M {declaración;} , selecciona a todos los elementos de tipo M tales que E y M tienen el mismo padre y E precede a M en el código ,salvo los comentarios

6.-Selector de atributos

Permite seleccionar de 4 formas distintas.

- [atributo]: Selecciona los elementos que tengan ese atributo independientemente del valor que tome
- [atributo=valor]: Selecciona los elementos que tengan el atributo con el valor especificado

7.-Selector de clase

Selecciona una instancia en particular de un elemento. Usamos la sintaxis

selector.nombre de la clase {declaración de estilos}

Ejemplo.

<h3 class ="grandote"> Texto 1 </h3>

<p class="grandote"> Texto 2 </p>

<p class="pequeño" >Texto 3</p>

El siguientes estilo `p.grandore{font-size: 50px}` muestra Texto 2 en grande

8.- Selector de ID

Sirve para seleccionar a un elemento determinado, permite elegir entre varias etiquetas del mismo tipo a cual le damos estilos

La sintaxis es : ***etiqueta#nombre_del_id{declaración de estilos;***

Ejemplo:

Supongamos que el código html tiene una declaración como: <p id="mio" >Texto </p> puedo dar estilos a Texto de la siguiente forma. `p#mio {estilos;}`

OBSERVACIÓN:

- Tanto en los selectores de clase como en los de id se puede omitir el nombre de la clase. Las siguientes declaraciones de estilos son válidas

.nombre_clase

#nombre_id

ACTIVIDAD 2: Prueba todos los sectores vistos hasta ahora, en un nuevo documento html

- **Emplea CSS interno (etiqueta <style> en el head)**
- **Emplea CC externo (archivo css externo)**

9.-Pseudo-clases

Damos estilos según la acción realizada por el usuario (evento). Algunas pseudo-clases son:

1. `:link` .Se aplica a los enlaces no visitados
2. `:visited`. Se aplica a los enlaces visitados al menos una vez.
3. `:hover`. Se aplica a un elemento seleccionado por el usuario pero sin activarlo, por ejemplo al situar el ratón sobre un objeto sin hacer clic

4. `:active`. Se aplica a los elementos que están siendo activados, por ejemplo mientras un usuario presiona el botón izquierdo del ratón
5. `:focus`. Se aplica al elemento que tiene el foco, por ejemplo un cuadro de entrada de un formulario.

Algunos ejemplos de pseudo-clases serían los siguientes:

<code>a:link {color: black;}</code>	Enlaces sin visitar -> negro
<code>a:visited {color: blue;}</code>	Enlaces visitados -> azul
<code>a:hover {color: green;}</code>	Con el ratón encima -> verde
<code>a:active {color: red;}</code>	Presionando el botón -> rojo
<code>a:focus {background-color: yellow;}</code>	Con el foco -> fondo amarillo

OBSERVACIÓN: Si usas, durante las pruebas “#” para los enlaces la pseudoclase `:link` falla. Debes darme una dirección aunque sea falsa.

ACTIVIDAD 3. Crea una tabla con 4 enlaces y utiliza las tres primeras pseudo-clase, de modo que los enlaces aparezcan en negro al principio, en azul al ser visitados y en verde con el ratón encima

10.Pseudo-elementos:

Algunos pseudo-elementos de interés son

1. `:first-line`. Se aplica el estilo a la primera línea de un texto
2. `:first-letter`. Se aplica el estilo a la primera letra de un texto
3. `:before`. Sirve para generar texto antes del contenido de un elemento
4. `:after`. Sirve para generar texto después del contenido de un elemento

Ejemplo

```

<html>
<head>
  <style>
    h1:before {content: "Encabezado --- "; }
    h1:after {content: " --- Fin de encabezado"; }
    p:first-line {font-size: 20px; }
    p:first-letter {font-size: 3em; }
  </style>
</head>
<body>
  <h1>Textol</h1>
  <p>
    Esta es la primera línea <br>
    Esta es la segunda línea <br>
    Esta es la tercera línea <br>
    Esta es la cuarta línea <br>
  </p>
</body>
</html>

```



ACTIVIDAD 4: Modifica el código anterior y experimenta con los Pseudo-elementos.

OBSERVACIÓN

CSS 3 incorpora otros selectores pero no todos los navegadores ofrecen soporte para ellos. Por esta razón no se verán en este curso

4. MODELO DE CAJAS

El modelo de cajas es el elemento fundamental de CSS porque determina la composición de la página web. Todos los elementos que se insertan en el documento HTML se representan automáticamente mediante cajas rectangulares (excepto head)

En principio son cajas invisibles salvo que se les asignen bordes o colores de fondo.

