CSS Y CSS3

CSS son las siglas de "Cascade StyleSheet". Se trata de una especificación sobre los estilos físicos aplicables a un documento HTML, y trata de dar la separación definitiva de la lógica (estructura) y el aspecto (presentación) del documento.

El estilo lógico se refiere a la lógica del documento: cabeceras, párrafos, tablas ... no se preocupa de la apariencia final, sino de la estructura del documento. Por el contrario, el estilo físico no se preocupa de la estructura del documento, sino de la apariencia final: párrafos con un cierto tipo de letra, tablas con un determinado color de fondo, ...

HISTORIA:

CSS1 (1996): Introducción básica de estilos, centrado en textos y colores.

CSS2 (1998): Posicionamiento avanzado, soporte para impresiones y tablas mejoradas.

CSS2.1 (2011): Corrección de errores y mejor compatibilidad.

CSS3 (2001 - actual): Sistema modular, con mejoras como Flexbox, Grid, animaciones, transiciones, y responsive design.

CSS4: No existe formalmente como versión, pero CSS sigue evolucionando a través de módulos independientes.

CSS está en constante evolución, y los navegadores modernos están implementando continuamente nuevas características para ofrecer mejores herramientas de diseño a los desarrolladores.

Si ya has visto CSS prueba tus conocimientos con este test:

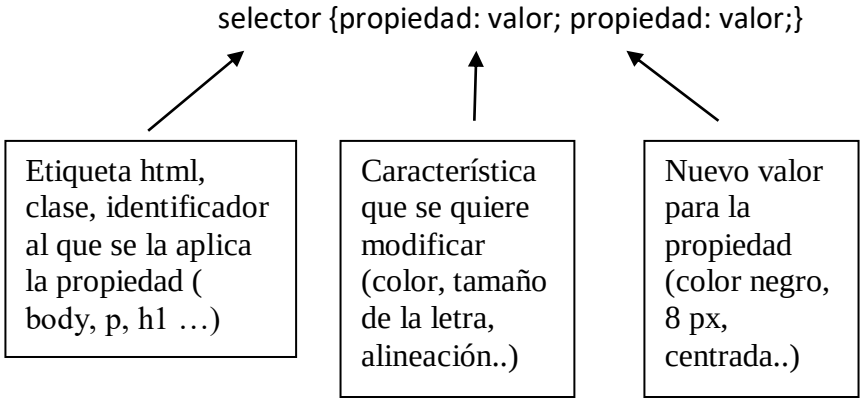
W3Schools CSS Quiz

1.REGLA O ESTILO CSS

En CSS para cambiar el color de fondo de una página lo haremos con esta regla:

body { background-color:#FF0000; }

Las reglas en CSS por lo general tienen esta forma:



2. DÓNDE SE SITÚAN LAS REGLAS CSS

Existen tres posibilidades:

- **Aplicación directa en etiquetas**: especificarlo directamente en la etiqueta en la que queremos usarlo:

```
<etiqueta style="propiedad1:valor;...;propiedadN:valor"> ... </etiqueta>
```

Por ejemplo, si tenemos un texto en un párrafo y queremos que salga con un tamaño de letra 30px y en color rojo, haremos:

```
<p style="color:red;font-size:30px;">Esto es un párrafo con estilos</p>
```

Esta forma es la menos debemos usar para insertar estilos en un documento para separar la estructura del documento de su forma.

- **Aplicación a todo el documento**: Definiremos estilos globales a todo el documento en la cabecera del documento por medio de la etiqueta <style>…</style>

Por ejemplo:

```
<head>
<style type=”text/css”>
body {
   background-color: linen;
}
h1 {
   color: maroon;
   margin-left: 40px;
}
</style>
</head>
```

- **Separar HTML de CSS:** definir estilos globales en un fichero distinto al documento HTML y después hacer una referencia a él en la cabecera del documento.

Por ejemplo:

```
<head>
<link rel="stylesheet" type="text/css" href="ejemplo.css">
</head>
```

El archivo ejemplo.css podría tener esta forma:

```
/* Definición de estilos en un archivo aparte */

body {
        background-color: linen;
    }
    h1 {
        color: maroon;
        margin-left: 40px;
    }
```

Si quisiéramos que otros documentos tuvieran el mismo estilo, sólo tendríamos que usar la etiqueta link para aplicarlos. Esa es la ventaja de las hojas de estilo externas.

En las hojas css se pueden introducir comentarios usando /* */.

3. REGLAS PARA LA APLICACIÓN DE LOS ESTILOS

3.1. Herencia

En el conjunto de las etiquetas HTML podemos establecer una jerarquía de etiquetas que contienen a otras, para darnos una relación de herencia. En primer lugar, tendríamos la etiqueta <body> ... </body>, que hace referencia a todo el documento, y podemos considerarla como la etiqueta "padre" de todas las demás etiquetas ya que todas ellas se encuentran contenidas en el cuerpo del documento.

Después, tenemos las etiquetas de párrafo (... , ... , cabeceras, ...) y etiquetas de elementos insertados en línea (...,..., ...). Las etiquetas de párrafo serán contenedoras de las etiquetas de elementos insertados en línea, estableciéndose así una nueva relación "padre-hijo".

