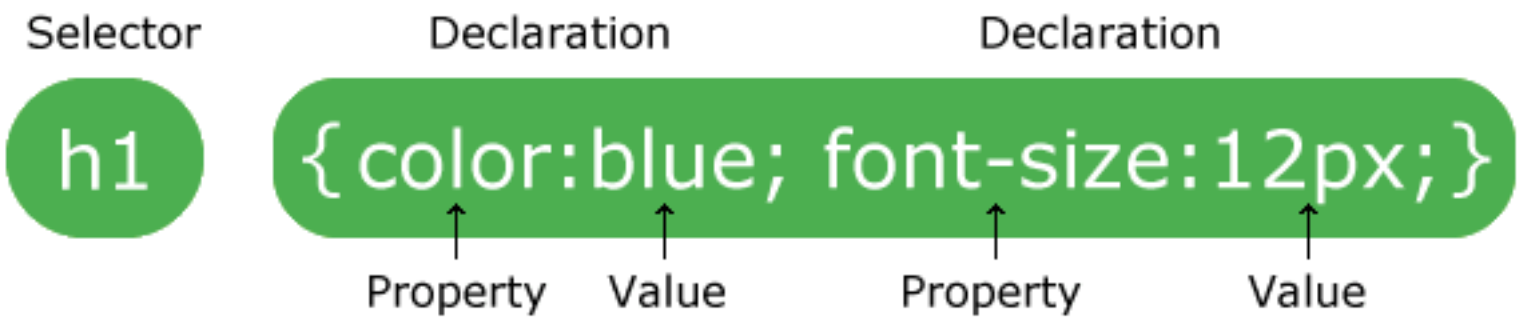
ESTILOS CSS


Selector
Declaration
Declaration
h1
{color:blue; font-size:12px;}
Property
Value
Property
Value

ESTILOS EN LÍNEA

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo style.

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este es el título del documento</title>
</head>
<body>
<h1 style="color:blue; text-
align:center;">Estilos </h1>
<p style=”font-size: 20px;
color:blue;”>Contenido del párrafo</p>
</body>
</html>
```

Atributo	Valores
background-color	color en inglés, en español, o el valor hexadecimal
color	color en inglés, color en español, o el valor hexadecimal
text-align	justify, center, right, left
font-family	arial, georgia, serif
font-size	14px

ESTILOS EMBEBIDOS

Una mejor alternativa es insertar los estilos en la cabecera del documento.

```
<style>
p{font: italic bold 25px arial, georgia, serif;}
</style>
</head>
<body>
<p>Mi texto</p>
</body>
</html>
```

ARCHIVOS EXTERNOS

Declarar los estilos en la cabecera del documento ahorra espacio y vuelve al código más consistente y actualizable, pero nos requiere hacer una copia de cada grupo de estilos en todos los documentos de nuestro sitio web.

La solución es mover todos los estilos a un archivo externo y luego utilizar el elemento <link> para insertar este archivo dentro de cada documento que los necesite.

Este método permite cambiar los estilos por completo incluyendo un archivo diferente.

También permite modificar o adaptar los documentos a cada circunstancia o dispositivo.

ARCHIVO EXTERNO

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este texto es el título del documento</title>
<link rel="stylesheet" href="misestilos.css">
</head>
<body>
<p>Mi texto</p>
</body>
</html>
```

REFERENCIAS

Existen varios métodos para seleccionar cuáles elementos HTML serán afectados por las reglas CSS:

Referencia por la palabra clave del elemento

Referencia por el atributo id

Referencia por el atributo class

REFERENCIA POR PALABRA CLAVE

Al declarar las reglas CSS utilizando la palabra clave del elemento afectamos cada elemento de la misma clase en el documento.

Por ejemplo, la siguiente regla cambiará los estilos de todos los elementos <p>:

```
p { font-size: 20px }
```

REFERENCIA CON EL ATRIBUTO ID

El atributo id es como un nombre que identifica al elemento.

El valor de este atributo **no** puede ser duplicado.

Debe ser **único** en todo el documento.