Las partes que componen una cajas son:

1. **Contenido** (content): texto, imágenes, videos, listas, etc. del elemento
2. **Relleno** (padding): espacio entre el contenido y el borde
3. **Borde** (border): línea que encierra completamente el contenido y si relleno
4. **Imagen** de fondo (background-image): imagen que se muestra por detrás del contenido y el espacio de relleno.
5. **Color de fondo** (background-color): color que se muestra por detrás del contenido y el espacio de relleno.
6. **Margen** (margin). Espacio libre entre la caja y las cajas adyacentes

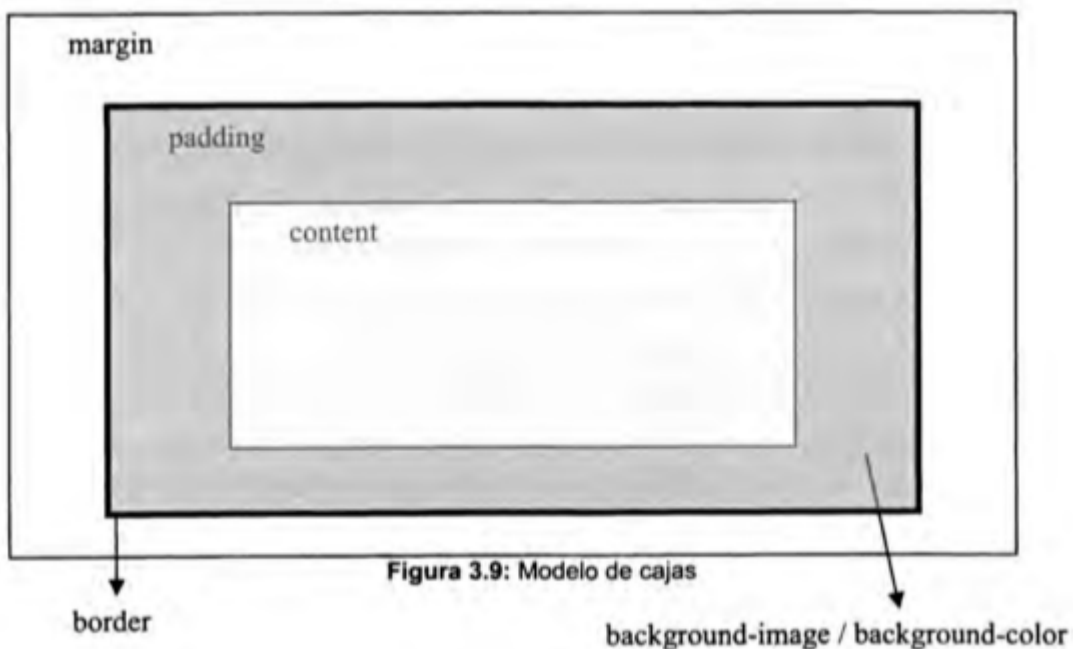
Padding y margin son transparentes

- En el espacio ocupado por padding se muestra el color o imagen de fondo
- En ese espacio ocupado por margin se muestra el color o imagen de fondo de su elemento padre.

Si ningún padre tiene definido un color o imagen de fondo se muestra el color o imagen de fondo de la página

Si una caja define tanto un color como una imagen de fondo, la imagen tiene mas prioridad y se visualiza antes que el color

Si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, entonces se visualiza el color de fondo en dichas zonas



4.1.-Propiedades de las cajas

A) ANCHO Y ALTO

Las propiedades de la caja que controlan el alto y ancho son: width y height

Propiedades: **width/height**

Significado: **ancho/alto de la caja**

Valores: **<númeroico px> | <númeroico %> | inherit | auto**

Aplicación: **elementos de bloque, imágenes y columnas de tabla**

<númeroico px>: establece un ancho/alto fijo de la caja en píxeles.

<númeroico %>: establece un ancho/alto fijo de la caja según un porcentaje del ancho/alto del elemento padre.

<inherit>: establece un ancho/alto fijo heredado del elemento padre.

<auto>: es el valor por defecto y determina que el navegador debe calcular el ancho/alto teniendo en cuenta el espacio disponible en la página.

Ejemplo:

```
#menu {width: 180px; height: 600px;}
```

```
<div id="menu">
```

```
...
```

```
</div>
```

B) MARGIN, PADDING

Propiedades: **margin, margin-top, margin-right, margin-bottom, margin-left**

Significado: **margen, margen superior, derecho, inferior e izquierdo**

Valores: **<número px, em> | <número %> | inherit | auto**

Aplicación: **a todos los elementos** (excepto margin-top y margin-bottom)

Margin es una propiedad 'shorthand' que equivale a las otras cuatro a la vez.

El margen permite el centrado horizontal de una caja mediante `→ margin: 0 auto;` que significa margen superior e inferior 0 y margen derecho e izquierdo automático.

Ejemplo:

```
#body {margin-top: 2em;}  
#caja1 {margin: 10px;}  
#centrado {margin: 0 auto;}
```

Propiedades: **padding, padding-top, padding-right, padding-bottom, padding-left**

Significado: **relleno, relleno superior, derecho, inferior e izquierdo**

Valores: **<número px, em> | <número %> | inherit | auto**

Aplicación: **a todos los elementos excepto cabeceras y pies de tablas**

Ejemplo:

```
#caja1 {padding: 5 10 5 10px;}
```

C) BORDES

Se pueden establecer tres tipos de propiedades a los bordes: el ancho, el color y el estilo.

Ancho.

