

Aula 02 - Introdução ao css (folhas de estilo)

CSS é a sigla para o termo em inglês *Cascading Style Sheets* que, traduzido para o português, significa Folha de Estilo em Cascatas. O CSS é fácil de aprender e entender e é facilmente utilizado com as linguagens de marcação HTML ou XHTML.

Vantagens do CSS

A diferença entre um site que implementa CSS e outro que não o usa é gigantesca e notável.

Você já deve ter visto um site que não carrega completamente ou tem um plano de fundo branco com texto azul e preto. Isso significa que a parte CSS do site não foi carregada corretamente ou não existe.

E é assim que um site somente com HTML se parece. Acredito que você vai concordar comigo de que isso não é muito bonito, certo?

Antes de usar CSS, toda a estilização tinha que ser incluída na marcação HTML. Isso significa que você deve descrever separadamente todo o plano de fundo, as cores das fontes, os alinhamentos, etc.

Antes de usar CSS, toda a estilização tinha que ser incluída na marcação HTML. Isso significa que você deveria descrever separadamente todo o plano de fundo, as cores das fontes, os alinhamentos, etc.

Mas o CSS permite que você estilize tudo em um arquivo diferente, criando assim o estilo separadamente. E, mais tarde, faça integração do arquivo CSS na parte superior da marcação HTML.

Isso mantém a marcação HTML limpa e fácil de manter. Resumindo, com o CSS você não precisa mais escrever repetidamente como os elementos individuais se parecem. Isso economiza tempo, encurta o código e diminui a chance de erros.

O CSS permite que você tenha vários estilos em uma página HTML, tornando as possibilidades de personalização quase infinitas. Hoje em dia, isso está se tornando mais uma necessidade do que um simples recurso.

Como CSS Funciona

O CSS é uma ferramenta muito potente que possibilita criar diversas funcionalidades ao invés de usar JavaScript ou outra linguagem mais pesada. Se usado com moderação, CSS pode viabilizar uma ótima experiência ao desenvolvedor e usuários das páginas web.

Com o *Cascading Style Sheets* é possível criar animações complexas, criar efeitos com uso de *parallax*, que faz parecer que a imagem de fundo tem uma profundidade diferente um dos outros, criar sites interativos e também jogos com HTML5 e CSS3.

O CSS usa uma sintaxe simples baseada em inglês com um conjunto de regras que o governam. Como mencionamos anteriormente, o HTML nunca teve a intenção de usar elementos de estilo, apenas a marcação da página.

Foi criado para descrever apenas o conteúdo. Por exemplo: `<p>Este é um parágrafo.</p>`.

Mas como você estiliza o parágrafo? A estrutura da sintaxe CSS é bem simples. Tem um seletor e um bloco de declaração.

Você seleciona um elemento e depois declara o que deseja fazer com ele. Bastante simples, certo?

Mas tem algumas regras que você precisa saber. Isso também é simples, não se preocupe.

O seletor aponta para o elemento HTML que você deseja estilizar. O bloco de declaração contém uma ou mais declarações separadas por ponto e vírgula.

Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos.

Uma declaração CSS sempre termina com um ponto-e-vírgula e os blocos de declaração são cercados por chaves.

Vamos ver um exemplo:

Todos os elementos **<p>** serão estilizados e serão coloridos de azul e negrito.

```
1 <style>
2
3   p {
4     color: blue;
5     font-weight: bold;
6   }
7
8 </style>
```

Em outro exemplo, todos os elementos **<p>** serão centralizados, com tamanho 16x e de cor pink.

```
1 <style>
2
3   p {
4     text-align: center;
5     font-size: 16px;
6     color: pink;
7   }
8
9 </style>
```

Estilos CSS Interno, Externo e Inline

Analisaremos cada estilo de forma rápida! Para uma explicação detalhada, acesse o link que vamos disponibilizar logo abaixo.

Vamos começar com o estilo **interno**. Estilos CSS feitos desta forma são carregados cada vez que um site é atualizado, o que pode aumentar o tempo de carregamento.

Além disso, você não poderá usar o mesmo estilo CSS em várias páginas, pois está contido em uma única página.

Mas a vantagem disso é que ter tudo em uma página facilita o compartilhamento do modelo para uma visualização.

O método **externo** pode ser o mais conveniente.

Tudo é feito externamente em um arquivo **.css**. Isso significa que você pode fazer todo o estilo em um arquivo separado e aplicar o CSS a qualquer página desejada.

O estilo externo também pode melhorar o tempo de carregamento.