Referenciar:

La regla se declara con el símbolo # antes del valor que se usa para identificar el elemento:

```
<style>
#texto1
 { font-size: 20px }
</style>

<body>
<p id="texto1">Mi texto</p>
</body>
```

REFERENCIANDO CON EL ATRIBUTO CLASS

La mayoría del tiempo, en lugar de utilizar el atributo id para propósitos de estilos es mejor utilizar class.

Este atributo es más flexible y puede ser asignado a cada elemento HTML en el documento que comparte un diseño similar:

```
<style>
.texto1 { font-size: 20px }
</style>

<body>
<p class="texto1">Texto</p>
<p class="texto1">Texto</p>
<p>Texto</p>
</body>
```

REFERENCIANDO CON EL ATRIBUTO CLASS

La razón por la que debemos utilizar un punto delante del nombre de la regla es que es posible construir referencias más complejas.

Por ejemplo, se puede utilizar el mismo valor para el atributo class en diferentes elementos pero asignar diferentes estilos para cada tipo:

.titulo2{color:pink;}

<h2 class="titulo2">Hojas de estilo externas</h2>

p.texto1 { font-size: 20px }

SELECTOR UNIVERSAL

El selector universal (*) selecciona todos los elementos HTML de la página.

```
* {
text-align: center;
color: blue;
}
```

SELECTOR DE AGRUPACIÓN

El selector de agrupación selecciona todos los elementos HTML con las mismas definiciones de estilo.

```
h1 {
  text-align: center;
  color: red;
}
h2 {
  text-align: center;
  color: red;
}
```

```
p {
  text-align: center;
  color: red;
}
```

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS | Pseudoclases

PSEUDOCLASES

Los cuatro elementos <p> son hermanos entre sí e hijos del mismo elemento :

```
<div id=“conjunto">

<p class="mitexto1">Mi texto1</p>

<p class="mitexto2">Mi texto2</p>

<p class="mitexto3">Mi texto3</p>

<p class="mitexto4">Mi texto4</p>

</div>
```

Usando pseudo clases podemos aprovechar esta organización y referenciar un elemento específico sin importar cuánto conocemos sobre sus atributos y el valor de los mismos.

PSEUDOCLASES

```
p:nth-child(2)
{
background: #999999;
}
```

La pseudo clase es agregada usando dos puntos luego de la referencia y antes de su nombre. Esta regla puede incluir otras referencias.

Por ejemplo:

.miclase:nth-child(2) para referenciar todo elemento que es hijo de otro elemento y tiene el valor de su atributo class igual a miclase. La pseudo clase puede ser aplicada a cualquier tipo de referencia.

La pseudo clase nth-child() nos permite encontrar un hijo específico.

El documento HTML tiene cuatro elementos <p> que son hermanos.

Esto significa que todos ellos tienen el mismo padre que es el elemento .

Lo que esta pseudo clase está realmente indicando es algo como: “el hijo en la posición…” por lo que el número entre paréntesis será el número de la posición del hijo, o índice. La regla está referenciando cada segundo elemento <p> encontrado en el documento.

EJEMPLO 2

```
*{
margin: 0px;
}
p:nth-child(1){
background: #999999;
} p:nth-child(2){
background: #CCCCCC;
} p:nth-child(3){
background: #999999;
} p:nth-child(4){
background: #CCCCCC;
}
```

EJEMPLO 2

La primera regla usa el selector universal * para asignar el mismo estilo a cada elemento del documento.

Este nuevo selector representa cada uno de los elementos en el cuerpo del documento y es útil cuando necesitamos establecer ciertas reglas básicas.

En este caso, configuramos el margen de todos los elementos en 0 pixeles para evitar espacios en blanco o líneas vacías como las creadas por el elemento <p> por defecto.

En el resto del código usamos la pseudo clase nth-child() para generar un menú o lista de opciones que son diferenciadas claramente en la pantalla.

EJEMPLO 3

```
*{
margin: 0px;
}
p:nth-child(odd)/*impares*/
{
background: #999999;
}
p:nth-child(even)/*pares*/
{
background: #CCCCCC;
}
```

ODD

La palabra clave odd para la pseudo clase nth-child() afecta los elementos <p> que son hijos de otro elemento y tienen un índice impar.