Propiedades: **border-width, border-top-width, border-right-width, border-bottom-width, border-left-width**

Significado: **ancho de borde, ancho de borde superior, ancho de borde derecho, ancho de borde inferior y ancho de borde izquierdo**

Valores: **<número px,em> | thin | medium | thick | inherit**

Aplicación: **a todos los elementos**

1.- Color

Propiedades: **border-color, border-top-color, border-right-color, border-bottom-color, border-left-color**

Significado: color de borde, color de borde superior, color de borde derecho, color de borde inferior y color de borde izquierdo

Valores: **<color> | transparent | inherit**

Aplicación: **a todos los elementos**

2.-Estilo

Propiedades: **border-style, border-top-style, border-right-style, border-bottom-style, border-left-style**

Significado: **estilo de borde, estilo de borde superior, estilo de borde derecho, estilo de borde inferior y estilo de borde izquierdo**

Valores: **none | hidden | dotted | dashed | solid | double | groove | ridge | inset | outset | inherit**

Aplicación: **a todos los elementos**

Si el estilo es none o hidden no aparece borde

En la siguiente imagen te muestro los diferentes estilos para bordes

None

Hidden

Dotted

Dashed

Solid

Double

Groove

Ridge

Inset

Outset

D) COLORES Y FONDOS

1.-Color

Hay dos formas de especificar los colores en CSS

- Mediante palabras reservadas
- Mediante números hexadecimales en el modelo RGB

Palabras reservadas

CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:

aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

Ejemplo: `p{color:red;}`

Números en hexadecimal

Podemos obtener cualquier color mediante la mezcla de los tres colores primarios (Red, Green, Blue) en la cantidad adecuada

Son varios los formatos para dar cada color; hexadecimal de 3 dígitos (rgb), hexadecimal de 6 dígitos (rrggbb) decimal con tres valores y en tanto por ciento

Ejemplos

```
p{color:#ff0;}  
p{color:#ffff00;}  
p{color:rgb(255.255.0;}  
p{color:rgb(100%,100%0%;}
```

2.-Fondos

Solo se visualiza en la zona ocupada por el contenido y si relleno (content-padding) dado que el borde tiene su propio color y los márgenes son transparentes.

Para establecer un color o imagen de fondo para toda la página se debe aplicar el estilo al elemento `<body>`. En este caso todos los elementos de la página se visualizan con el mismo fondo salvo que algún elemento tenga el suyo propio

Hay cinco propiedades para establecer el fondo de cada elemento.

Propiedad: **background-color**

Significado: **color de fondo**

Valores: **<color> | transparent | inherit**

Aplicación: **a todos los elementos**

El valor inicial es transparent

Propiedad: **background-image**

Significado: **imagen de fondo**

Valores: **<url (“ruta a la imagen”)> | none | inherit**

Aplicación: **a todos los elementos**

El valor inicial es none

Propiedad: **background-repeat**

Significado: **sirve para repetir la imagen de fondo hasta rellenar el elemento**

Valores: **repeat | repeat-x | repeat-y | no-repeat | inherit**

Aplicación: **a todos los elementos**

El valor inicial es repeat

Propiedad: **background-position**

Significado: **indica el desplazamiento de la imagen desde la esquina superior izquierda, mediante dos valores, el desplazamiento horizontal y el desplazamiento vertical**

Valores: **(<num%> | <medida> | left | center | right | inherit) y**

(<num%> | <medida> | top | center | bottom | inherit)

Aplicación: **a todos los elementos**

El valor inicial es 0% 0%. Si uno de los dos se omite, el valor por defecto es 50%.

Propiedad: **background-attachment**

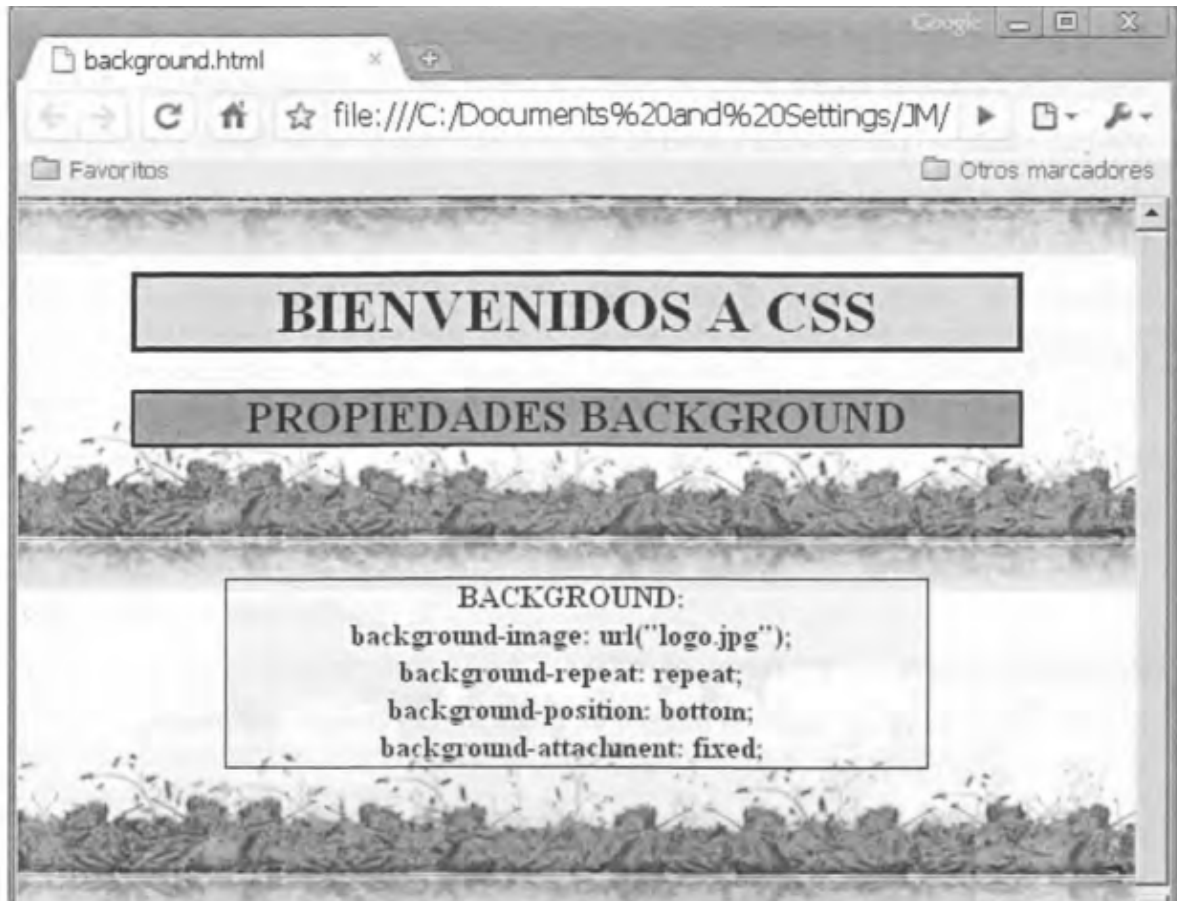
Significado: **controla el comportamiento de la imagen de fondo respecto a la scroll-bar**