La mayoría de los estilos que se definen se heredan, es decir, si definimos un cierto estilo para una etiqueta, este estilo será heredado por las etiquetas "hijas", con lo que no tendremos que volver a definirlo para ellas. Por ejemplo, si definimos un tipo de letra y un color para la fuente para la etiqueta <body> ... </body>,este estilo será heredado por todas las etiquetas del documento y no tendremos que definirlo para las otras etiquetas.

Sin embargo, si tenemos definido un estilo para una etiqueta "padre", podremos definir un estilo distinto para una etiqueta "hija", es decir, un estilo heredable se hereda a no ser que especifiquemos lo contrario. Se heredarán aquellas características que no pongamos, y se aplicarán aquellas que definamos para la etiqueta que no tenga la etiqueta "padre".

Ejemplo:

```
<html>
<head>
<title> Ejemplo con bloque de estilo </title>
<style>
   body {color:blue;}
   h2 {color:red;}
   span {color:green;}
</style>
</head>
<body>
Este texto se ve azul
<h2>Este texto se ve rojo y <span>este verde</span></h2>
</body>
</html>
```

3.2. Orden de las reglas.

Las distintas reglas para un mismo selector se definan dónde se definan se aplican a dicho selector. Pero a reglas iguales tiene preferencia la última regla definida.

Por ejemplo, si establecemos el color de un párrafo azul en una regla en la cabecera y a continuación enlazamos con un archivo css en el que el color del párrafo es rojo, el párrafo se mostrará de color rojo.

De igual manera una regla en una etiqueta tendrá prioridad sobre cualquier otra regla de esa propiedad que esté en el head o en un documento aparte.

Por otro lado si en un documento definimos una característica mediante un atributo html y mediante una regla css, tiene preferencia esta última. Por ejemplo, si usamos width como atributo y width como un estilo, se aplicará este último valor.

3.3. Especificidad

Cuánto más específico es el selector más prioridad tiene, por ejemplo, (un selector de ID tiene más prioridad que uno de clase).

3.4. ¡important

En **CSS**, !important es una palabra clave que se utiliza para **forzar la prioridad de una regla de estilo** por encima de otras reglas que puedan entrar en conflicto. Al agregar !important a una declaración de estilo, esa declaración tendrá más peso que cualquier otra regla CSS que se aplique al mismo elemento, incluso si otra regla es más específica o está definida más tarde en la hoja de estilos.

Ejemplo:

```
p { color: blue !important; /* Se aplica este estilo */ }

 p { color: red; }
```

No hay que abusar de este modificador. Puede ser útil cuando tengamos que modificar un estilo en un CMS y no podamos editarlo.

4. SELECTORES

Un selector en una regla CSS nos dice a quien hay que aplicarle dicha regla.
Una misma regla se puede aplicar sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

Hay varios tipos de selectores.

4.1. Selector universal

Se utiliza para seleccionar todos los elementos de una página.
El siguiente ejemplo elimina el margen y el relleno de todos los elementos:

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

4.2. Selector de etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector.
El siguiente ejemplo, selecciona todos los párrafos de la página y los pone en azul.

```
p {color:blue;}
```

4.3. Selector de grupo

Si se quieren aplicar los mismos estilos a varios selectores diferentes, los selectores se separan por comas:

```
h1,h2,h3 {
            color:white;
            font-family: Arial;
          }
```

Si a continuación escribimos las siguiente regla:

```
h1 { background-color:blue;}
```

Estamos diciendo que los títulos h1 se muestren en blanco , con fuente arial y de color de fondo blue.

4.4. Selector descendente

Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

En el siguiente ejemplo se muestran en rojo los elementos de la página que se encuentren dentro de un elemento <p>.

```
p span {color:red}
```

Ejemplo:

```
<p>
...
<span>Este texto va en rojo</span>
...
<a href="">...<span>Este texto va en rojo tambien</span></a>
...
</p>
<h1><span>Este texto no va en rojo</span></h1>
```

4.5. Descendiente directo

Selecciona solo los elementos que son **hijos directos** de otro elemento. Utiliza el símbolo >.

Ejemplo:

```
div > p {
  color: orange;
}
```

Selecciona solo los elementos <p> que son hijos directos de un , pero no los que están más profundamente anidados.

4.6. Selector de clase

Lo usamos cuando un estilo concreto lo queremos aplicar a varias etiquetas.

Para ello, en primer lugar definimos la clase (en el bloque de estilos o en la hoja externa) como un estilo más, de esta forma:

```
.Nombre_de_la_Clase {propiedad1:valor;...;propiedadN:valor}
```

Ahora, para aplicar el estilo de una clase a una etiqueta concreta, utilizaremos el parámetro class como sigue:

```
<etiqueta class="Nombre_de_la_Clase"> ... </etiqueta>
```

donde Nombre_de_la_Clase es el nombre que le hemos dado a la clase, sin el punto.