La palabra clave even, por otro lado, afecta a aquellos que tienen un índice par.

EJEMPLO 4

```
*{
margin: 0px;
}
p:last-child
{
background: #999999;
}
```

Existen otras importantes pseudo clases relacionadas con esta última, como

first-child, **last-child** y **only-child**, algunas de ellas recientemente incorporadas.

La pseudo clase **first-child** referencia solo el primer hijo, **last-child** referencia solo el último hijo, y **only-child** afecta un elemento siempre y cuando sea el único hijo disponible.

Estas pseudo clases en particular no requieren palabras clave o parámetros.

EJEMPLO 5

```
:not(p)
{
margin: 0px;
}
```

La regla del ejemplo asignará un margen de 0 pixeles a cada elemento del documento excepto los elementos <p>.

A diferencia del selector universal utilizado previamente, la pseudo clase not() nos permite declarar una excepción.

OTROS SELECTORES

Hay algunos selectores más que fueron agregados o que ahora son considerados parte de CSS3 y pueden ser útiles para nuestros diseños.

Estos selectores usan los símbolos >, + y ~ para especificar la relación entre elementos.

```
div > p.mitexto2{
color: #990000;
}
```

El selector > está indicando que el elemento a ser afectado por la regla es el elemento de la derecha cuando tiene al de la izquierda como su padre.

La regla modifica los elementos <p> que son hijos de un elemento . En este caso, se referencia solamente el elemento <p> con el valor mitexto2 en su atributo class.

La pseudo-clase CSS :nth-child selecciona los elementos hermanos dentro del árbol del DOM, que coincidan con la condicion an+b-1, siendo n un numero natural, y teniendo un elemento padre.

Esto puede ser descrito con mayor claridad de esta manera: el elemento coincidente es el hijo bth de un elemento donde sus hijos han sido separados en grupos de a elementos cada uno.

Los valores de a y b deben de ser enteros, y el índice del elemento first-child seleccionado debe de ser 1.

Entre otras cosas, esto permite que los selectores coincidan con cada dos filas de una tabla.

EJEMPLOS

tr:nth-child(2n+1)

Representa las filas impares de una tabla HTML.

tr:nth-child(odd)

Representa las filas impares de una tabla HTML.

tr:nth-child(2n)

Representa las filas pares de una tabla HTML.

tr:nth-child(even)

Representa las filas pares de una tabla HTML.

EJEMPLOS

span:nth-child(0n+1)

Representa un elemento span el cual es el first-child del parent; lo que es lo mismo que el selector :first-child.

span:nth-child(1)

Equivalente al anterior.

span:nth-child(-n+3)

Representa a los tres primero elementos span.

MODELO DE CAJA

Los navegadores consideran cada elemento HTML como una caja. Una página web es en realidad un grupo de cajas ordenadas siguiendo ciertas reglas.

Estas reglas son establecidas por estilos provistos por los navegadores o por los diseñadores usando CSS.

CSS tiene un set predeterminado de propiedades destinados a sobrescribir los estilos provistos por navegadores y obtener la organización deseada.

Estas propiedades no son específicas, tienen que ser combinadas para formar reglas que luego serán usadas para agrupar cajas y obtener la correcta disposición en pantalla.

La combinación de estas reglas es normalmente llamada modelo o sistema de disposición. Todas estas reglas aplicadas juntas constituyen lo que se llama un modelo de caja.


UNIDADES DE MEDIDA

Para establecer las dimensiones, se utilizan unidades.

Tipos de unidades de medida:

Absolutas (in, cm, mm, pt, pc, px)

Porcentajes (x%)

Relativas (em, ex).

Las medidas relativas se adaptan mejor a todos los dispositivos.

UNIDADES DE MEDIDA ABSOLUTAS

Las unidades de medida en CSS de tipo absoluto hacen referencia a las unidades que no cambian, esas que en todos lo contextos son iguales.