Valores: **fixed | scroll | inherit**

Aplicación: **a todos los elementos**

El valor inicial es scroll

ACTIVIDAD 5. Crea una página como la de la figura con la imagen que prefieras, utilizando las propiedades que aparecen en la figura sobre el elemento <body>



- *Las propiedades background también pueden usarse dentro de formularios, incluso dentro de cada campo del formulario.*

5. POSICIONAMIENTO

CSS tiene 5 formas de establecer la posición en pantalla de una caja:

Estático, Relativo, Absoluto, Fijo y Flotante

Para establecer la posición disponemos de las propiedades:

position, top, bottom, right, left y float

Propiedad: **position**

Significado: **establece la posición de la caja en pantalla**

Valores: **static | relative | absolute | fixed | inherit**

Aplicación: **a todos los elementos**

El valor por defecto es static

- static: la caja ignora las propiedades top, bottom, right y left
- relative: la caja se desplazará según los valores top, bottom, right y left
- absolute: la caja se desplazará según los valores top, bottom, right y left partiendo de su elemento contenedor
- fixed: como el absoluto pero la caja permanece inmóvil independientemente del resto de elementos y del movimiento de la página por parte del usuario
- inherit: heredado

Propiedades: **top, bottom, right, left**

Significado: **establece el desplazamiento vertical y horizontal de la caja.**

Valores: **<número px> | <número %> | auto | inherit**

Aplicación: **a todos los elementos**

El valor por defecto es auto

Propiedad: **float**

Significado: **establece la posición flotante de la caja**

Valores: **left | right | none | inherit**

Aplicación: **a todos los elementos**

El valor por defecto es none

1.-Posicionamiento estático

Es el posicionamiento por defecto.

- Si los elementos a posicionar son bloques, como un párrafo, las cajas se colocan verticalmente una debajo de otra dentro del elemento contenedor
- Si los elementos a posicionar son de línea, como un hipervínculo, las cajas se colocarán una junto a otra horizontalmente dentro del bloque. Si las cajas con caben en una línea ocuparan la siguiente y si caben en una línea se alinean horizontalmente

OBSERVACIÓN

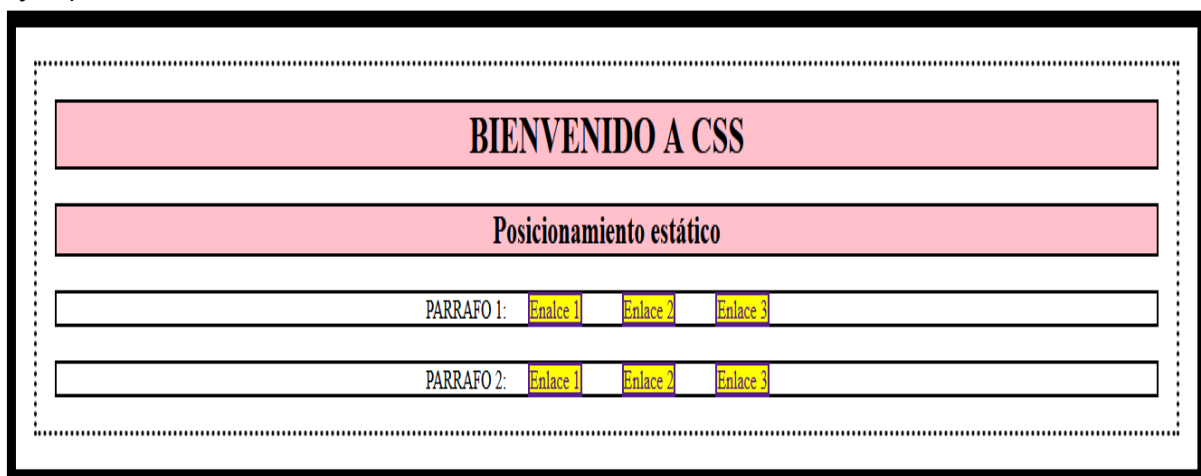
Los elementos en línea definidos por HTML son: `a`, `abbr`, `acronym`, `b`, `basefont`, `bdo`, `big`, `br`, `cite`, `code`, `dfn`, `em`, `font`, `i`, `img`, `input`, `kbd`, `label`, `q`, `s`, `samp`, `select`, `small`, `span`, `strike`, `strong`, `sub`, `sup`, `textarea`, `a`, `tt`, `u`, `var`.