Ejemplo:

```
<html>
<head>
<title> Ejemplo de uso de clases </title>
<style>
```

```
    p {color:red; font-family:sans-serif; }
    .Azul {color:blue}
</style>
</head>
<body>
En este ejemplo vamos a ver cómo se aplican las clases.
<h4>Este es un título normal</h4>
<h4 class="Azul">Este es un título con clase y aparecerá en azul</h4>
<p>Este es un párrafo normal</p>
<p class="Azul">Este es una parrafo con clase y aparecerá en azul</p>
</body>
</html>
```

4.7. Selector de id

Cualquier etiqueta HTML puede tener como parámetro la etiqueta id seguida de un nombre, por ejemplo:

```
<etiqueta id="Nombre"> ... </etiqueta>
```

Este "Nombre" **debe ser único en el documento HTML** (es decir, no debe haber dos etiquetas con el mismo id), puesto que nos servirá para tratarla (si lo necesitamos) desde JavaScript. El id no puede empezar por número.

Para definir un estilo mediante un id, escribimos # seguido del nombre del id (en un bloque de estilo o en la hoja externa):

```
#Nombre_del_id {propiedad1:valor;...;propiedadN:valor}
```

```
<html>
<head>
<title> Ejemplo de uso de id</title>
<style>
    p {color:red;  }
    #miparrafo {color:blue}
</style>
</head>
<body>

<p>Este es un párrafo normal</p>
<p id="miparrafo">Este es una parrafo con id y aparecerá en azul</p>
</body>
</html>
```

4.8. Pseudoclases

a:link Nos dice el estilo de un enlace que no ha sido visitado.
a:visited Nos dice el estilo de un enlace que ha sido visitado.
a:active Nos dice el estilo de un enlace que está siendo pulsado.
a:hover Nos dice el estilo de un enlace sobre el que está pasando el ratón.

Por ejemplo, si deseásemos que apareciesen todos los enlaces sin subrayar, definiríamos los siguientes estilos:

```
a:link,a:visited,a:active {text-decoration:none}
```

Las pseudoclases :hover, :active y :focus se pueden aplicar a cualquier elemento.

:hover	Se activa cuando el usuario pasa el ratón por encima de un elemento.
:active	Se activa cuando el usuario pulsa el botón del ratón sobre un elemento, sólo dura hasta que se suelta el botón.
:focus	Se activa cuando el elemento está seleccionado, sobre todo se usa en los elementos input.

4.9.Pseudo-elementos

Se usan para especificar parte de un elemento.

Existe un pseudoelemento ::firstletter que se utiliza para cambiar la primera letra de un texto.
El siguiente ejemplo muestra la primera letra de cada párrafo 40px:

```
p::first-letter{
   font-size: 40px;}
```

Otro pseudoelemento es ::first-line que selecciona la primera línea de texto de un elemento.

Otros pseudoelementos son ::before y ::after

Ejemplo

```
/* Inserta contenido antes de un elemento */
p::before { content: "Nota: "; font-weight: bold; }
```

4.10. Selector de atributo

Selecciona elementos que tienen un atributo específico o un valor de atributo determinado.

Ejemplos:

```
/* Selecciona todos los elementos con el atributo title */
[title] {
  color: red;
}
/* Selecciona los enlaces que empiezan por http, e.d.  que apuntan a URLs externas */
a[href^="http"] {
  text-decoration: none;
}
/* Selecciona los inputs de tipo texto */
input[type="text"] {
  border: 1px solid black;
}
```

4.11. Combinación de selectores

Ejemplos:

```
/* Todos los elementos de tipo "p" con atributo id="aviso" */
p#aviso { ... }
/* Todos los elementos con atributo id="aviso" que estén dentro
de cualquier elemento de tipo "p" */
p #aviso { ... }
/* Todos los elementos "p" de la página y todos los elementos con
atributo id="aviso" de la página */
p, #aviso { ... }
/* Todos los elementos de tipo "div" con clase "aviso" */
div.aviso { ... }
/* Todos los elementos que sean de clase "especial" y que además estén dentro de
una etiqueta que sea de clase "aviso" */
.aviso .especial { ... }
```

Aquí puedes encontrar todos los selectores CSS:

CSS Selectors Reference (w3schools.com)

5. VALORES

5.1. Unidades de medida

Las medidas en CSS se emplean para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto.
Todas las medidas se indican como un valor numérico entero o decimal seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida). Si no ponemos unidad de medida se ignora dicha propiedad.
CSS divide todas las unidades de medida en :

5.1.1. Absolutas: Su valor es el indicado.

- mm (milímetros),
- cm (centímetros).
- pt (puntos), 0.35 mm
- pc(picas), 4,23 mm
- in (pulgadas), 2.54 cms
- px (píxel). (1/96 pulgadas) Relativa a la pantalla del usuario, un px en un dispositivo de baja resolución equivale a un pixel del dispositivo, si es de alta resolución equivale a varios pixels.

Ejemplos:

```
body { margin: 0.5in; }
h1 { line-height: 2cm; }
p { word-spacing: 4mm; }
a { font-size: 12pt }
span { font-size: 1pc }
```

5.1.2.Relativas: Su valor depende de otra medida.

- em relativa al font-size del elemento (1em significa igual tamaño)
- ex (la altura de la x minúscula), relativa a la letra empleada (1ex es aproximadamente 0.5em) (Se utiliza poco)
- vh 1% del alto de la ventana del navegador.
- vw 1% del ancho de la ventana del navegador.

Ejemplos:

```
body { font-size: 10px; }
h1 { font-size: 2.5em; }
```

Los títulos h1 serán de 25 px.

