

CADERNO HTML & CSS

HTML



CSS



Sumário

Sumário	2
Aula 01	5
HTML	5
TAGs	5
ESTRUTURA BÁSICA PÁGINA HTML	5
Aula 02	6
COR DE FUNDO	6
IMAGEM DE FUNDO	6
Adicionar ícones na página	8
Cabeçalhos	8
PARÁGRAFO	9
Quebras de linha	10
ESTILOS DE TEXTO	10
Formatação ()	11
Tipo da fonte:	11
Tamanho da fonte:	12
Combinação de comandos:	12
Aula 03	13
Lista não ordenada ()	13
Atributos	13
type :	13
Lista ordenada ()	14
Atributos	14
start	14
type	14
Lista Aninhadas	15
 e Aninhados	15
Aula 04	16
CSS	16
Introdução	16
Como CSS Funciona	17
Anatomia de um comando CSS	18
Estilos CSS Interno, Externo e Inline (local)	18

CSS externo	19
CSS Interno	20
CSS Inline (Local)	20
Utilizando classes:	21
Aula 05	21
DIV <div> :	21
Linha de separação <hr> :	22
Atributos	22
align:	22
color:	22
size:	22
width:	22
Aula 06	23
Hyperlinks <a> :	23
Atributo:	23
href:	23
Links internos:	23
Links externos	24
Links verticais	24
Aula 07	25
Imagens :	25
Atributos:	25
alt:	25
height:	25
width:	26
src:	26
Propriedades de css:	26
Imagens arredondadas:	26
Imagem transparente:	26
Filtros de imagens:	27
Aula 08	28
Tabela <table> :	28
O que são tabelas?	28
A estrutura básica de uma tabela HTML	28
Célula de título da tabela HTML	28
Tags estruturais da tabela	28
Células que abrangem mais que uma linha/coluna	29

Aula 01

HTML

A sigla HTML é uma abreviação de HyperText Markup Language é um linguagem de marcação usada para criar páginas web, que por sua vez será composta de textos e comandos especiais os quais são chamados de Tags.

Para podermos começar a criação de páginas HTML precisamos utilizar um editor de textos simples como bloco de notas, gedit, entre outros ou editores de texto especiais como Visual Studio Code.

TAGs

Os comandos em HTML são chamados de tags, e elas são responsáveis que irão dizer ao navegador como serão exibidos as informações.

As tags são identificadas por serem envolvidos pelos sinais < > e </ >. Entre os sinais < > são especificados as tags proprietárias. No caso de tags que precisam envolver um texto, a sua finalização deverá ser feita usando </ > , indicando que a tag está finalizado a marcação de um texto, o que faz as tags normalmente andam em pares. o Formato genérico de um tag é :

<nome da tag> Texto</nome da tag>

ESTRUTURA BÁSICA PÁGINA HTML

Assim como em outras linguagens, existem estruturas básicas que identificam partes de um programa HTML.

```
1  <!DOCTYPE html> <!-- define o tipo de documento -->
2  <html lang="pt-br"> <!-- define o início do documento html, todos as demais tags devem estar dentro desta tag -->
3
4  <head> <!-- início do cabeçalho, onde ficar os metadados do documento, como metatags, css e javascript -->
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>Aula 1</title> <!-- define o título do documento, mostrado na barra de títulos do navegador-->
9  </head> <!-- Fim do cabeçalho -->
10
11 <body> <!-- Início corpo ou área principal do documento, dentro deste deve ficar o conteúdo da sua página -->
12     Olá Mundo
13 </body> <!-- Fim corpo ou área principal do documento -->
14
15 </html> <!-- Define o fim do documento html -->
```

No VSCode podemos digitar html:5 e pressionar enter para colocar a estrutura do html.

Aula 02

COR DE FUNDO

No HTML uma das formas de colocar uma cor de fundo na página é utilizando o comando bgcolor dentro da tag <body>. Para escolher a cor colocamos o nome da cor em inglês ou o código da cor, utilizando o código da cor podemos ter mais variedades de cores e

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>Aula 1</title>
9  </head>
10
11 <body bgcolor="#cecccc">
12
13 </body>
14
15 </html>
```

tonalidades.

No HTML 5 não é mais utilizado dessa maneira para colocar uma cor de fundo se utiliza o [CSS](#) para fazer.

Seletor de cores: <https://g.co/kgs/W7fS9f>

IMAGEM DE FUNDO

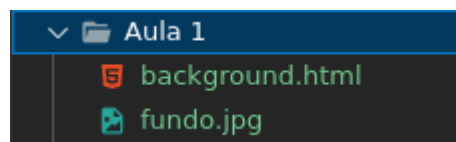
Para adicionar uma imagem de fundo, basta especificar na tag <body>, o atributo background contendo o endereço da imagem.