Los elementos de bloque definidos por HTML son: `address`, `blockquote`, `center`, `div`, `dl`, `fieldset`, `form`, `h1`, `h2`, `h3`, `h4`, `h5`, `h6`, `hr`, `isindex`, `menu`, `noframes`, `noscript`, `ol`, `p`, `pre`, `table`, `ul`.

Los siguientes elementos también se considera que son de bloque: `dd`, `dt`, `frameset`, `li`, `tbody`, `td`, `tfoot`, `th`, `thead`, `tr`.

Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: `button`, `del`, `iframe`, `ins`, `map`, `object`, `script`.

Ejemplo



El código es:

```
<style>
  /*todos los elementos tendrán un borde solido y estarán centrados*/
  *{border:2px solid; text-align: center; margin:20px;}

  html{border:10px solid;} /*toda la página, salvo los elementos que reciban otro estilo*/
  body{border:dotted;}/*todos los elementos colocados en el body excepto los que reciban otro estilo*/
  h1,h2{background-color: #pink;}
  a{background-color: #yellow;}
</style>
</head>
<body>
  <h1>BIENVENIDO A CSS</h1>
  <h2>Posicionamiento estático</h2>
  <p> PARRAFO 1:
    <a href="#">Enlace 1</a>
    <a href="#">Enlace 2</a>
    <a href="#">Enlace 3</a>
  </p>
  <p>PARRAFO 2:
    <a href="#">Enlace 1</a>
    <a href="#">Enlace 2</a>
    <a href="#">Enlace 3</a>
  </p>
</body>
</html>
```

2.-Posicionamiento relativo

Consiste en desplazar la caja una cantidad respecto a la posición que le asignaría el posicionamiento estático.

- Cuando le asignamos un valor a la propiedad **top lo desplazamos hacia abajo** respecto a la posición que le corresponde en el posicionamiento estático
- Cuando le asignamos un valor a la propiedad **bottom lo desplazamos hacia arriba** con respecto a la posición que le corresponde en el posicionamiento estático.

Ejemplo

BIENVENIDOS A CSS

Posicionamiento estático

PARRAFO 1: [enlace1](#) [enlace2](#)

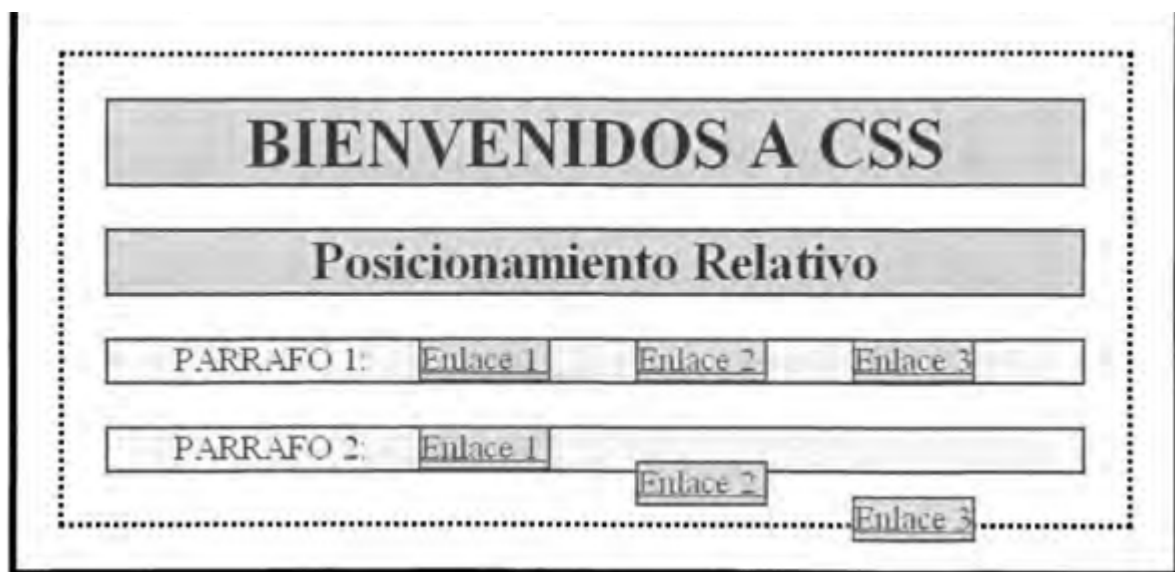
PARRAFO 2: [enlace1](#) [enlace2](#)

[enlace3](#)

```
a#test{border:3px solid; color:■ seagreen ; position: relative; top:4em;
```

Observa que he establecido el atributo “id” para referirme a él en la hoja de estilos

ACTIVIDAD 6. Usa el posicionamiento relativo para obtener la imagen que te proporcione