5.1.3. Porcentajes

Son un valor numérico más el símbolo del porcentaje %. Se suele usar para definir el alto y ancho y representan el porcentaje de la medida de su elemento padre.

Ejemplo:

```
#uno { width:50px;}
#dos {width:80%;}
.....
<div id="uno">
        <div id="dos">
....
```

El div dos medirá 40px

Podemos usar la función calc() para calcular tamaños, fundamentalmente se usa para hacer los cálculos entre porcentajes y pixels.

Ejemplo:

```
#div1 {
  position: absolute;
  left: 50px;
  width: calc(100% - 100px);
  border: 1px solid black;
  background-color: yellow;
  padding: 5px;
  text-align: center;
}
```

5.2. Colores

Se pueden indicar de varias maneras:

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

rgb decimal: Cantidad de rojo, verde y azul con un un número de 0 a 255.

```
p { color: rgb(200, 95, 100);  }
```

rgb porcentual: Igual que el anterior pero con valores entre 0% y 100%

```
p { color: rgb(90%, 55%, 85%);  }
```

rgb hexadecimal: Igual que en html, cantidad de rojo, verde y azul en hexadecimal.

```
p { color:#239A25;  }
```

Podemos añadir un canal alfa a los colores rgb utilizados, de esta forma los colores se representan usando **rgba** con 4 parámetros, este cuarto parámetro que se añade indica el grado de opacidad de 0 a 1 (0 es transparente, 1 es opacidad total):

Ejemplo: El color de fondo tiene una opacidad intermedia.

```
body { background-color:rgba ( 100, 200,40, 0.5);}
```

También se puede usar el atributo **opacity** con valores del 0 a 1, la diferencia con lo anterior es que afecta a todo lo incluido dentro del elemento:

Ejemplo: Toda la página tiene una opacidad del 0.5

```
body{ background-color:rgb( 100, 200, 40); opacity:0.5; }
```

gradiente

Te permite definir transiciones entre uno o más colores. Puede ser sólo de transparencia:

```
<head>
    <style>
.gradiente-transparente {
       height: 300px;
       background-image: linear-gradient(to right, rgba(255, 126, 95, 1), rgba(255, 126, 95, 0)); /* De opaco a transparente */ }
    </style>
</head>
     <body> <div class="gradiente-transparente"></div> </body>
```

Puede ser lineal, diagonal o conic

CSS Gradients (w3schools.com)

6. PROPIEDADES

6.1. Fondos

PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
background-color	Color de fondo	Color, transparent, inherit	transparent	Todos los elementos	1
Background-image	Imagen de fondo	url(nombre del archivo imagen), none, inherit	none	Todos los elementos	1
Backgroun-repeat	Repetición de la imagen de fondo	Repeat, repeat-x, repeat-y, no-repeat, inherit	repeat	Todos los elementos	1
Background-position	Posición de la imagen de fondo, respecto de la esquina superior izda.	Desplazamiento. horizontal(left, center, right,x%,xpos), desplazamiento vertical(top,center, Bottom, y%,ypos)	0% 0%	Todos los elementos	1
Background-attachment	Indica si la imagen se desplaza o no cuando se hace scroll en el navegador	Scroll, fixed, inherit	scroll	Todos los elementos	1
Background-size	Tamaño de la imagen de fondo	Ancho y alto, Cover (rellena todo el fondo, partes de la imagen pueden no verse) Contain (hace la imagen lo más grande posible que coja en el fondo)	auto	Todos los elementos	3

En firefox , Opera y Chrome, para que funcione background- position, background-attachment tiene que ser fixed.

En la forma abreviada esta propiedad tiene los siguientes valores y además en este orden:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

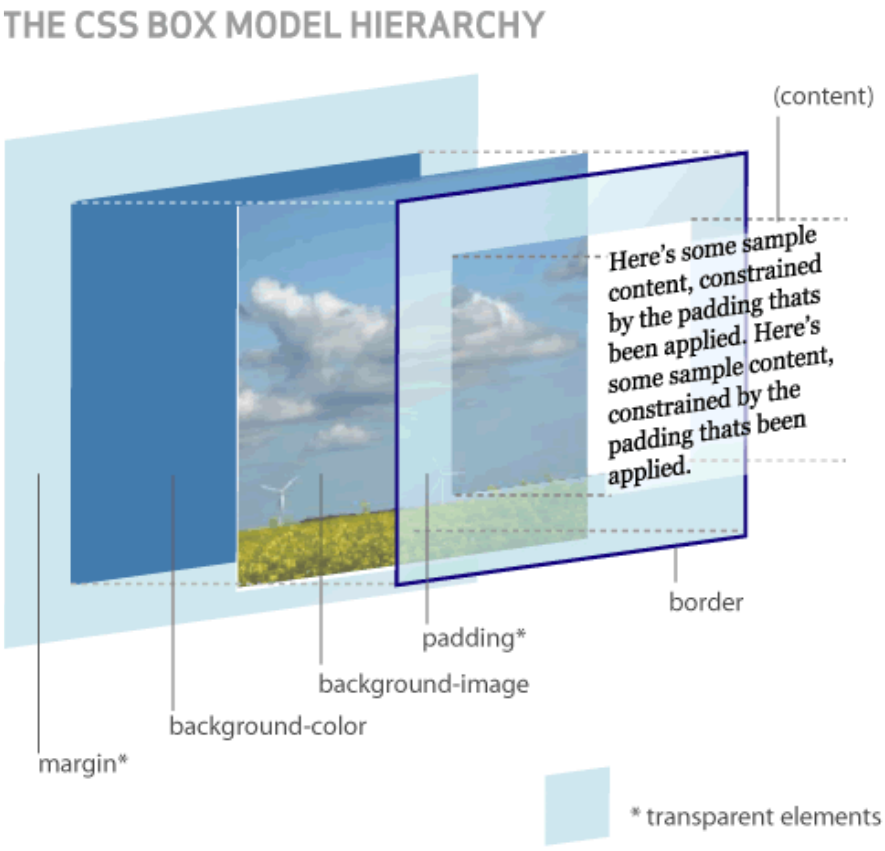
Si falta el valor de alguna propiedad no pasa nada si las demás conservan el orden.
Por ejemplo sería correcto:

```
background: #ffffff url("img_tree.png") no-repeat right top;
```