```

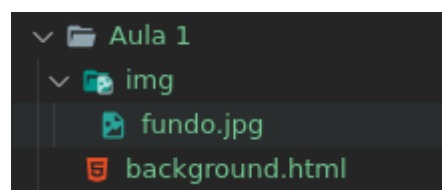
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>Background</title>
9  </head>
10
11 <body background="fundo.jpg">
12
13 </body>
14
15 </html>

```

Para que a imagem seja carregada na página HTML ela deve estar na mesma pasta que o arquivo.html e o nome deve ser igual ao da imagem.



Também podemos trabalhar com pastas criando uma pasta chamada img (imagens) e todas as imagens que forem utilizadas no html ficam dentro desta pasta. Para carregar uma imagem que está dentro desta pasta colocamos no background o nome da pasta/nome da imagem.



No HTML:

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Testando imagem de fundo</title>
9  </head>
10
11 <body background="img/fundo.jpg">
12
13 </body>
14

```

Adicionar ícones na página

Para poder adicionar ícones na página vamos utilizar o comando `<link>`, que permite carregar arquivos, porém precisamos informar para o html o que fazer com o mesmo, então precisamos adicionar o atributo `rel`, ex:

```
<link rel="shortcut icon" href="image.png">
```

- O atributo `href` carrega o arquivo desejado;
- O atributo `rel` informa o que fazer com ele, no caso adicionar como ícone na página.
- A tag `link` deve ser colocada no cabeçalho do html, entre a tag `<head>` e `</head>`.

O link pode ser utilizado para carregar outros tipos de arquivos.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Ícones na página</title>
9      <link rel="shortcut icon" href="img/icone.png" type="image/x-icon">
10 </head>
11
12 <body>
13
14 </body>
15
16 </html>
```

Cabeçalhos

No html podemos adicionar tags com formatações prontas de títulos, com os comandos `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` e `<h6>`. Quanto maior o número, menor será o tamanho do texto, pois o número se refere a prioridade de utilização.


```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>Cabçalhos</title>
9  </head>
10
11 <body>
12     <h1>Titulo 1</h1>
13     <h2>Titulo 2</h2>
14     <h3>Titulo 3</h3>
15     <h4>Titulo 4</h4>
16     <h5>Titulo 5</h5>
17     <h6>Titulo 6</h6>
18 </body>
19
20 </html>

```

No navegador:

Titulo 1

Titulo 2

Titulo 3

Titulo 4

Titulo 5

Titulo 6

PARÁGRAFO

No html podemos adicionar textos em forma de parágrafos, ou seja, com margens abaixo, ex:

```
<p>A prendendo HTML e utilizando tag de Formatação</p>  
<p>hoje esta um belo dia</p>
```

Quebras de linha

Você já deve ter percebido que não importa a quantidade de enters que você aplica a uma determinada linha de seu texto a quebra de linha nunca acontece. Isso ocorre porque esta função de quebra de linha é designada a tag `
`.

ESTILOS DE TEXTO

Aqui em baixo citarei várias outras tags para a formatação de textos em HTML, e em seguida seus respectivos exemplos de utilização.

Tag	Função	Exemplo
<code></code> , <code></code>	Coloca o texto em negrito.	<code>Texto em negrito</code> <code>Texto em negrito</code>
<code><i></code>	Coloca o texto em itálico.	<code><i>Este texto está em itálico</i></code>
<code><u></code>	Coloca um sublinhado no texto.	<code><u>Texto sublinhado</u></code>
<code><sub></code>	Coloca o texto como subscrito.	Texto <code><sub>subscrito</sub></code>
<code><sup></code>	Coloca o texto como sobrescrito.	Texto <code><sup>Sobrescrito</sup></code>
<code><strike></code>	Coloca o texto com taxado (Não utiliza mais o HTML 5)	<code><strike>Este texto está tachado</strike></code>

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Estilos de texto</title>
8  </head>
9
10 <body>
11     <strong>Negrito</strong>
12     <br>
13     <i>Italico</i>
14     <br>
15     <u>Sublinhado</u>
16     <br>
17     H<sup>2</sup></sup>0
18     <br>
19     <sub>3</sub>
20     <br>
21     <del>Tachado</del>
22 </body>
23
24 </html>
25

```

No navegador:

Negrito
Italico
Sublinhado
H²O
3
~~Tachado~~

Formatação ()

Em HTML é possível alterar o tipo da fonte, a sua cor e o seu tamanho, tudo com a tag .

Tipo da fonte:

Usa-se o comando face para designar o tipo da fonte dentro da tag .

Os valores a serem aplicados no comando face são: Arial, Verdana entre outros nomes de letras.

Exemplo:

```
<font face="Arial">Texto com fonte do tipo Arial.</font>
```

Cor da fonte:

Usa-se o comando color para designar a cor da fonte dentro da tag .

Seus possíveis valores são: nome de cores em inglês (blue, black, yellow, ...), cores representadas por códigos hexadecimais (#FF0000 (código da cor vermelha em hexadecimal)), entre outras formas. Mas não é preciso sair decorando números de cores, para isto existem tabelas e softwares que auxiliam no processo.

Exemplo:

```
<font color="red">Texto com fonte em cor vermelha.</font>
```

Seletor de cores: <https://g.co/kgs/W7fS9f>

Tamanho da fonte:

Usa-se o comando size para designar o tamanho da fonte dentro da tag .

