

Desenvolvimento Web

Fundamentos do CSS



Índice

Introdução	3
Sintaxe da etiqueta CSS	4
Onde vai o código CSS?	4
Declaração externa	4
Declaração interna	4
Declaração online	5
Comentários	5
Fontes personalizadas	5
Modelo de caixa-contentor	6
Float, margin e padding	7
Selecionadores	8
Seletor de etiquetas	8
Seletor de classe	8
Seletor de ID	9
Seletor do grupo	10
Seletor universal	10
Seletor de irmãos	10
Seletor filho	11
Estilos para o texto	11
Cores	11
Cores RGBA	11
Gradientes	12
Estilos para links	12
Unidades no CSS	13
Comprimentos absolutos	13
Comprimento relativo	14

Introdução

A razão de ser do **CSS** é dar às nossas páginas uma certa aparência.

Antes do aparecimento do **CSS**, o que era feito era escrever tags **HTML** com seus atributos. Por exemplo, altura, largura, cor... etc.

Atualmente separamos a estrutura (**HTML**) da aparência (**CSS**).

Sintaxe da etiqueta CSS

Todas as etiquetas **CSS** terão duas partes:

- **Um seletor:** que determinará a área da página WEB que queremos modificar, ou seja, a etiqueta a ser modificada.
- **As instruções:** Eles se referem ao formato que vamos aplicar a essa zona.

Onde vai o código CSS?

Para aplicar as regras **CSS** à nossa página **HTML**, temos três opções:

- **Declaração externa.** Em um arquivo separado com extensão .CSS que será "linkado" à nossa página HTML. Desta forma, podemos modificar todas as páginas do meu site. Esta é a melhor opção para melhorar o posicionamento SEO.
- **Aplicação em estilo interno.** No cabeçalho do documento HTML dentro da **head**. Desta forma, modificaremos a aparência de uma única página da web em particular.
- **Estilo in-line.** Dentro de uma etiqueta. Com isso, modificamos a aparência do conteúdo dessa etiqueta. Opção não recomendada, SEO não lhe parece favorável. O ideal é utilizá-lo apenas em casos muito específicos.

Ao aplicar estilos em linha, temos que repetir a aplicação de estilos em cada uma das linhas que queremos modificar e cada vez que afetamos apenas uma linha. Com a aplicação de estilos internos é suficiente declarar uma vez o estilo e o tipo de elemento ao qual é aplicado, e será automaticamente aplicado a todos os elementos desse tipo dentro da página web.

Declaração externa

Consiste em criar uma página com extensão CSS e "linkarla" à nossa página HTML.

Uma vez criada a página CSS, vamos para o head de nossa página HTML e a ligamos com uma <link> tag.

Em nosso cabeçalho, ou seja, na <head> de nossa página, vamos colocar:

```
<link rel="stylesheet"
href="Hoja_Estilos.css">
```

Onde **Hoja_Estilos.css** será o arquivo .css que dará a aparência à nossa página web e onde colocaremos as etiquetas CSS.

A partir deste momento podemos usar os seletores para fazer as modificações que nos interessam nas tags de nossa página HTML.

Este seria um exemplo de <head> com a etiqueta .css anexada:

```
<head>
  <meta charset="UTF-8">
  <title>Ejemplo de HTML</title>
  <link rel="stylesheet"
href="Hoja_Estilos.css" >
</head>
```

Declaração interna

Para criar uma folha de estilo interna, teremos que usar a tag <style></style> no **head** de nossa página HTML. No interior, criaremos nossos estilos CSS usando os seletores CSS disponíveis no CSS.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    h1 {
      text-align: center;
      font-family: "Lucida Grande";
      font-size: 32px
    }
  </style>
</head>
```

Desta vez dissemos `html` que as tags `h1` terão que ter um alinhamento centralizado, a fonte será "*Lucida Grande*" e terá um tamanho de fonte de 32px. Desta forma, todas as etiquetas `h1` em nossa página terão este formato.

Declaração online

Desta vez usaremos `<style></style>` tags em qualquer tag em nosso documento HTML. O estilo que escolhemos só alterará essa etiqueta.