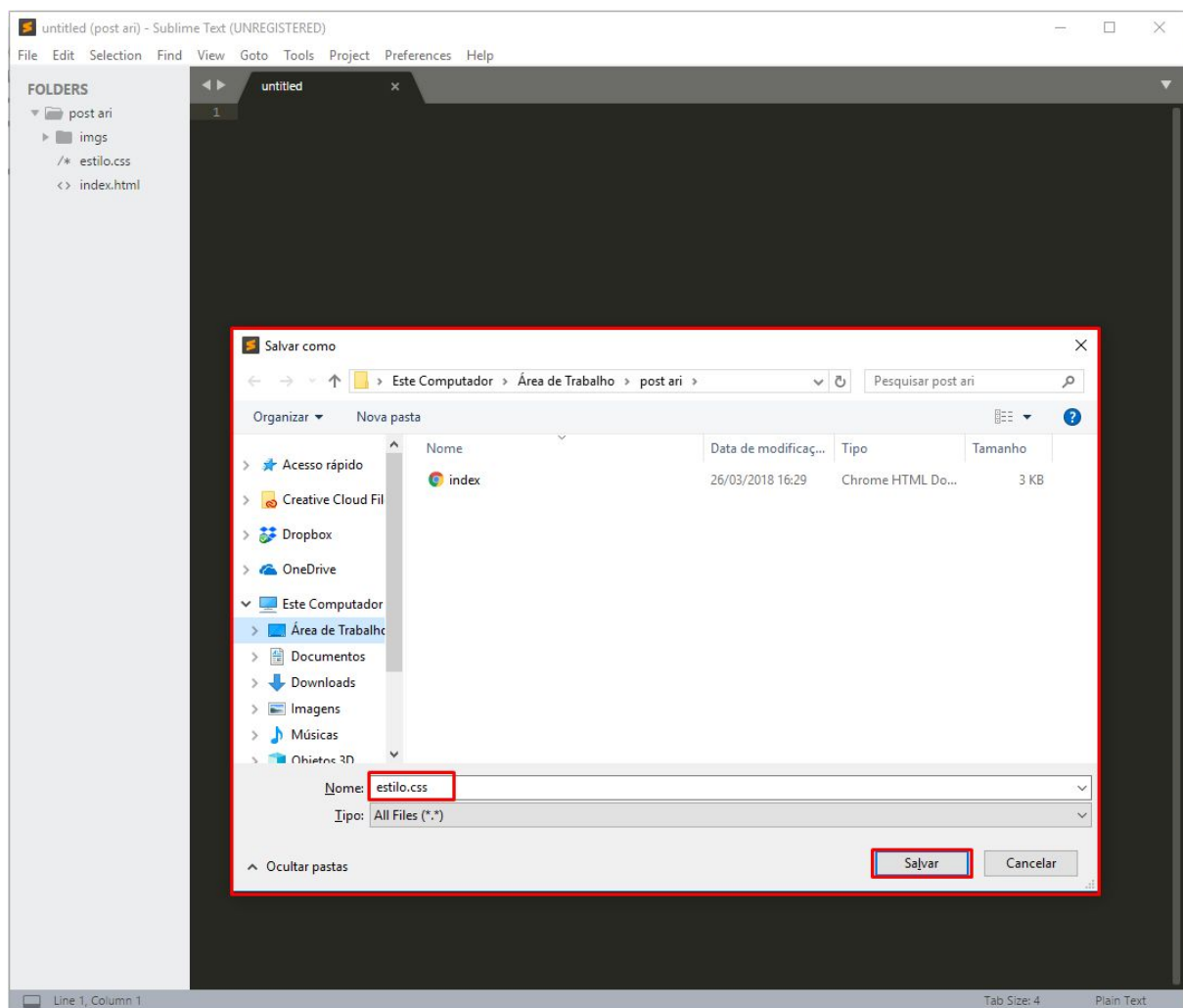
Por fim, tem também o estilo **Inline** do CSS. Inline trabalha com elementos específicos que possuem a tag `<style>`.

Cada componente deve ser estilizado, por isso talvez não seja o melhor ou o mais rápido para lidar com CSS.

Como Criar uma Folha de Estilo

Para criar uma folha de estilos é bem simples. Basta abrir seu editor de códigos, como o Sublime, e criar um documento novo.

Depois de criado, salve como `estilo.css`, mesmo sem ainda ter escrito nada (este `.css` é o que define um arquivo ser reconhecido pelo navegador como a folha de estilo).



Assim que for salvo, o editor de texto vai conseguir ler o arquivo com mais facilidade, até mesmo com sugestões de *auto complete* enquanto digita os códigos da folha de estilo, como mostra o print abaixo:

Como integrar o CSS no HTML?

Após salvar como estilo.css precisamos colocar o arquivo em nosso arquivo HTML, porque é ele quem vai carregar o CSS, o CSS sozinho não faz site! E é com *tags* de referência que você avisa ao HTML que existe um arquivo de estilo para ele, veja:

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

Esta *tag* vai indicar ao HTML que tem um arquivo CSS para ser adicionado à página.