6.2. Cajas

Todos los elementos html se representan mediante cajas rectangulares.

El elemento html por excelencia que representa una caja y que vamos a utilizar para organizar el contenido de una página es la etiqueta div.
Un div es un elemento de bloque, esto quiere decir que es un elemento que por defecto ocupa todo el ancho de página completo disponible, y tiene un salto de línea antes y después de ella.
Las propiedades de una caja son:



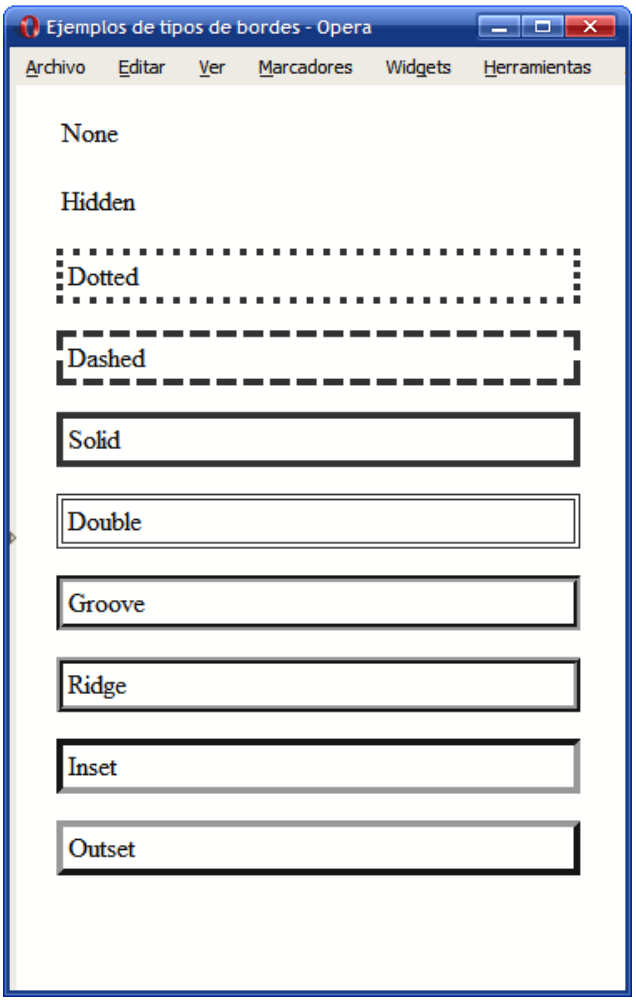
PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
width	Ancho del contenido	Medida, porcentaje, auto o inherit	auto	Todos los elementos de bloque e imágenes	1
Max-width	Ancho máximo del contenido	Medida: Si la ventana del navegadores más pequeña se reduce el width, si es más ancha como mucho sólo se ve el width.			
height	Altura del contenido	Medida, porcentaje, auto o inherit	auto	Todos los elementos de bloque e imágenes	1
margin-top margin-right margin-bottom margin-left	Margen superior Margen derecho Margen inferior Margen izquierdo	Medida, porcentaje, auto o inherit	0	Margin-top y margin-bottom a elementos de bloque e imagenes. El resto a todos.	1
Padding-top Padding-right Padding-bottom Padding-left	Relleno superior Relleno derecho Relleno inferior Relleno izquierdo	Medida, porcentaje, inherit	0	Todos los elementos	1
Border-top-width Border-right-width Border-bottom-width Border-left-width	Ancho borde superior Ancho borde derecho Ancho borde inferior Ancho borde izquierdo	Medida, thin, medium, thick o inherit	medium	Todos los elementos	1