```
<body>
  <h1 style="color: aqua">Titulo de
la página</h1>
  <h2>Subtítulo de la página</h2>
  <h2>Subtítulo2 de la página</h2>
</body>
```

Neste exemplo, modificamos o estilo de apenas uma tag `h1` em particular, mudando sua cor.

Caso uma etiqueta seja afetada por vários estilos de CSS, o navegador aplicará os seguinte critério:

Declaração on-line > Declaração interna > Declaração externa

Comentários

No CSS, a sintaxe para comentários é;

```
/*Aquí va el comentario*/
```

Como em qualquer linguagem de programação, os comentários serão ignorados ao executar o código. São simplesmente anotações que o HTML não levará em conta.

Fontes personalizadas

Se quisermos usar um tipo de fonte específico que tenha que ser baixado, temos que usar a propriedade `@font-face` e temos que especificar o caminho onde o arquivo do true-type dessa fonte está localizado:

```
@font-face {
  font-family: 'miFuente';
  src: url('Bauhaus.ttf');
}
```

Se a fonte estiver localizada localmente, ou seja, em nosso disco rígido, ela tem que estar na raiz. Se o colocarmos em uma pasta, normalmente não funciona.

E então usaremos o tipo de fonte que criamos:

```
p{
  font:bold 32px 'miFuente';
}
```

Nesta ocasião, dissemos `html` para usar esta fonte particular para `<p>` tags.

Modelo de caixa-contentor

Há dois tipos de tags em HTML:

- **Etiquetas block.** Elas causam uma quebra de linha acima e abaixo da etiqueta e ocupam 100% da largura da página. A maioria das etiquetas de estrutura são deste tipo.
- **Etiquetas in-line.** Eles não geram uma quebra de linha.

MODELO CAJA-CONTENEDOR

- Etiquetas BLOCK

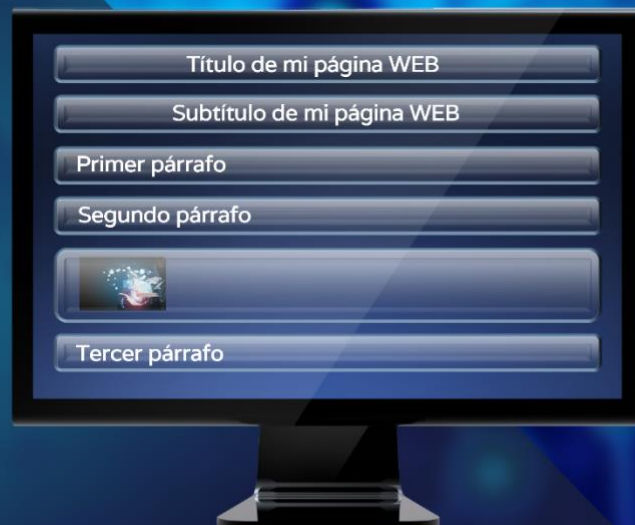
Provocan un salto de página por arriba y por debajo de la etiqueta y ocupan el 100% del ancho de la página. La mayoría de etiquetas de estructura son de este tipo.

