

CSS: Sintaxe, Formas de Utilização e Seletores

NESTA AULA SERÁ APRESENTADA A SINTAXE PARA USO DE CSS (FOLHAS DE ESTILO EM CASCATA) , AS FORMAS QUE ESSA LINGUAGEM DE ESTILO PODEM SER APLICADAS À UMA PÁGINA E O QUE SÃO SELETORES E COMO UTILIZÁ-LOS.

AUTOR(A): PROF. ADILSON ROGERIO DE ALMEIDA

AUTOR(A): PROF. ADILSON ROGERIO DE ALMEIDA

O que é CSS?

CSS (Folhas de estilo em Cascata, do inglês, Cascading Style Sheets) é uma linguagem de estilo utilizada em conjunto com o HTML, que é uma linguagem de marcação. (MOZILA DEVELOPER, ONLINE).

Sintaxe do CSS

A sintaxe para o CSS possui duas partes principais: um seletor e uma ou mais declarações, conforme exemplificado a seguir:(MAUJOR, ONLINE).

RESUMINDO:

- O que é um seletor?

O seletor é utilizado para indicar em qual tag HTML é aplicada uma determinada propriedade de formatação.(MAUJOR, ONLINE).

- O que é uma propriedade?

A propriedade pode ser, por exemplo, a cor de fundo de uma página, a cor do texto, a fonte do texto, alinhamento do texto, tipo de borda, dentre outras formatações possíveis, discutidas adiante. (MAUJOR, ONLINE).

- O que é um valor?

O valor da propriedade é justamente a característica, cor, tamanho, etc, que você define!(MAUJOR, ONLINE).

Formas de utilização do CSS

Existem três formas que podemos utilizar as CSS: Método Inline, Método Interno e Método Externo. (MOZILLA DEVELOPER, ONLINE).

Qual a diferença entre esses métodos? Como você pode aplicar CSS em uma página HTML? Aprenderemos isso agora!!!

Método Inline (utilizando o atributo style na linha de comando html)

Neste método colocamos o atributo style na tag que desejamos inserir um estilo, e a sintaxe para esse método ficaria da seguinte maneira:

```
1. <body style="background-color: red;">
```

A propriedade “background-color” define a cor do fundo da página; no caso acima definimos que a cor será o vermelho.

É possível definir quantas propriedades quiser com o atributo style, em qual tag desejar, basta separar cada uma das propriedades com “;” (ponto e vírgula). O ponto e vírgula é um aviso de que o comando foi terminado.

A seguir temos um exemplo de como apresentar texto de uma tag h1 na cor verde e além disso inserir um espaçamento da margem esquerda 10px (10 pixels):

```
1. <h1 style="color:green;margin-left:10px;">Este é um cabeçalho verde</h1>
```

O código HTML completo, contendo os dois exemplos dados, ficaria assim:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3.   <head>
4.     <title>Método CSS Inline</title>
5.   </head>
6.
7.   <body style="background-color: red;">
8.
9.     <h1 style="color:green;margin-left:10px;">Este é um cabeçalho verde
10.    <p>Este é um parágrafo sem formatação alguma.</p>
11.
12.   </body>
13. </html>
```

DICA:

Dica: tente fazer um exercício de fixação utilizando seu editor Brackets seguindo os seguintes passos:

1. Crie um novo arquivo no editor;
2. Copie e cole o código acima;
3. Salve como html;
4. Visualize no navegador;
5. Altere os valores para ver como ficaria com outras formatações.

VAMOS TREINAR?

Acesse o link a seguir e edite o código preexistente: https://www.w3schools.com/css/tryit.asp?filename=trycss_howto_inline
(https://www.w3schools.com/css/tryit.asp?filename=trycss_howto_inline)

Método Interno (utilizando a tag style)

Neste método utilizamos diretamente a tag `style`, que deve ser colocada entre a abertura e o fechamento da tag `head`. Nesse caso, entre a abertura e o fechamento da tag `style` você deve colocar as tags que deseja formatar com as propriedades e valores correspondentes. Fica mais simples de entender com o exemplo a seguir, onde utilizamos a tag `style` para formatar a cor do texto (azul) e inserir uma margem de 30 pixels à esquerda em um cabeçalho `h1`, além de definir uma cor para o fundo da página (amarelo).

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3. <head>
4.     <title>Método CSS Interno</title>
5.     <style>
6.         body {
7.             background-color: yellow;
8.         }
9.         h1 {
10.            color: blue;
11.            margin-left: 30px;
12.        }
13.    </style>
14. </head>
15. <body>
16.     <h1>Este é um cabeçalho de nível h1 de cor azul e com margem de 30 pixels</h1>
17.     <p>Este é um parágrafo para o qual não foi definida formatação.</p>
18.
19. </body>
20. </html>
```