Border-top-color Border-right-color Border-bottom-color Border-left-color	Color borde superior Color borde derecho Color borde inferior Color del borde izquierdo	Color, Transparent, inherit		Todos los elementos	1
Border-top-style Border-right-style Border-bottom-style Border-left-style	Estilo borde superior Estilo borde derecho Estilo borde inferior Estilo borde izquierdo	None, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit	none	Todos los elementos	1
Border-top Border-right Border-bottom Border-left	Estilo completo de cada uno de los bordes	Tres valores separados: medida, color y estilo.		Todos los elementos.	1
outline	Contorno alrededor de una caja fuera de los borde.	Tres valores: color, tipo y grosor.		Todos los elementos	2
border-top-left-radius border-top-right-radius border-bottom-left-radius border-bottom-right-radius	Redondea los bordes de una caja	Medida , porcentaje .	0	Todos los elementos	3
Box-shadow	Sombra.	5 valores: Distancia horizontal, distancia vertical, borrosa, tamaño. color		Todos los elementos	3

Box-sizing	Medidas del width y el height	Content-box: El width y el height es solo para el contenido. Border-box:El width y el height incluyen el border y el padding.	Content -box	Todos los elementos	3

Cuando se juntan dos o más márgenes verticales se fusionan de forma automática y la altura del nuevo margen es la del margen más alto. Lo mismo si un elemento está dentro de otro.

Estilos de bordes:



La forma abreviada del borde es:

border: border-width border-style border-color;

6.3. Posicionamiento

Un elemento de bloque, como hemos dicho anteriormente, es un elemento que ocupa todo el ancho de página completo disponible, y tiene un salto de línea antes y después de ella.

Ejemplos de elementos de bloque: div,h1,..,h7, p,ul, ol, li , hr, fieldset, table...

Para centrar un elemento de bloque horizontalmente se pone como valor auto en las propiedades margin-left y margin-right. Para que funcione tiene que tener definido un ancho (width) ya que si no ocuparía toda la línea y en ese caso no se podría centrar.

Un elemento en línea sólo ocupa el ancho que sea necesario, y no salta de línea.

Ejemplos de elementos en línea: span, a, img, input, select, textarea, sub, sup ...

Los elementos de bloque no pueden insertarse dentro de
elementos en línea y tan sólo pueden aparecer dentro de otros elementos de bloque. En cambio, un elemento en línea puede aparecer tanto dentro de un elemento de bloque como dentro de otro elemento en línea.

Nociones importantes sobre posicionamiento

- Si quiero que varios elementos compartan una línea:
  - Si los elementos son de línea la comparten sin hacer nada.
  - Si los elementos son de bloque:
    - Flotar: Siempre a la izquierda. La suma de los anchos no puede superar al ancho del contenedor.
    - Display:
      - Inline: No me respeta el ancho y el alto.
      - Inline-block: Me respeta el ancho y el alto pero me deja un margin entre los elemento que puedo quitar fácilmente.

- Box-sizing:border-box
  respeta el width y el heigth a pesar de que se introduzca padding o border. Hay que tenerlo en cuenta para poner el ancho de los elementos, te evita hacer muchos cálculos.

- Overflow:auto
  Cuando tengo un contenedor que tiene dentro elementos flotando tengo que poner esta propiedad al contenedor para que los elementos no se salgan del contenedor.
- Clear:both
  Se pone a un elemento que no quieres que flote con otros, rompe el float.

PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
display	Cambiarelemento de tipo bloque a tipo línea y alrevés.	None (no se muestra) Block (se muestra como elemento de bloque) Inline (se muestra como elemento de línea) Inline-block: (Se muestra como un elemento de línea al que se le puede dar width y heigth)	Dependiendo de cada elemento.	Todos los elementos	1
float	Posición de una caja	Right: La caja se mueve a su posición más a la derecha posible y el resto fluye a su izquierda. Left: La caja se mueve a su posición nás a la izquierda posible y el resto fluye a su derecha.	none	Todos los elementos	1
clear	Rompe el float	Rigth: rompe el float rigth Left: rompe el float left Both: rompe el float left y rigth	none	Todos los elementos	2
overflow	Indica que se hace con el contenido que excede las dimensiones de la caja	Visible: se muestra Hidden: no se muestra Scroll: se muestran barras de despazamiento. Auto: cuando un elemento con float sobrepasa el tamaño de la caja donde está contenido,al poner esta opción se adapta el tamaño de la caja al elemento float.	visible	Todos los elementos	2

visibility	Ver o no un elemento	Visible (se ve) Hidden (no se ve, pero se respeta su lugar)	visible	Todos los elementos	2
position	Posición de una caja	Static (normal) Relative (desplazamiento respecto a su posición normal : top, bottom, rigth, left reservando su espacio original) Absolute (desplazamiento : top, bottom, right,left respecto a su elemento contenedor posicionado no static) Fixed: Igual que el absolute pero permaneciendo fijo en la pantalla.	static	Todos los elementos	2
Left Top Right bottom	Posicionamiento de un elemento con position distinta de static	Medida o porcentaje.	0	Todos los elementos	2
z-index	Profundidad de las cajas. Tienen que tener un posicionamiento distinto de static.	Número: Las cajas con número más alto se muestran por encima de las de número más bajo.	auto	Todos los elementos	2

Position

Position **static** es el posicionamiento que tiene una caja por defecto, es decir si no pongo nada en mi caja, ésta tiene posicionamiento static. Cuando un elemento tiene este position, si es de bloque se coloca sólo en la línea (si no usas float) y si es de línea comparte la línea con otros elementos de línea. Cuando pones un margen superior por ejemplo a un elemento, afecta también a los elementos que tenga por debajo ya que todos se desplazan hacia abajo. Con position static es imposible que dos elementos se solapen, es decir que uno se ponga por encima de otro.