```
<h1>Titulo de mi página WEB</h1>
<h2>Subtitulo de mi página WEB</h2>
<p>Primer párrafo</p>
<br>
<p>Segundo párrafo</p>
<div></div>
<br>
<p>Tercer párrafo</p>
```

- Etiquetas IN-LINE

No generan salto de línea. No se encuadran en un contenedor.

`` `<a>` `<image>`



Para fazer um elemento de **"block"** se torne um **"in-line"** usamos o imóvel:

`display: inline`

Float, margin e padding

Ao trabalhar com a posição dos elementos de nossa página HTML, há várias propriedades a serem conhecidas:

Float: Altera a forma como os elementos são colocados em uma página da web. Ao aplicar este imóvel, poderemos colocar um elemento onde quisermos na página, ele não ocupará mais toda a largura da página e o resto dos elementos será distribuído ao redor dele, em vez de acima ou abaixo dele. Para que o elemento retorne a ter propriedades de caixa-contendor, usamos a propriedade `clear`.

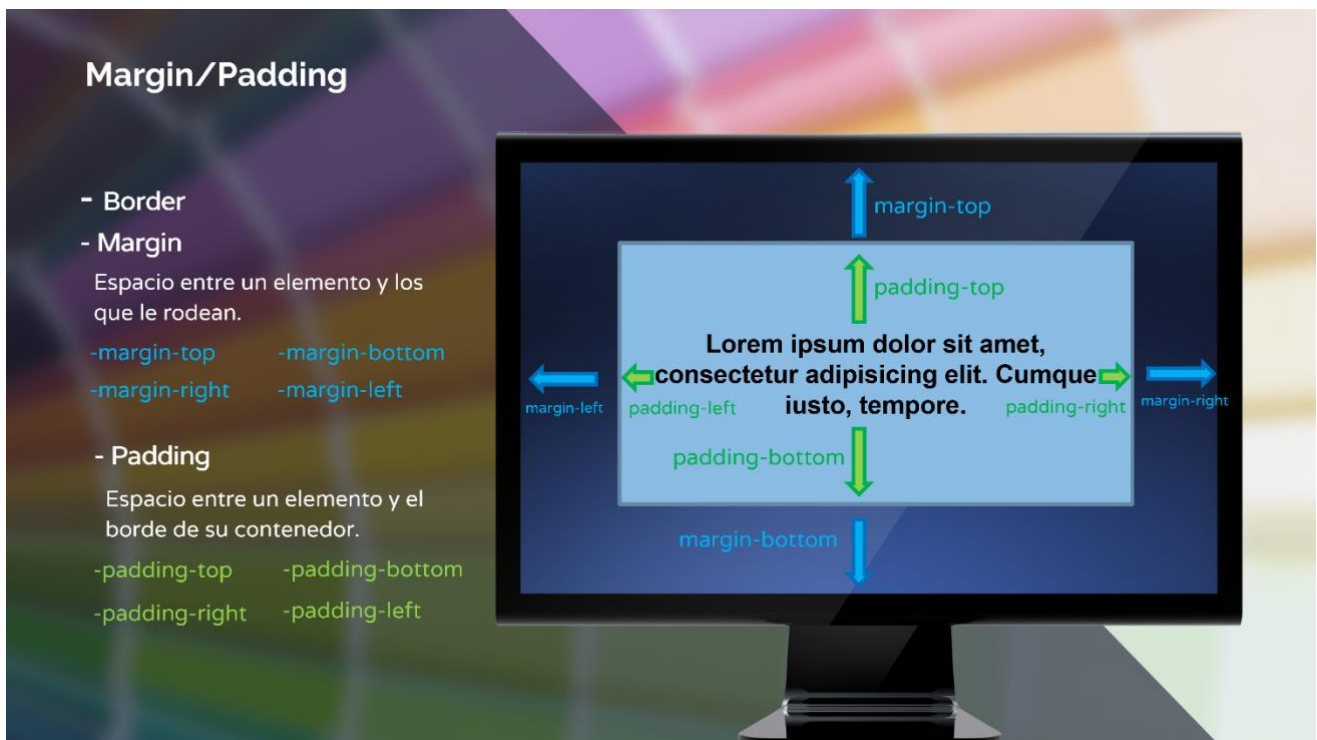
Margin: Define o espaço entre a borda da caixa-contendor e o que quer que esteja ao redor dela (elementos com propriedade de `float` não contam para isso).

Cada elemento de nosso site terá uma margem no topo (`margin top`), da direita (`margin right`), embaixo de (`margin bottom`) e, à esquerda (`margin left`).

Padding: Define o espaço entre o conteúdo da caixa-contendor e sua borda.

Análogo à propriedade do imóvel `margin`, cada elemento de nosso site terá um acolchoamento na parte superior (`padding top`), da direita (`padding right`), embaixo de (`padding bottom`) e, à esquerda (`padding left`).

Border: Modifica a borda de um elemento, por exemplo, uma imagem, uma caixa de texto, etc.



Selecionadores

No CSS, os seletores são padrões usados para selecionar os elementos que você deseja modificar.

Os seguintes seletores estão disponíveis no CSS.

- Selecionadores de rótulos.
- Selecionadores de classe.
- Selecionador de identificação.
- Selecionador de grupo.
- Selecionador universal.

Há mais tipos de selecionadores, mas estes são os mais usados.



Quando se trata de criar nossos estilos, as instruções são divididas em dois grupos:

- **Selecionadores:** Eles estão acostumados a especificar dentro de nossa página web, a quais elementos da mesma vamos aplicar o estilo.
- **Bloco de declaração:**

Selecionador ➡ `p{color:#3333}` ➡ Bloco de declarações.

Seletor de etiquetas

Ele muda o estilo de todos os elementos do tipo de etiqueta que especificamos.

Ela pode ir em uma folha de estilo externa ou interna, na interna iria na head.

Sua sintaxe em uma folha interna seria a seguinte:

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    p{color: aqua}
  </style>
</head>
```

Desta vez, mudamos a cor de todas as etiquetas `<p>` em nossa página.

Em uma folha de estilo externa (um arquivo **CSS**):

```
p{color: aqua}
```

Isto também mudaria o estilo de todas as etiquetas `<p>` na página.