Isto deve ser adicionado dentro da *tag* <head> </head> do seu HTML, como pode ver nesse print abaixo.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title></title>
5      <link rel="stylesheet" type="text/css" href="estilo.css">
6  </head>
7  <body>
8  </body>
9  </html>
```

Agora abra o documento estilo .css com o Sublime, que provavelmente está na mesma pasta do nosso HTML.

```
1  /*lembra da tag body do seu html?
2  então, o css consegue colocar estilo em tags html*/
3  body{
4
5  }
6  /*esse parâmetro de classe começa com "." e pode ser adicionado ao html
7  com o atributo class="" do html mais de uma vez e mais de uma classe por tag html*/
8  .classe{
9
10 }
11 /*já o parâmetro ID começa com # + o nome que quiser. Só é possível usar um ID por tag html
12 ele é adicionado através do atributo id="" do html*/
13 #id{
14
15 }
```

A partir desta última imagem, vamos entender como o CSS funciona com três parâmetros básicos:

Elementos Básicos do CSS

Os elementos são tags do HTML, #id e .classe.

```
1  body
2  h1
3  h2
4  h3
5  h4
6  h5
7  h6
8  span
9  p
```

2. Classes CSS

Criar e usar uma classe CSS é simples.

No documento CSS basta colocar .nome-da-classe-que-deseja . Abaixo criamos a classe título:

```
.titulo{
  font-size: 3rem; /*tamanho da fonte do título*/
  color: #333; /*esta é a cor do texto*/
  text-align: center; /*o alinhamento do texto que carregar esta classe*/
}
```

Note que existe um ponto (.) antes do nome da classe. Isso faz com que o CSS entenda que se está criando uma classe para ser usada no HTML.

Para se adicionar essa classe que criamos dentro do CSS, basta que na *tag* HTML tenha o parâmetro class="".

Observação: dentro das aspas, não precisa usar o ponto (.) antes da classe pois o nome *class* no HTML já avisa para o CSS que é para ele buscar pelo nome da classe com “.”

```
<h1 class="titulo">Melhor site</h1>
```

Lembrando que uma *tag* HTML, como h1 ou qualquer outra, pode ter mais de uma classe, basta separá-las com espaço como mostra no exemplo abaixo:

```
<h1 class="titulo fonte-grande sublinhado">Melhor site</h1>
```

Nesta tag foram usadas três classes: “titulo”, “fonte-grande” e “sublinhado”.

Mas para que todas estas classes tenham efeito é preciso criá-las no CSS.

Veja como criamos estas classes:

```
.titulo{
  font-size: 3rem; /*tamanho da fonte do título*/
  color: #333; /*esta é a cor do texto*/
  text-align: center; /*o alinhamento do texto que carregar esta classe*/
}
.fonte-grande{
  font-size: 40px; /*tamanho da fonte do seu título ou texto em pixels*/
}
.sublinhado{
  text-decoration: underline; /*ess propriedade faz com que seu texto fique sublinhado*/
}
```

3- Criando ID's CSS:

O uso dos ID's em CSS é parecido com as classes, mas as principais diferenças são que os id's são identificados com # e só podem ser usados um id por tag html.

Para adicionar um ID ao HTML, adicione como parâmetro id="".

Veja que a quando usamos no HTML, também excluimos o # do id da mesma forma que excluimos o . da classe.

```
.titulo{
  font-size: 3rem; /*tamanho da fonte do título*/
  color: #333; /*esta é a cor do texto*/
  text-align: center; /*o alinhamento do texto que carregar esta classe*/
}
#botao-bonito{
  color:white; /*cor da fonte*/
  padding-top: 15px; /*espaço superior entre o texto e o elemento*/
  padding-bottom: 15px; /*espaço inferior entre o texto e o elemento*/
  padding-right: 35px; /*espaço a direita entre o texto e o elemento*/
  padding-left: 35px; /*espaço a esquerda entre o texto e o elemento*/
  background-color:green; /*cor verde para o fundo do elemento, nesse caso o botão*/
}

<h1 class="titulo fonte-grande sublinhado">Melhor site</h1>
<a id="botao-bonito" href="link-para-alguma-pagina">clique nesse botão</a>
```

Dessa forma, o botão deverá ficar assim:



DICA: Este parâmetro de ID's não é o mais apropriado para se usar em uma página para muitos botões pois pode causar conflitos com a linguagem de programação JavaScript, que utiliza normalmente dos ID's no HTML para realizar ações mais complexas.

referência para aprofundar os conhecimentos:

<http://tableless.github.io/iniciantes/manual/css/>

fonte:

<https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css/>