Si un elemento tiene en el atributo position un valor distinto de static, se dice que el elemento **está posicionado**.

Cuando a un elemento le ponemos position **relative**, pero nada más, el elemento se comporta igual que un static. Si a ese elemento con position relative le ponemos además atributos top, left, etc... el elemento se desplazará lo que le indiquemos en top y left, pero dejando el hueco que ocuparía si fuera static.

Cuando a un elemento le ponemos position **absolute** pero no le ponemos atributos left y top, se coloca en la esquina superior izquierda del primer elemento en el que esté contenido que esté posicionado (es decir, que tenga una posición distinta a static). Si todos los elementos en los que está contenido tienen position static, se colocará en la esquina superior izquierda del navegador. Los atributos top y left que le pongamos a un elemento con position absolute le hacen colocarse esas distancias respecto a la esquina superior izquierda que haya tomado como referencia.

Tanto los elementos con position relative, como absolute, se pueden solapar con otros. En este caso, para decidir qué elemento queda por encima de cual, se usa el atributo **z-index**. El atributo z-index puede tomar cualquier valor entero. Si dos elementos se solapan, se mostrará por encima el elemento que tenga el z-index más alto.

Cuando se establecen las propiedades display, float y position sobre una misma caja,su interpretación es la siguiente:

1. Si display vale none, se ignoran las propiedades float y position y la caja no se muestra en la página.
2. Si position vale absolute o fixed, la caja se posiciona de forma absoluta, se considera que float vale none y la propiedad display vale block para todos los elementos.
3. En cualquier otro caso, si float tiene un valor distinto de none, la caja se posiciona de forma flotante y la propiedad display vale block para todos los elementos.

6.4. Fuentes

PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
color	color	Color, inherit	Black	Todos los elementos	1
Font-family	Tipo de letra	Familia tipográfica (Arial,Verdana, Garamond...) Familia genérica (serif, sans-serif, cursive, fantasy, monospace)	Depende del navegador	Todos los elementos	1
Font-size	Tamaño de la letra del texto	Medida, porcentaje, inherit, xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller	Medium	Todos los elementos	1

Font-weight	Grosor de la letra	Normal, bold, bolder, lighter, 100,200,300,400,500,600,700, 800, 900, inherit	Normal	Todos los elementos	1
Font-style	Si es cursiva	Normal, italic, oblique, inherit	Normal	Todos los elementos	1
Font-variant	mayusculas	Normal, small-caps, inherit	Normal	Todos los elementos	1

Forma abreviada:
font-style font-variant font-weight font-size/line-height font-family
(font-size y font-family son obligatorias)

Familias genéricas, el navegador busca entre sus fuentes la más parecida a:
serif, equivale a New Roman.
sans-serif, equivale a Arial.
cursive, equivale a Comic Sans.
fantasy, equivale a Impact.
monospace, equivale a Courier New.

Para hacer que nuestra página utilice una fuente dada, primero nos definimos con @font-face el nombre de la nueva fuente diciendo dónde se encuentra ubicado el fichero .ttf, y después aplicamos esa nueva fuente al selector deseado.(CSS3)
Ejemplo:

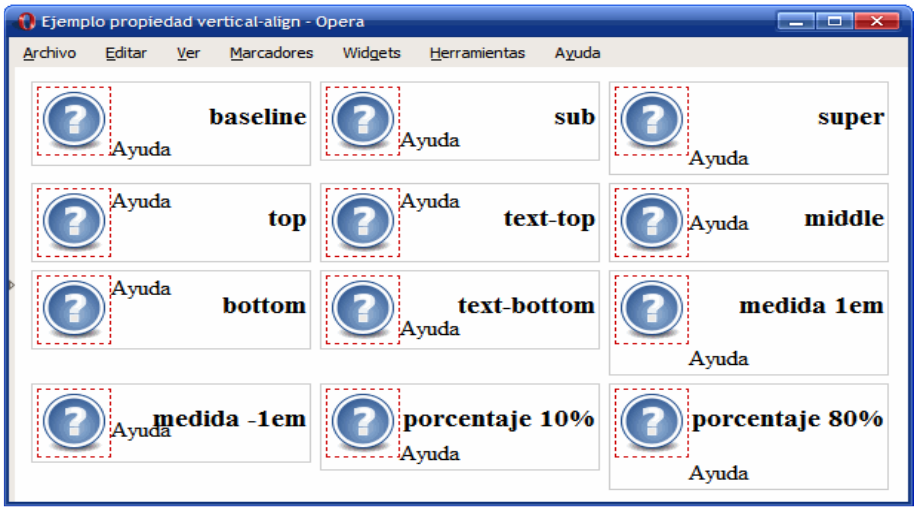
```
@font-face {font-family: mia; src: url('teutonic1.ttf'); }
body {font-family: mia;}
```