- ***in:*** hace referencia a las pulgadas, que son iguales a 2.54cm.
- ***cm***: se refiere a los centímetros.
- ***mm***: hace referencia a los milímetros.
- ***q:*** se refiere a un cuarto de la unidad mm. 1q=0.248mm.
- ***pt***: un punto es igual a 1/72 de una pulgada o 0.35mm.
- ***pc***: una pica es igual a 12 puntos, o sea 4.23mm.
- ***px***: esta etiqueta se refiere a los píxeles que, aunque son absolutos (0.26mm), también son relativos a la densidad de la pantalla.

UNIDADES DE MEDIDA RELATIVAS

- ✓Las unidades de medida en CSS de tipo relativo dependen del elemento o factor al que hagan referencia.
- ✓Aunque pueden ser inicialmente más complejas, algunos prefieren las unidades de medida relativas porque los tamaños de los elementos dependen el uno del otro.
- ✓Esto hace que las proporciones entre los elementos se mantengan y todo encaje.

UNIDADES DE MEDIDA RELATIVAS

em

Esta unidad es relativa al tamaño de letra o font size establecida en el navegador. Su nombre es em porque el tamaño de letra se basa en el tamaño de la letra eme. A menos que haya sido modificada por el usuario, normalmente este tamaño es de 16px.

ex

Esta unidad es relativa a la altura de la «x» del elemento. También se conoce por ser más o menos la mitad del tamaño de la fuente del navegador o 0.5em.

ch

Conocido en inglés como zero width, esta unidad de medida es relativa al tamaño del ancho del cero en la fuente del navegador.

UNIDADES DE MEDIDA RELATIVAS

%

Esta unidad es relativa al tamaño del elemento padre.

rem

La unidad rem o root em es similar a la unidad em, pero, en vez de tomar como base el tamaño de letra del navegador, la unidad em toma el tamaño base del documento HTML. Este tamaño se personaliza bajo la etiqueta :root {font-size}. De este modo, podemos usar rem para dimensionar nuestros elementos con un múltiplo del tamaño base.

Esta unidad puede ser muy útil a la hora de dimensionar una página cuando el usuario hace zoom, pues los elementos serán relativos unos a otros y mantendrán su proporción.

em = ancho de la letra M mayúscula de la fuente que se esté utilizando.

ex = alto de la letra x de la fuente que se esté utilizando.

UNIDADES DEL VIEWPORT

Las siguientes unidades son relativas al viewport, que es el espacio o trozo de pantalla donde se renderiza y se muestra la página web. Estas unidades sirven para dimensionar la pantalla y organizar los elementos dentro de esta.

vw

Como sigla de la unidad en inglés viewport width, esta unidad es relativa al ancho del viewport.

vh

Como sigla de la unidad en inglés viewport height, esta unidad es relativa a la altura del viewport.

vmin

Esta unidad, también conocida como viewport minimum, es relativa al factor que sea más pequeño entre el ancho y al alto del viewport.

vmax

Esta unidad, conocida como viewport maximum, es relativa al factor que sea más grande entre el ancho y el alto del viewport. Junto con vmin, esta unidad puede ser muy útil si queremos que nuestros diseños sean flexibles y se adapten al tamaño visible de la página web.

EM

Un *em* es el tamaño de una letra "**M**" del elemento al cual se esté aplicando esta medida. Es decir, si el elemento tiene aplicado un tamaño de fuente de 16 pixeles, entonces 1 em será igual a 16px (los navegadores de manera predeterminada definen un *font-size* de 16px al elemento HTML, por lo tanto, por defecto 1em es igual a 16px).

La unidad *em* es escalable y siempre depende de su elemento padre. Por ejemplo, si el elemento body tiene un tamaño de fuente de 16px y un elemento hijo tiene una fuente con tamaño 1.3em, este texto se mostrará de un tamaño un 30% más grande que el del body (20.8px), mientras que si dentro de ese elemento tenemos otro hijo con un *font-size* de 1.3 em, el tamaño de fuente de este objeto sería un 30% más grande que el tamaño de su padre (27.04px).