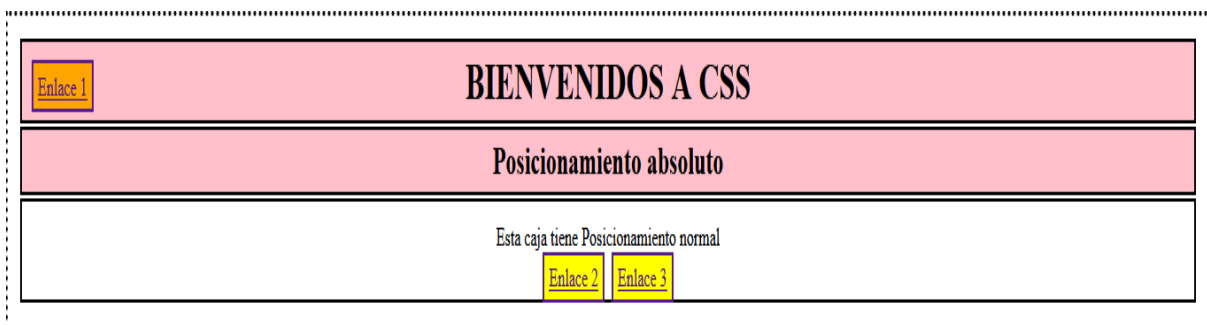
3.-Posicionamiento absoluto. {position:absolute}

Consiste en desplazar la caja una cantidad respecto de sus contenedores. La caja sale del flujo normal de la página y el resto de cajas ocupa su lugar

- El origen de coordenadas está en la esquina superior izquierda

ACTIVIDAD 7: Escribe el siguiente código y a continuación modificarlo estableciendo posicionamiento absoluto a los distintos enlaces

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     *{text-align: center; margin:2px; padding: 5px;}
10    html{border: dotted;}
11    h1,h2{background-color: pink; border:2px solid;}
12    div{border:2px solid;}
13    a{background-color: yellow; border :2px solid;}
14    a.absoluto{background-color: orange; position:absolute;top: 30px;left: 30px;}
15  </style>
16 </head>
17 <body>
18   <h1>BIENVENIDOS A CSS</h1>
19   <h2>Posicionamiento absoluto</h2>
20   <div>
21     <p>Esta caja tiene Posicionamiento normal</p>
22     <a href="#" class="absoluto">Enlace 1</a>
23     <a href="#">Enlace 2</a>
24     <a href="#">Enlace 3</a>
25   </div>
26 </body>
27 </html>
```



4.Posicionamiento fijo {position:fixed}

Cuando el usuario desplaza la página las cajas se mantienen fijas en el mismo lugar. Es útil para encabezados y pies de página

También es útil para menús de navegación lateral que deseamos que estén siempre visibles

ACTIVIDAD 8. Escribe el código html que falta para fijar un elemento html haciendo uso de los estilos que te proporcione

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    *{text-align: center;}
    p{border: 2px solid ■saddlebrown;}
    h1{border: 2px solid ■salmon; position: fixed;width: 80%;}
    h2{border:2px solid ■yellow;}
    div{border:3px double}
  </style>
</head>
```

5.-Posicionamiento flotante. {float:left | right | none}

Desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en que se encuentran. El resto de las cajas ocupan el lugar dejado por la caja flotante.

No se produce solapamiento de cajas.

ACTIVIDAD 9: Prueba y modifica este código

```

<style>
  h1{text-align: center;}
  a#uno{border:2px solid;border-color: red;}
  a#dos{border:2px solid;border-color: royalblue;}
  a#tres{border:2px solid;border-color: yellow;}

  a{float: right;}
  a.fijo{float:left;}
</style>
</head>
<body>
  <header>
    <h1>Posicionamiento float:right y estático</h1>
  </header>
  <section>
    <a id="uno" href="#">UNO</a>
    <a id="dos" href="#">DOS</a>
    <a id="tres" href="#">TRES</a>
  </section>
  <section>
    <h2>Posicionamiento estático</h2>
    <a class="fijo" href="#">UNO</a>
    <a class="fijo" href="#">DOS</a>
    <a class="fijo" href="#">TRES</a>
  </section>
</body>
</html>

```

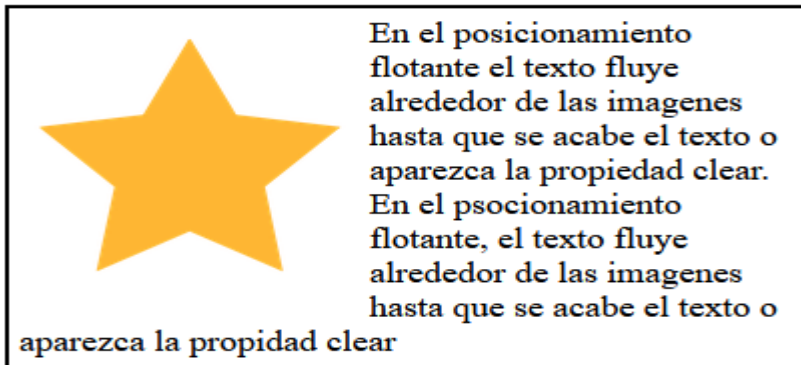
ACTIVIDAD 10. Prueba y modifica el siguiente código

```

<style>
  #principal {border:2px solid;width: 400px; font-size:large;}
  img{float:left;}
  p{margin: 5px;}
</style>
</head>
<body>
  <h1>Posicionamiento flotante</h1>
  <div id="principal">
    
    <p>En el posicionamiento flotante el texto fluye alrededor de
      las imagenes hasta que se acabe el texto o aparezca la
      propiedad clear. En el psocionamiento flotante, el texto fluye
      alrededor de las imagenes hasta que se acabe el texto
      o aparezca la propiedad clear
    </p>
  </div>
</body>
</html>