6.5. Texto

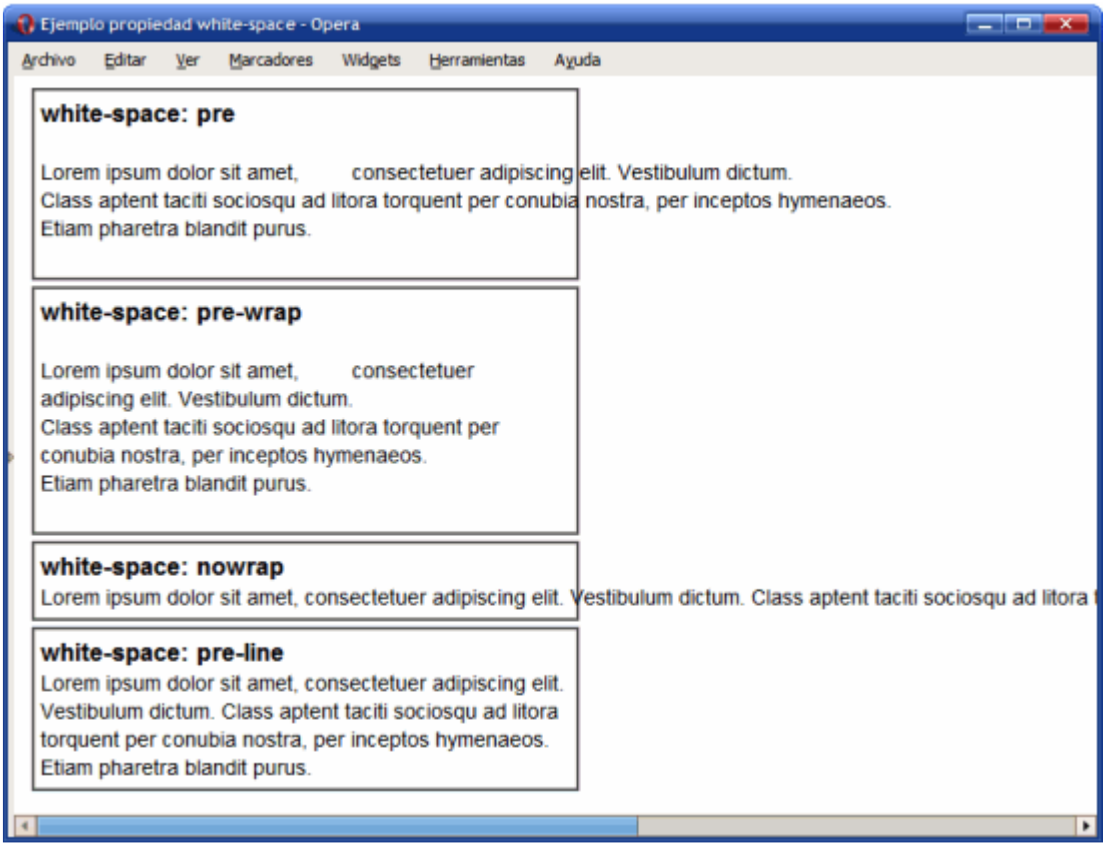
PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
Text-align	Alineación del texto e imágenes contenido en el elemento	Left, center,right,justify	left	Elementos de bloque y celdas.	1
Line-height	Interlineado	Medida, porcentaje, numero, inherit, normal	normal	Todos los elementos	1

Text-decoration	Subrayado, linea por encima, etc	None,Underline(subrayado), overline (linea por encima), blink (parpadeante), line-through (tachada), Inherit	none	Todos los elementos	1
Text-transform	Mayusculas, minúsculas...	None, capitalize (primera en mayusculas), lowercase (minuscula), uppercase(mayuscula)	none	Todos los elementos	1
Vertical-align	Alineación vertical	Baseline,sub,super,top,text-top,middle,bottom,text-bottom,orcentaje, medida,inherit	baseline	Elementos en línea y celdas.	1
Text-indent	Tabula la primera línea del texto.	Medida, porcentaje, inherit	0	Elementos de bloque y celdas	1
Letter-spacing	Espacio entre letras	Normal, medida, inherit	normal	Todos los elementos	1
Word-spacing	Espacio entre palabras	Normal, medida, inherit	normal	Todos los elementos	1
White-space	Tratamiento de los espacios en blanco	Normal,pre,nowrap,pre-wrap,pre-line,inherit	normal	Todos los elementos	1
Text-shadow	Sombra en el texto	4 valores: posición horizontal, posición vertical, difuminado y color	none	Todos los elementos	3

Valores de vertical-align:



Valores de white-space:



6.6. Listas

PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
List-style-type	Tipo de viñeta	None, disc ●, circle ◦, square ▪, decimal (1,2,3..) decimal-leading-zero (01,02..) lower-latin(a,b,c ..), Upper-latin(A,B,C...), lower-roman (i,ii,iii..), upper-roman(I,II,III..)	ul: disc ol:decimal	listas	1
List-style-position	Posición de la viñeta	Inside (dentro de la caja) Outside (fuera de la caja)	outside	listas	1
List-style-image	Imagen en lugar de viñeta	None (ninguna), url(imagen)	none	listas	1

6.7 Tablas

PROPIEDAD	DESCRIPCION	VALORES	DEFECTO	SE APLICA A	VER
Border-collapse	Indica si los bordes se juntan	Collapse:se juntan Separate:se separan	separate	table	2
Border-spacing	Separación entre bordes de celdas	Length length: horizontal vertical		table	2