O resultado no navegador é:



CONHECENDO OS EDITORES ONLINE

Além de utilizar o editor Brackets instalado em sua máquina para fazer exercícios e atividades, você pode também treinar utilizando um editor on-line.

Veja abaixo o endereço e assista o vídeo que explica como utilizá-los!

1. Editor online disponível no <https://codepen.io/pen/> (<https://codepen.io/pen/>).



Método Externo (link para a folha de estilos)

O uso de folhas de estilos externas é interessante porque você pode alterar o estilo de um site inteiro alterando apenas um arquivo externo. Esse método é o mais indicado, pois esse arquivo externo com as instruções de estilo pode ser linkado em várias páginas de um site, tendo um maior aproveitamento do código criado.

Existem duas coisas que você precisa fazer para utilizar esse método:

- Criar um arquivo de extensão “.css” com as formatações que desejar;
- Referenciar esse arquivo no código HTML da página que deseja implantar as instruções.

Estes passos estão detalhados a seguir:

Criando um arquivo de extensão “.css”

Para criar um arquivo com as instruções de estilo siga os seguintes passos:

1. Abra o seu editor HTML (Brackets, por exemplo);
2. Gere um novo arquivo em branco;
3. Nesse arquivo você deve colocar o nome da tag que deseja formatar, abrir o símbolo de chaves “{”, inserir as propriedades e valores desejados para aquela tag e posteriormente fechar o símbolo de chaves “}”, com a mesma sintaxe e de forma semelhante ao que fizemos no método interno, certo?.

Segue um exemplo para melhor visualização do que deve ser inserido neste documento, nos baseando no exemplo dado no método interno:

VAMOS TREINAR?

Teste o código acima no seu editor HTML Brackets e acesse o link a seguir e edite o código preexistente: https://www.w3schools.com/css/tryit.asp?filename=trycss_howto_internal
(https://www.w3schools.com/css/tryit.asp?filename=trycss_howto_internal)

```
1. body{
2. background-color: yellow;
3. }
4.
5. h1 {
6. color: blue;
7. margin-left: 30px;
8. }
```

O código acima ficaria da seguinte maneira no editor HTML Brackets:



```
1 body {  
2     background-color: yellow;  
3 }  
4  
5 h1 {  
6     color: blue;  
7     margin-left: 30px;  
8 }  
9
```

The screenshot shows a code editor with a dark theme. The left sidebar has a 'Working Files' section with 'exemplo id.html' and 'minha_folha_de' listed. Below it is a 'Getting Started' section with 'screenshots', 'index.html', and 'main.css'. The main editor area displays the CSS code shown in the code block. The status bar at the bottom indicates 'Line 9, Column 1 — 9 Lines', 'INS', 'CSS', and 'Spaces: 4'.

ATENÇÃO

Note que neste arquivo em criação você não deve colocar nenhuma tag!!! Em um arquivo css não há cabeça e nem corpo (head ou body). Há apenas a inserção do código css diretamente no arquivo, como no exemplo acima.

Agora temos que salvar esse documento com a extensão “.css”, para isto crie um arquivo com o código acima e salve com o nome “minha_folha_de_estilos.css”.

Mas como utilizar essa folha de estilos no meu documento HTML? Descobriremos a seguir! :D



Utilizando a folha de estilos externa em uma página

Para utilizar a folha de estilos criada, você deve utilizar a tag link entre a abertura e o fechamento da tag head.