```

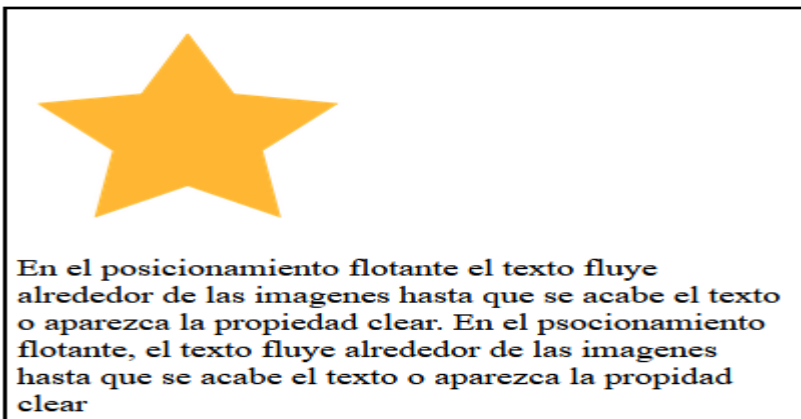
Posicionamiento flotante



- La propiedad clear permite evitar el comportamiento del posicionamiento flotante. En el valor se establece el lado que no debe ser adyacente a ninguna caja flotante.

ACTIVIDAD 11. *Obtén la siguiente página*

Posicionamiento flotante



6.TEXTO

Las principales propiedades relacionadas con el texto son:

Propiedad: valor	Significado
color: <color> inherit	Color del texto
font-family: <fuente> inherit	Tipo de fuente de letra
font-size: <absoluto> <relativo> <%> inherit	Tamaño de letra
font-weight: normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	Grosor de letra
font-style: normal italic oblique inherit	Estilo de letra
font-variant: normal small-caps inherit	Mayúsculas en pequeño
text-align: left right center justify inherit	Alineación horizontal
line-height: normal <numero> <medida> <porcentaje> inherit	Interlineado
text-decoration: none (underline overline line-through blink) inherit	Decoración
text-transform: capitalize uppercase lowercase none inherit	Transformación
text-shadow: none h-shadow v-shadow blur color	Sombreado del texto
vertical-align: baseline sub super top text-top middle bottom text-bottom <porcentaje> <medida> inherit	Alineación vertical
text-indent: <medida> <porcentaje> inherit	Tabula las primeras líneas
letter-spacing: normal <medida> inherit	Espaciado entre letras
white-space: normal nowrap pre pre-line pre-wrap inherit	Tratamiento de los espacios en blanco
word-spacing: normal <medida> inherit	Espaciado entre palabras

ACTIVIDAD 12. Dota de estilos a los enlaces de la figura para que no estén subrayados hasta que los selecciones con el ratón (propiedad text-decoration)

CICLOS

[CS01, ASIR](#)
[CS02, DAM](#)
[CS03, DAW](#)
[CM04, SMR](#)

7.LISTAS Y TABLAS

7.1.-Listas.

- Las principales propiedades son:

Propiedad: valor	Significado
<code>list-style-type: disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none inherit</code>	Viñeta usada para los elementos de la lista
<code>list-style-position: inside outside inherit</code>	Posición de la viñeta
<code>list-style-image: <url> none inherit</code>	Sustituye la viñeta por una imagen
<code>List-style: (propiedad shorthand)</code>	Combina las anteriores

Prueba este ejemplo de lista con sin orden

Lista de la compra saludable

- Caja de leche desnatada 6 L
- Menestra de verduras 1 kg
- Dorada en filetes 450 g

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    ul{list-style-type: circle;}
  </style>
</head>
<body>
  <header>
    <h1>Lista de la compra saludable</h1>
  </header>
  <section>
    <ul>
      <li>Caja de leche desnatada 6 L</li>
      <li>Menestra de verduras 1 kg</li>
      <li>Dorada en filetes 450 g</li>
    </ul>
  </section>
</body>
</html>
```

7.1.1.-Menús

Se va a construir un menú horizontal utilizando una lista, para ello utilizaremos la propiedad `display:inline`, que visualiza los elementos en línea y sin viñetas.



- En un principio los elementos se visualizan como elementos de bloque (cada uno ocupa una línea) Para que el menú sea horizontal aplicamos `display:inline`
- Usaremos listas encadenadas.
- Quitaremos los estilos que por defecto ponen los navegadores con
 - `ul,li { list-style-type:none}`
 - Está misma propiedad la puedo usar para dar estilos a las listas
- Declararemos una clase para referirnos a través de ella al menú
 - `<ul class="miMenu">`
 - `<!-- Resto del código -->`
- Asignaré bordes a los elementos del menú
- Emplearé la propiedad `overflow` para decidir cómo actuará el navegador cuando el contenido de un elemento sale del navegador
- Estableceré la anchura de los elementos del menú (selector `li` descendiente de mi menú)
 - `ul.Mimenu li { width:30%; /*resto del código*/}`
- Convertiré los elementos de cada opción del menú en un elemento de bloque con la propiedad `display`
 - `ul.miMenu li a{ display:block; /*resto del código*/}`

Ejemplo. Crearemos el siguiente menú