Seletor de classe

Com eles, aplicamos um formato de estilo a um ou mais elementos específicos. Pode ser em uma folha de estilo externa ou na [head](#) de uma folha de estilo interna, então a aplicamos ao elemento que necessita.

Começa sempre com um ponto ".", seguido do nome que queremos.

Nota: O nome tem que começar com uma letra e é case sensitive, ou seja, é sensível a maiúsculas e minúsculas.

Folha de estilo interno:

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    .miEstilo {color: aqua}
  </style>
</head>
```

Folha de estilo externo:

```
.miEstilo {color: aqua}
```

Então, na etiqueta (ou rótulos) em questão, colocamos;

```
<p class= "miEstilo ">Párrafo</p>
```

Neste caso, aplicamos o estilo a uma etiqueta `<p>`, se quiséssemos aplicá-lo a uma única palavra ou a qualquer outro elemento não marcado na página, o faríamos com a etiqueta ``. Esta etiqueta não faz nada, mas me permite colocar um identificador e estilo, por exemplo:

```
Vamos a resaltar una <spam class=
"miEstilo">palabra</spam> de nuestro
texto
```

Podemos atribuir várias classes a um elemento:

```
<p class="clase1 clase2"></p>
```

Seletor de ID

É semelhante ao seletor de rótulos, mas este se aplica apenas a um elemento específico para diferenciá-lo dos outros. Em vez de começar com um ponto, ele começa com um `#`.

Podemos dar-lhe um nome, não é essencial.

Pode ser em uma folha de estilo externa ou no `head` de uma folha de estilo interna.:

Folha interna:

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    #miEstilo {color: aqua}
  </style>
</head>
```

Em uma folha externa:

```
#miEstilo {color: aqua}
```

Então, na etiqueta (ou etiquetas) em questão, colocamos;

```
<p id= "miEstilo ">Párrafo</p>
```

Lembre-se de que **um elemento pode ter apenas uma ID e que cada ID deve ser única.**

Seletor do grupo

Com ela, aplicamos algumas características a todas as etiquetas que escolhemos.

Em nossa folha de estilo, colocaremos:

```
P[name="miEstilo"] {
  color: aqua
}
```

Então, na etiqueta (ou etiquetas em questão, colocamos;

```
<p name="miEstilo">Párrafo</p>
```

Há três modificadores disponíveis aqui para especificar a quais atributos o estilo será aplicado. Eles seguiriam imediatamente o **name**:

^ : Aplica-se a todos os elementos que começam com...

```
<p name^= "mi">Párrafo</p>
```

\$: Aplica-se a todos os elementos que terminam em...

```
<p name$= "ilo">Párrafo</p>
```

***** : Aplica-se a todos os itens que contenham...

```
<p name*= "Est">Párrafo</p>
```

Seletor universal

Aplique as características desejadas a todos os elementos da página *web*.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    * {color: aqua}
  </style>
</head>
```

Neste caso, aplicamos a cor "aqua" a todos os elementos de nosso website.

Seletor de irmãos

Ela modifica os elementos que seguem o que eu lhes digo para modificar.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style type="text/css">
    h1+p {color: aqua}
  </style>
</head>
```

Isto mudaria a cor de <p> parágrafos após uma <h1> tag.

Seletor filho

Aplica um estilo às etiquetas que são elas próprias dentro de outras etiquetas. Tomando como pai a etiqueta contendo e como filha a etiqueta contida dentro dela.

```
<head>
  <meta charset="utf-8">
  <title>Pagina2</title>
  <meta name="Description"
content="Pagina2">
  <meta name="Keywords" content="Curso,
HTML, CSS, programacion WEB">
  <style>
    Li a {color: aqua}
  </style>
</head>
```

O que conseguimos aqui é aplicar o estilo apenas aos links `<a>` que estão dentro de uma etiqueta ``, excluindo todos os outros links que possam existir na página.

Nota: Se eu quiser estilizar apenas uma palavra ou um grupo de palavras dentro de um texto, lembre-se que temos que usar a tag ``

Estilos para o texto

Cores

Para mudar as cores de um elemento de nossa página web, temos várias opções:

- **Cores pré-definidas:** Teremos acesso direto digitando seu nome em inglês.:
`p {color:blue}`
- **Rgb (vermelho, verde, azul):** Onde vermelho, verde e azul são números inteiros desde 0 a 255 ou porcentagens de 0% a 100%:
`p {color:rgb(42,44,156)}`
- **#RGB:** Onde R, G e B são números hexadecimais de 0 a F:
`p {color: #353E9A}`

Cores RGBA

Os valores de cor RGBA são uma extensão dos valores de cor RGB com um canal alfa, que especifica a opacidade do objeto.