Suponha que você salve o arquivo CSS com o nome “minha_folha_de_estilos.css” e o mesmo seja salvo na mesma pasta que o documento HTML que deseja aplicá-la.

Teríamos que adicionar a tag "link" no documento HTML da seguinte maneira:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3. <head>
4.     <title> CSS externo </title>
5.     <!-- abaixo segue a linha de código necessária para avisar ao navegador
6.     <link rel="stylesheet" type="text/css" href="minha_folha_de_estilos.css">
7. </head>
8. <body>
9.
10.    <h1>Este é um cabeçalho de nível h1 de cor azul e com margem de 30px</h1>
11.    <p>Este é um parágrafo para o qual não foi definida formatação.</p>
12.
13. </body>
14. </html>
```

Nesta linha de comando as tags e atributos representam:

1. link: ligação, avisa ao navegador que a página html mantém uma ligação externa.
2. rel: abreviação de relationship ou relação/relacionamento.
3. type: tipo - avisa ao navegador o tipo de ligação de relacionamento existente, neste caso, uma ligação do tipo texto/css.
4. href: abreviação de hypertext reference ou referência de hipertext - ou seja, avisa ao navegador o local em que o arquivo CSS que mantém a ligação com o arquivo html, ou seja, o caminho da folha de estilos a ser utilizada.


```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3. <head>
4.     <title> Criando um link para outra pasta </title>
5.     <link rel="stylesheet" type="text/css" href="style/minha_folha_de_
6. </head>
7. <body>
8.
9.     <h1>Este é um cabeçalho de nível h1 de cor azul e com margem de 30px</h1>
10.    <p>Este é um parágrafo para o qual não foi definida formatação.</p>
11.
12. </body>
13. </html>
```

LIGANDO O ARQUIVO EXTERNO CSS AO ARQUIVO HTML

Para ligar os dois arquivos, salve esse arquivo com a extensão “.html” na mesma pasta que você salvou o código em CSS. Abra o arquivo HTML em seu navegador e veja o resultado!



Indicando o caminho de um arquivo

Imagine agora que a folha de estilos não seja salva na mesma pasta que o documento HTML, mas sim em uma subpasta no mesmo diretório, que você pode nomear de “style”, por exemplo, temos de indicar isso no valor dado a “href”. Ficaria assim:

ATENÇÃO

Se você salvar um arquivo dentro de uma pasta que esteja na mesma raiz (pasta) do arquivo html, para que o navegador saiba qual é o caminho correto você deve informar qual é o nome da pasta, seguido do nome do arquivo que deve ser aberto. Neste exemplo acima temos que a pasta se chama “style” e que o arquivo que deve ser acessado é o `minha_folha_de_estilos.css`, por isto ficou `href="style/minha_folha_de_estilos.css"`, ok?

Caso você precise avisar que ele deve ir “entrando” em novas pastas então basta colocar o nome destas pastas no caminho criado, como por exemplo:

```
1. <link rel="stylesheet" type="text/css" href="style/outra_pasta/outra_pasta"
```

Caso você precise avisar ao navegador que ele deve "sair" de uma pasta, então a forma correta é apenas colocar dois pontos (..) e uma barra (/) antes do nome da pasta ou arquivo.

Neste exemplo acima se tivéssemos que pedir para o navegador:

a) sair da pasta e retroceder um nível então faríamos assim:

```
href="../style/minha_folha_de_estilos.css"
```

A linha de comando ficaria assim:

```
1. <link rel="stylesheet" type="text/css" href="../style/minha_folha_de_estilos.css">
```

b) sair da pasta e retroceder dois níveis então faríamos assim:

```
href="../../style/minha_folha_de_estilos.css"
```

A linha de comando ficaria assim:

```
1. <link rel="stylesheet" type="text/css" href="../../style/minha_folha_de_estilos.css">
```

b) sair da pasta e retroceder três níveis então faríamos assim:

```
href="../../../style/minha_folha_de_estilos.css"
```

A linha de comando ficaria assim:

```
1. <link rel="stylesheet" type="text/css" href="../../../style/minha_folha_de_estilos.css">
```

E assim por diante...

Identidades e Classes em CSS

Para que a página seja apresentada uma forma mais organizada e a seleção dos elementos por CSS ocorram da melhor forma, podemos identificar as tags por classes e/ou identidades. (GUILHERME MULLER, ONLINE).

Criando uma identidade (id)

Uma id é uma identificação única e uma determinada id deve ser utilizada apenas uma vez no documento inteiro para um determinado elemento HTML.

Geralmente, as Ids são utilizadas na identificação de elementos estruturais da página. (W3SCHOOLS, ONLINE).

A sintaxe de um id é definida da seguinte maneira:

```
# nome_identidade{propriedade:valor;}
```

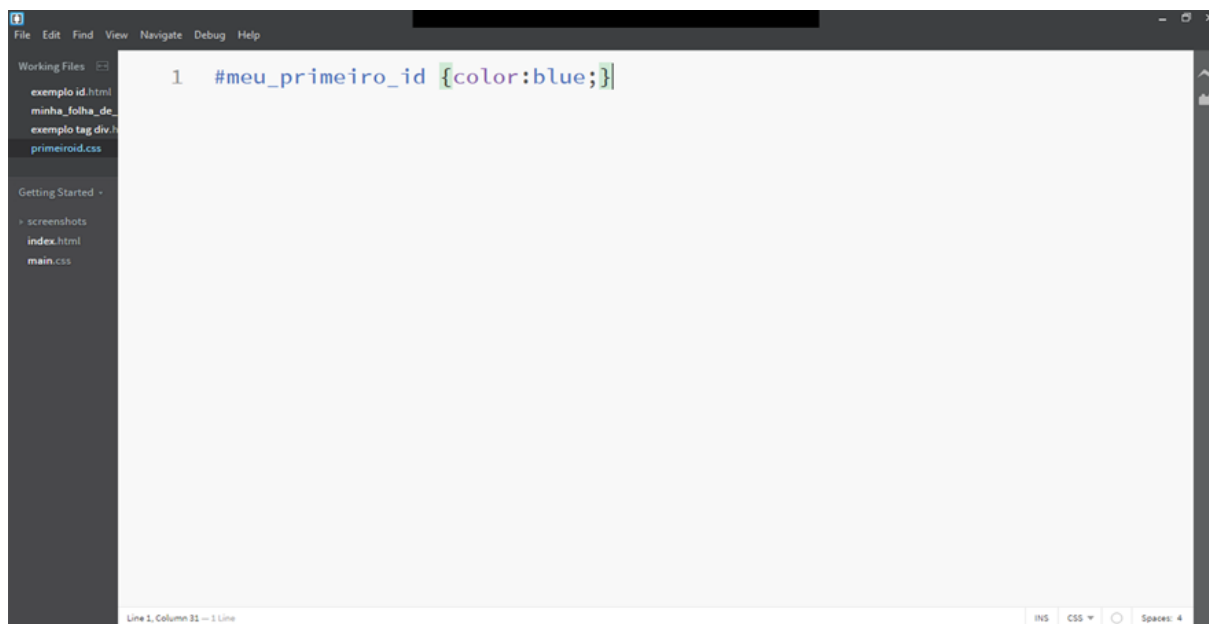
Você pode escolher o nome do Id seguindo algumas regras básicas, sendo:

- não deve conter acentos e nem caracteres especiais tais como ç;
- não pode conter espaços;
- é case sensitive (ou seja, se for criada com letras maiúsculas e minúsculas, deve ser chamada com o mesmo nome de sua criação)

Abaixo temos um exemplo de uma Id criada em um documento CSS:

```
#meu_primeiro_id {color:blue;}
```

No editor HTML em um arquivo css externo ficaria assim:



Para atribuir o valor desse id em algum elemento do seu documento HTML, siga o procedimento padrão de referenciamento por CSS externo em seu código e indique o nome da id presente no documento CSS.

Salve, por exemplo o código CSS acima com o nome “primeiroid.css” em uma pasta em sua máquina.

A seguir crie um documento HTML com o código abaixo:

```
1. <!DOCTYPE html>
2. <html lang="pt-br">
3. <head>
4.     <title> Criando identidades em CSS </title>
5.     <link rel="stylesheet" type="text/css" href="primeiroid.css" />
6. </head>
7. <body id="meu_primeiro_id">
8.
9.     <p>Parágrafo azul</p>
10.
11.    <h2>Cabeçalho de nível 2 azul</h2>
12.
13.    <p>Conteúdo de um parágrafo azul</p>
14.
15. </body>
16. </html>
```

COMO CHAMAR O ID NO HTML

Observe no código acima que para chamar o id criado no css no arquivo html utilizamos o id=nome_do_id.

Sendo assim temos que:

- Para criar o id no CSS utilizamos # e fica assim:
#meu_primeiro_id {color:blue;}
- Para chamar o id no html utilizamos o id=nome_id e fica assim:
id="meu_primeiro_id"

Sendo assim a linha de comando de chamada do id ficaria:

```
1. <body id="meu_primeiro_id">
```

VAMOS TREINAR?

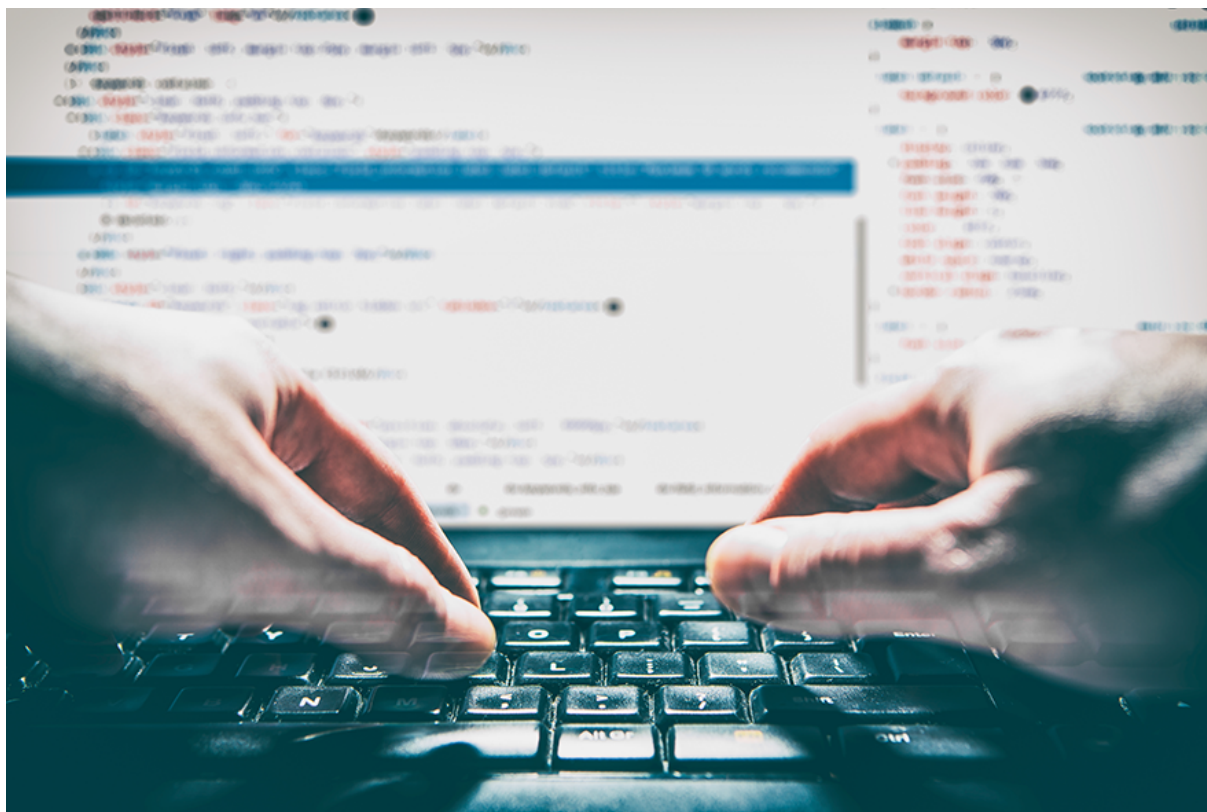
Para treinar a utilização do código acima você tem as seguintes opções:

1. Utilizar o editor Brackets instalado em sua máquina.
2. Utilizar um editor online disponível em: <https://codepen.io/pen/> (<https://codepen.io/pen/>).

Para fixar bem este conceito também você também pode

1. Editar diretamente um código pronto Clicando aqui (https://www.w3schools.com/css/tryit.asp?filename=trycss_syntax_id) e utilizando o editor online do W3Schools.

Escolha a melhor opção e treine bastante!!



Criando uma classe

A diferença entre a identidade e a classe é que as classes são reutilizáveis!!!

Você pode criar diferentes estilos para parágrafos em um arquivo CSS, por exemplo, e utilizá-los nos parágrafos que quiser no documento HTML usando o atributo “class” junto ao elemento a se aplicar o estilo. (GUILHERME MULLER, ONLINE).

A sintaxe para o seletor classe é definida da seguinte maneira:

```
.nome_da_classe: { propriedade: valor;}
```

Podem ser colocadas diversas propriedades e seus respectivos valores para uma mesma classe.

E ainda mais! Podem ser declaradas mais de uma classe para um mesmo elemento HTML! Entenderemos melhor com exemplos.

Imagine que você precisa utilizar dois estilos de parágrafos na sua página, um estilo de parágrafo com o texto na cor azul e um outro estilo de parágrafo na cor vermelha.

O primeiro passo é criar essas duas classes de parágrafos no documento CSS que será aplicado à página.

Teríamos o seguinte código CSS:

```
1. .cor1 {  
2.   color: blue;  
3. }  
4.  
5. .cor2 {  
6.   color: red;  
7. }
```

O arquivo CSS pode ser salvo com o nome “testeclass1.css”, por exemplo.

Criaremos agora um documento HTML aplicando essas classes como atributos aos parágrafos.

```
1. <!DOCTYPE html>  
2. <html lang="pt-br">  
3.   <head>  
4.     <link rel="stylesheet" type="text/css" href="testeclass1.css" />  
5.   </head>  
6. <body>  
7.  
8.   <p class="cor1">Parágrafo azul</p>  
9.   <p class="cor2">Parágrafo vermelho</p>  
10.  
11. </body>  
12. </html>
```

Viu como é simples utilizar classes e id's?

Abaixo seguem algumas informações sobre como combinar seletores que podem ser úteis para entender códigos em CSS mais complexos!

SUGESTÃO DE EDITORES HTML ON-LINE.

COMBINANDO SELETORES

Podemos também combinar os seletores que aprendemos acima, para conseguir diferentes elementos e partes mais específicas de nossos sites. Alguns exemplos:

- `p.destaque` seleciona apenas os parágrafos que possuem a classe "destaque";
- `div#cabecalho h1` seleciona tags `h1` que estejam dentro da `div` com a `id` "cabecalho";
- `#conteudo ul li a` seleciona links (tag `a`) dentro de itens de lista dentro de tags `ul` que estejam dentro de um elemento com a `id` "conteudo";
- `#conteudo p.destaque strong` seleciona elementos `strong` dentro de parágrafos com a classe "destaque" que estejam dentro de um elemento com a `id` "conteudo";
- `.mensagem.destaque` seleciona apenas elementos que tenham a classe "mensagem" e a classe "destaque";
- Separando itens por vírgulas, como `p.destaque, h1, a.saiba-mais` seleciona todos os respectivos elementos para as regras. Muito útil para diminuir a repetição de regras no arquivo CSS.

[HTTP://GUILHERMEMULLER.COM.BR/PT/ELEARNING/HTML_CSS_BASICO/LICAO/2/SINTAXE-SELETORES-CSS](http://GUILHERMEMULLER.COM.BR/PT/ELEARNING/HTML_CSS_BASICO/LICAO/2/SINTAXE-SELETORES-CSS)

Para treinar a utilização do código acima você tem as seguintes opções:

1. Utilizar o editor Brackets instalado em sua máquina.
2. Utilizar um editor online disponível em <https://codepen.io/pen/> (<https://codepen.io/pen/>).

Para fixar bem este conceito também você também pode

1. Editar diretamente um código pronto Clicando aqui (https://www.w3schools.com/css/tryit.asp?filename=trycss_syntax_class) e utilizando o editor online do W3Schools.

Escolha a melhor opção e treine bastante!!

Regras de Precedência

Quando um mesmo elemento HTML recebe duas regras de estilo, o navegador decide qual delas será aplicada ao elemento. (UFPA, ONLINE).