Body = 1em (*16px*)
Hijo = 1.3em (*16px* x *1.3* = *20.8px*)
Nieto = 1.3em (*20.8px* x *1.3* = *27.04px*)

Es recomendable usar la unidad de medida *em* para definir los tamaños de fuente, los altos de línea y también para elementos de diseño que no requieran ser muy exactos o que requieran una medida que tenga relación con el tamaño del texto, como por ejemplo el margen entre párrafos, el relleno interior de los *blockquotes*, etc.

También se puede aplicar a elementos generales del layout aunque no es muy recomendable, ya que si eventualmente se cambia el tamaño de fuente de uno de ellos, se podría estropear el diseño.

REM

La unidad de medida *rem* es muy similar a *em*, con la única diferencia de que no es escalable, esto quiere decir que no depende del elemento padre, sino del elemento raíz del documento, el elemento HTML.

Rem significa "*Root Em*", es un *em* basado en la raíz.

Esto significa que si el elemento HTML tiene un tamaño de fuente de 16px (como es por defecto), entonces 1rem, sería igual a 16px, y si queremos aplicar un tamaño basado en *rem* a cualquier elemento de la página, no importará cual sea el tamaño de fuente que tenga asociado ese elemento, ya que 1 rem siempre será igual a 16 pixeles a no ser que se modifique el elemento raíz.

EJEMPLO

Unidades de medida

```
body {
    font-family: sans-serif;
    color: #666;
    background: #f4f4f4;
    line-height: 1.6em;
}

h1 {
    margin: 0 0 1em 0;
    line-height: 1.4em;
}

p {
    margin: 1em 0;
}
    p:last-child {
        margin-bottom: 0;
    }

img {
    max-width: 100%;
    height: auto;
```

UNIDADES DE MEDIDA ABSOLUTAS

Unidad (absoluta)	Descripción
%	Porcentaje. El tamaño actual de la fuente es el 100% (escalable)
cm	Centímetros
in	Pulgadas
mm	Milímetros
pc	Picas (1pc = 12 pt)
pt	Puntos (1pt = 1/72 pulgada)
px	Pixel del dispositivo

UNIDADES DE MEDIDA RELATIVAS

Unidad (relativa)	Descripción
ch	Ancho del glifo cero “0” de la fuente
em	Tamaño de la fuente actual. Altura de la M de la fuente del elemento actual
ex	Altura de la x de la fuente del elemento actual
gd	La gradilla definida por layout-grid. Si no está definida, equivale a em
rem	El tamaño de la fuente del elemento raíz
vh	1% Altura del dispositivo de visionado (variable dinámicamente)
vw	1% Anchura del dispositivo de visionado (variable dinámicamente)
vmin (vm)	El menor de los valores, 1% altura o anchura, del dispositivo de visionado (variable dinámicamente)
vmax	El mayor de los valores, 1% altura o anchura, del dispositivo de visionado (variable dinámicamente)

MEDIA QUERIES

Consulta de medios y responsive design

PANTALLA ENTRE 400 Y 700 PIXELES DE ANCHO

```
@media screen and (min-width: 400px) and (max-width: 700px)
        {
        body{
                background-color:red;
                font-family: Times New Roman;
                }
        header h3{
                text-align:right;
                }
        }
```

ANCHO DEL DISPOSITIVO: 800 PX

```
@media screen and (device-width: 800px)
        {
        body{     background-color:blue;
                font-family: verdana;
                }
        header h3
                  {  text-align:center;
                   }
        }
```

ORIENTACION
PORTRAIT (RETRATO) VERTICAL ALTURA>ANCHURA
LANDSCAPE (PAISAJE) HORIZONTAL ALTURA<ANCHURA

```
@media all and (orientation:portrait)
{           body{      background-color:yellow;
                       font-family: arial;
                  }
}
@media all and (orientation:landscape)
{           body{     background-color:orange;
                      font-family: arial;
                      }
 }
```

ALTO DEL DISPOSITIVO: 600PX

```
@media screen and (device-height: 600px)
{
body{
        background-color:purple;
        font-family: arial;
        }
header h3
{
text-align:left;
}
}
```