

ESTILOS CSS



ESTILOS

- A pesar de que cada navegador garantiza estilos por defecto para cada uno de los elementos HTML, estos estilos no necesariamente satisfacen los requerimientos de cada diseñador.



ELEMENTOS BLOCK

- A pesar de que cada navegador garantiza estilos por defecto para cada uno de los elementos HTML, estos estilos no necesariamente satisfacen los requerimientos de cada diseñador.
- Con respecto a la estructura, básicamente cada navegador ordena los elementos por defecto de acuerdo a su tipo:
- block(bloque) o inline (en línea).
- Esta clasificación está asociada con la forma en que los elementos son mostrados en pantalla.



ELEMENTOS BLOCK

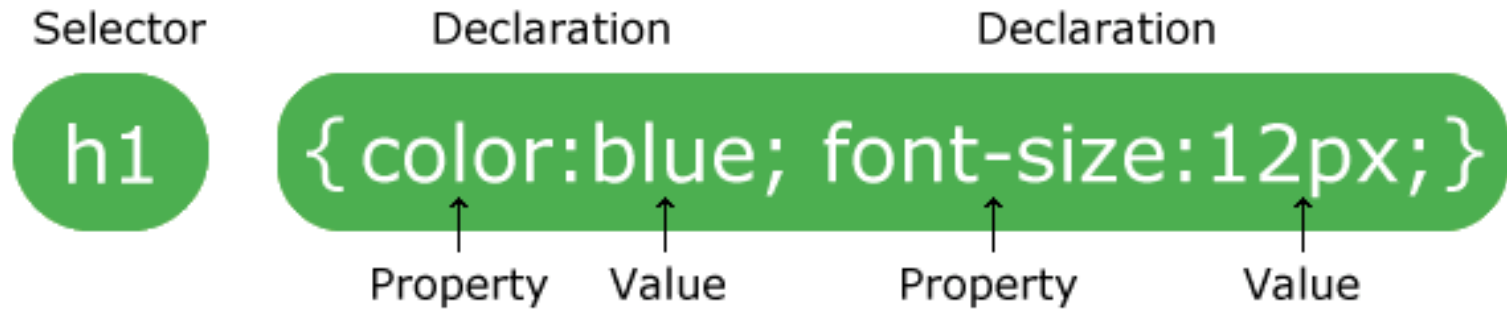
- Los elementos block son posicionados uno sobre otro hacia abajo en la página.
- Los elementos inline son posicionados lado a lado, uno al lado del otro en la misma línea, sin ningún salto de línea a menos que ya no haya más espacio horizontal para ubicarlos.



- Casi todos los elementos estructurales en nuestros documentos serán tratados por los navegadores como elementos block por defecto.
- Esto significa que cada elemento HTML que representa una parte de la organización visual.
- Por ejemplo:
`<section>`, `<nav>`, `<header>`, `<footer>`, `<div>`
será posicionado debajo del anterior.



CSS



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>`
- `<p style="font-size: 20px">Mi texto</p>`
- `</body>`
- `</html>`



ATRIBUTOS

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



ESTILOS EMBEBIDOS

- Una mejor alternativa es insertar los estilos en la cabecera del documento y luego usar referencias para afectar los elementos HTML correspondientes.



ESTILOS EMBEBIDOS

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Este texto es el título del documento</title>
<style>
p { font-size: 20px }
</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 nos permite cambiar los estilos por completo simplemente incluyendo un archivo diferente.
- También nos permite modificar o adaptar nuestros documentos a cada circunstancia o dispositivo, como veremos al final del libro.



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



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 }
```



REFERENCIANDO CON EL ATRIBUTO ID

- El atributo id es como un nombre que identifica al elemento. Esto significa que el valor de este atributo no puede ser duplicado.
- Este nombre debe ser único en todo el documento.
- Para referenciar un elemento en particular usando el atributo id desde nuestro archivo CSS la regla debe ser declarada con el símbolo # al frente del valor que usamos para identificar el elemento:



REFERENCIANDO CON EL ATRIBUTO ID

```
#texto1 { font-size: 20px }
```

```
<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:



REFERENCIANDO CON EL ATRIBUTO CLASS

```
.texto1 { font-size: 20px }
```

```
<body>
```

```
<p class="texto1">Mi texto</p>
```

```
<p class="texto1">Mi texto</p>
```

```
<p>Mi 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:
- `p.texto1 { font-size: 20px }`



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)
 - Porcentajes (x%)
 - Relativas (px, em, ex).
- Las medidas relativas se adaptan mejor a todos los dispositivos.
- px = píxel (relativa a los puntos de la pantalla en la que se visualice la página).
- 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.



EM

- Un *em* es el tamaño de una letra “**M**” del elemento al cual se esté aplicando esta medida. Es decir, si en 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).



EM

- **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 *blockquote*s, 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)



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
px	Pixel del dispositivo
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 PÍXELES 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;
  }
}
```



	Resolución	%	Histograma
1	1920x1080 16:9 HD 1080	19.43%	
2	1366x768 HD	17.40%	
3	1536x864	5.38%	
4	1600x900 16:9 HD+ 900p	4.72%	
5	360x640	4.63%	
6	1440x900 8:5 WSXGA	4.57%	
7	1024x768 4:3 XGA	3.66%	
8	1360x768	3.14%	
9	1280x1024 5:4 SXGA	3.06%	
10	1280x720 16:9 HD 720	2.85%	
11	1680x1050 8:5 WSXGA+	2.69%	
12	2560x1440	2.08%	
13	1280x800 8:5 WXGA	2.03%	
14	800x600 4:3 SVGA	1.65%	



Resolución (píxeles)	Porcentaje
1366 x 768 – HD	23%
1920 x 1080 – FHD	10%
1024 x 768 – XGA	9%
1280 x 1024 – SXGA	8%
1440 x 900 – WSXGA	7%
1280 x 800 – WXGA	5%
1600 x 900 – HD+900p	5%
1680 x 1050 – WSXGA+	4%
768 x 1024	4%