Uma cor RGBA é especificada com a função `rgba()`, que tem a seguinte sintaxe:

`rgba(red, green, blue, alpha).`

O parâmetro alfa é um número entre 0,0 (totalmente transparente) e 1,0 (totalmente opaco).

```
#p1 {background-color: rgba(255, 0, 0,
0.3);} /*Rojo con transparencia*/
#p2 {background-color: rgba(0, 255, 0,
0.3 /*Verde con transparencia*/ #p3
{background-color: rgba(0, 0, 255, 0.3);}
/*Azul con transparencia*/
```

Gradientes

Podemos aplicar um gradiente de cor a um elemento. Neste caso devemos especificar em ordem, tipo de gradiente, direção do gradiente, cor inicial e cor final.

```
div {
  background: linear-gradient(to
bottom, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to right,
#33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to top
left, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to top
right, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to top,
#33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to left,
#33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to bottom
left, #33ccff 0%, #ff99cc 100%);
}

div {
  background: linear-gradient(to bottom
left, #33ccff 0%, #ff99cc 100%);
}
```

Estilos para links

Uma etiqueta de link `<a>`;

```
<a href="http://www.google.es">Ir a
google</a>
```

Pode ter diferentes estatutos:

- **Link**; Em repouso, eles estão apenas aparecendo na página, não estamos nos masturbando por cima deles, etc.
- **Visited**; Quando o link já tiver sido visitado e estiver no cache do navegador.
- **Hover**; Quando o mouse está pairando sobre ele. Isto também pode ser usado em outro lugar na página, por exemplo, para destacar um parágrafo quando pairando sobre ele.
- **Active**; Quando você está fazendo "click".

Se todos os quatro estados forem usados ao mesmo tempo, devem ser usados nesta ordem, caso contrário não funcionarão. (regra **LoVe HAte**). Em outras palavras, **Link** => **Visited** => **Hover** => **Active**

A sintaxe para dar um status ao nosso estilo de link:

```
a:estado{....}
```

```
a:visited{color:red}
```

Muda a cor de todos os links já visitados para vermelho.

Unidades no CSS

O CSS tem várias unidades diferentes para expressar um comprimento, pixels, porcentagens e em escala. Confira o vídeo neste tópico para saber mais sobre eles.

Muitas propriedades do CSS assumem valores de "comprimento", tais como `width`, `margin`, `padding`, `font-size`, etc.

O comprimento é um número seguido por uma unidade de comprimento, tal como `10px`, `2em`, etc.

Diferentes valores de comprimento são definidos, usando px (pixels):

```
h1 {  
  font-size: 60px;  
}  
  
p {  
  font-size: 25px;  
  line-height: 50px;  
}
```

Nota: Um espaço em branco não pode aparecer entre o número e a unidade. Entretanto, se o valor for `0`, a unidade pode ser omitida.

Para algumas propriedades do CSS, são permitidos comprimentos negativos.

Há dois tipos de unidades de comprimento: **absoluto** e **relativo**.

Comprimentos absolutos

As unidades de comprimento absoluto são fixas e um comprimento expresso em qualquer uma delas aparecerá exatamente como aquele tamanho.

As unidades de comprimento absoluto não são recomendadas para uso na tela, porque os tamanhos das telas variam muito. Entretanto, eles podem ser usados se o meio de saída for conhecido, por exemplo, para layout de impressão.

Unidad	Descripción
cm	centímetros
mm	milímetros
in	Polegadas (1in = 96px = 2.54cm)
px *	pixels (1px = 1/96th of 1in)
pt	Puntos (1pt = 1/72 of 1in)
pc	picas (1pc = 12 pt)

Comprimento relativo

As unidades de comprimento relativo especificam um comprimento em relação a outra propriedade de comprimento. As unidades de comprimento relativo escalam melhor entre os diferentes meios de renderização.

Unidades	Descripción
em	Em relação ao tamanho da fonte do elemento (2em significa 2 vezes o tamanho da fonte atual)
ex	Em relação à altura x da fonte atual (raramente utilizada)
ch	Em relação à largura do "0" (zero)
rem	Em relação ao tamanho da fonte do elemento raiz
vw	Em relação a 1% da largura da janela gráfica *
vh	Em relação a 1% da altura da janela gráfica *
vmin	Relativo a 1% da menor dimensão da janela gráfica
vmax	Em relação a 1% da maior dimensão da janela gráfica
%	Relativo ao elemento pai