```
<body>
  <!-- usamos el caracter $#9660 para que aparezca una flecha. Si no te agrada eliminalo-->
  <ul class="miMenu">
    <li>
      <a href="#">Secundaria <span class="flecha">#9660</span></a>
      <ul>
        <li><a href="#">Matematicas</a></li>
        <li><a href="#">Fisica</a></li>
        <li><a href="#">Química</a></li>
      </ul>
    </li>
    <li>
      <a href="#">Bachillerato <span class="flecha">#9660</span></a>
      <ul>
        <li><a href="#">Idiomas</a></li>
        <li><a href="#">Historia</a></li>
      </ul>
    </li>
    <li>
      <a href="#">Ciclos <span class="flecha">#9660</span></a>
      <ul>
        <li><a href="#">ASIR</a></li>
        <li><a href="#">DAW</a></li>
        <li><a href="#">DAM</a></li>
      </ul>
    </li>
  </ul>
</body>
```

```

/*elimina las viñetas por defecto*/
ul,li{
    list-style-type: none;
}

/*METODO 1 (Quizas no sea al mejor) */

ul.miMenu{
    list-style-type: none;
    background-color: #rgb(165, 142, 42);
    border: 1px solid #ffffff;
    /*overflow define como actua el navegador cuando el contenido de un elemento se sale del contenedor
    1.-visible : el contenido se sale del contenedor
    2.-hidden: el navegador oculta el contenido que se sale de la caja*/
    overflow: hidden;
}

/*establecemos la anchura de las opciones que componen el menú*/
/* i)Me refier a ellos usando el selector de clase
ii)Establezco la anchura del item y lo hago flotar a la izquierda
('poca anchura' los apelmaza (unos junto al otro). Prueba distintos valores hasta que estes safisfecho)
*/
ul.miMenu li{
    width: 30%;
    float: left;
}

```

```

/*establecemos la propiedades de los items del menú , que en este caso son enlaces*/
ul.miMenu li a{
    border: 1px dotted gray;
    /*convierte los items de cada opción de menú en un elemento de bloque*/
    display: block;
    /*establecemos otros estilos como tipo de letra, tamaño de la fuente, .... (no es obligatorio)*/
    font-family: Impact, Haettenschweiler, 'Arial Narrow Bold', sans-serif;
    font-size: 1rem;
    /*OBSERVACIONES:
    |i)prueba con distintos tamaños y tipos de letras, Veras que cambia mucho el resultado
    */
}

/*que los submenús no se vean al cargar la página*/
ul.miMenu li ul{
    display: none;
}

```

```

/*mostrar los submenús al pasar el cursos como elementos de bloque
OBSERVACIONES:
    i) Buscamos que cualquier submenú se muestre como elemento de bloque
    ii)Para que el estilo se aplique a todos los submenús y dado que los submenús se implmentan
        con etiquetas ul que cuelgan de etiquetas li (listas anidadas) usamos el selector por contexto del tipo 'hermano'
    iii)como deseamos que el estilo se aplique cuando se ejecute un evento determinado (cursor
        por encima del menú) usamos una pseudoclase
    iv)Para que se muestren como elementos de bloque establezo la propiedad display:block*/

ul.miMenu li:hover > ul{
    display: block;
}

/*colocar el submenú a la derecha de su ancestro
OBSERVACIONES:
    i)Usamos un posicionamiento relativo porque de esta forma se colocará según el flujo normal
    pero pudiendose desplazar. De no hacerlo así se mostrará a la derecha pero no uno debajo del otro
    ii)Empleo un selector por contexto de tipo padre.
*/

ul.miMenu li ul li{
    position: relative;
}

/*ubicar todas las opciones del primer nivel de la jerarquia a la izquierda*/
ul.miMenu > li{
    float: left;
}

```

Activar Windows
Ve a Configuración para activar

8.TABLAS

- Propiedades CSS de las tablas

Propiedad: valor	Significado
<code>border-collapse: collapse separate inherit</code>	Determina la fusión de bordes
<code>border-spacing: <medida> <medida> inherit</code>	Separación de bordes horizontal y vertical
<code>empty-cells: show hide inherit</code>	Celdas vacías
<code>caption-side: top bottom inherit</code>	Posición del título

Ejercicio: Crea una tabla con 3 colores, 1 color para la fina head y el resto de filas intercambiando el color blanco y verde o gris muy claro . Además

- Usa selectores de clase , de id y descendientes
- Asigna estilos de tamaño de letra, fuente y bordes
- Usa la sintaxis mejora del html para tablas

Intenta conseguir un resultado parecido a este.

Tabla con estilos

Empresa	Contacto	Pais
Mercedes Benz	Maria Naiditsch	Alemania
Volvo	Cristina Carlsen	Suecia
Seat	Francisco Valle	España
Toyota	Hikaru Kamikaze	Japon
Ferrari	Danielle Rovelli	Italia
Cadillac	Simo Says	USA
Renault	Marie Curie	Francia