Imagine que em uma página é aplicada uma folha de estilos externa e além disso possui instruções de CSS interno. Qual dos dois estilos o navegador apresentará? (GUILHERME MULLER, ONLINE).

Para esse caso, a resposta seria: O navegador apresentará o estilo definido pelas regras do CSS interno. Mas porquê? (GUILHERME MULLER, ONLINE).

Tenha sempre em mente que as regras de precedência ocorrem sempre do mais específico para o mais genérico! (GUILHERME MULLER, ONLINE).

Tendo isso em mente, podemos definir a sequência de precedência da seguinte maneira (do mais específico ao mais geral): (UFPA, ONLINE).

1. Estilos definidos pelo método Inline;
2. Estilos definidos pelo método Interno;
3. Seletores de elementos em arquivos externos CSS.

IMPORTANTE!

O navegador sempre obedecerá ao CSS que está mais perto do elemento que tem de ser formatado, ok?

Frameworks

Frameworks são conjuntos de códigos genéricos prontos que você pode utilizar no seu projeto. Utilizando esses Frameworks, você otimiza seu tempo! (PROFISSIONAIS TI, ONLINE).

Seguem abaixo alguns Frameworks CSS para auxiliar no desenvolvimento! Acesse e confira!

- Blueprint CSS (<http://www.blueprintcss.org/> (<http://www.blueprintcss.org/>)).
- 960 Grid System (<https://960.gs/> (<https://960.gs/>)).
- SenCSS (<https://sencss.kilianvalkhof.com/> (<https://sencss.kilianvalkhof.com/>)).

Assista ao vídeo abaixo com um resumo desta aula.



O QUE VOCÊ PRECISA FAZER NESTE TÓPICO

- Utilizar o editor Brackets para fazer os exercícios propostos;
- Aprender a utilizar os editores on-line;
- Assistir aos vídeos;
- Fazer as atividades de treino.

O QUE VOCÊ PRECISA TER ENTENDIDO NESTE TÓPICO

- Qual a sintaxe de um comando em CSS;
- O que é CSS inline;
- O que é CSS Interno;
- O que é CSS Externo.

ATIVIDADE FINAL

Quais são os três métodos de aplicação do CSS?

- A. método inline, método identificador e método de classes
- B. método inline, método identificador e método externo
- C. método overline, método interno e método externo
- D. método inline, método interno e método externo

O atributo `style` é utilizada em qual método de aplicação de CSS?

- A. Método depurado
- B. Método interno
- C. Método inline
- D. Método externo

Qual a sintaxe correta das regras criadas em um documento de extensão CSS?

- A. `seletor {propriedade1:valor; propriedade2:valor;}`
- B. `seletor {propriedade1:seletor; propriedade2:valor;}`
- C. `seletor {propriedade1=valor; propriedade2=valor;}`
- D. `seletor {propriedade1:valor, propriedade2:valor,}`

REFERÊNCIA

CRIARWEB. Disponível em: <<http://www.criarweb.com/artigos/introducao-a-css3.html>>. Acesso em 05 mai. 2017.

GUILHERME MULLER. Disponível em: <http://guilhermemuller.com.br/pt/elearning/html_css_basico/licao/2/sintaxe-seletores-css>. Acesso em 07 mai.2017.

MAUJOR. Disponível em: <<http://www.maujor.com/tutorial/sintaxetut.php>>. Acesso em 06 mai.2017.

MOZILLA DEVELOPER. Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/How_CSS_works>. Acesso em 07 mai.2017.

W3SCHOOLS. Disponível em: <https://www.w3schools.com/css/css_howto.asp>. Acesso em 05 mai.2017.

UFPA. Disponível em: <<http://www.ufpa.br/dicas/htm/htm-esti.htm>>. Acesso em 07 mai. 2017.