Os possíveis valores são: vão de 1 a 8.

Exemplo:

```
<font size="8">Texto com fonte em tamanho 8.</font>
```

Combinação de comandos:

Também à possibilidade de se combinar comandos dentro de tags, sem a necessidade de uma ordem específica.

Exemplo:

```
<font color="blue" face="Verdana" size="8">Texto com fonte azul do tipo Verdana e  
tamanho de 8.</font>
```

Aula 03

Lista não ordenada ()

O elemento HTML (ou elemento HTML de Lista desordenada) representa uma lista de itens sem ordem rígida, isto é, uma coleção de itens que não trazem uma ordenação numérica e as suas posições, nessa lista, são irrelevantes. Caracteristicamente, os itens em uma lista desordenada são exibidos com um marcador que pode ter várias formas, como um ponto, um círculo, ou um quadrado. O tipo de marcador não é definido na descrição HTML da página, mas na CSS associada, utilizando a propriedade [list-style-type](#).

Atributos

type :

Usados para estabelecer o tipo de marcador da lista. Os valores definidos durante a HTML 3.2 e a versão de transição de HTML 4.0/4.01, são:

- circle
- disc
- square

A tag utilizamos para definir os itens da lista.

```
20  <ul>
21    <li>Item um</li>
22    <li>Item dois</li>
23    <li>Item tres</li>
24  </ul>
```

Lista ordenada ()

O Elemento HTML (ou Elemento HTML de lista ordenada) representa uma lista de itens ordenados. De forma característica esses itens ordenados em uma lista são mostrados com uma contagem que os precede, que pode ser de qualquer tipo, como numerais, letras, algarismos romanos, ou simples símbolos. Esse modelo numérico não é definido na descrição html da página, mas na folha de estilos CSS associada, pela propriedade [list-style-type](#).

Atributos

start

Este atributo inteiro especifica o valor inicial para a numeração dos itens da lista. Embora o tipo de ordenação dos elementos possa ser com algarismos romanos, tal como XXXI, ou letras, o valor inicial sempre é representado como um inteiro. Para iniciar a contagem a partir da letra "C", utilize <ol start="3">.

Nota: Este atributo, obsoleto na HTML4, foi reintroduzido na HTML5.

type

Indica o tipo de numeração:

- 'a' indica letras minúsculas,
- 'A' indica letras maiúsculas,
- 'i' indica algarismos romanos minúsculos,
- 'I' indica algarismos romanos maiúsculos,
- e '1' indica números (padrão).

O tipo de marcação é usado na lista inteira, a menos que um atributo type diferente seja utilizado dentro do elemento .

```
11  <body>
12    <h1>Lista Ordenada</h1>
13
14    <h3>Passos</h3>
15
16    <ol type="I" start="1">
17      <li>Passo 1</li>
18      <li>Psso 2</li>
19      <li>Psso 3</li>
20      <li>Psso 4</li>
21      <li>Psso 5</li>
22    </ol>
23  </body>
```

Lista Aninhadas

```
11 <body>
12   <ol>
13     <li>primeiro item</li>
14     <li>segundo item <!-- Veja que a tag de fechamento </li> não é colocada aqui! -->
15       <ol>
16         <li>segundo item primeiro subitem</li>
17         <li>segundo item segundo subitem</li>
18         <li>segundo item terceiro subitem</li>
19       </ol>
20     </li> <!-- Aqui está a tag de fechamento </li> -->
21     <li>terceiro item</li>
22   </ol>
23 </body>
```

No navegador:

1. primeiro item
2. segundo item
 1. segundo item primeiro subitem
 2. segundo item segundo subitem
 3. segundo item terceiro subitem
3. terceiro item

 e Aninhados

```
11 <body>
12   <ol>
13     <li>primeiro item</li>
14     <li>segundo item <!-- Observe que a tag de fechamento </li> não é colocada aqui! -->
15       <ul>
16         <li>segundo item primeiro subitem</li>
17         <li>segundo item segundo subitem</li>
18         <li>segundo item terceiro subitem</li>
19       </ul>
20     </li> <!-- Aqui está a tag de fechamento </li> -->
21     <li>terceiro item</li>
22   </ol>
23 </body>
```

No Navegador:

1. primeiro item
2. segundo item
 - segundo item primeiro subitem
 - segundo item segundo subitem
 - segundo item terceiro subitem
3. terceiro item

Aula 04

CSS

Introdução

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.

CSS é chamado de linguagem Cascading Style Sheet e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML. O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode criar tabelas, usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

CSS foi desenvolvido pelo W3C ([World Wide Web Consortium](https://www.w3.org/)) em 1996, por uma razão bem simples. O HTML não foi projetado para ter tags que ajudariam a formatar a página. Você deveria apenas escrever a marcação para o site.

Tags como foram introduzidas na versão 3.2 do HTML e causaram muitos problemas para os desenvolvedores. Como os sites tinham diferentes fontes, cores e estilos, era um processo longo, doloroso e caro para reescrever o código. Assim, o CSS foi criado pelo W3C para resolver este problema.

A relação entre HTML e CSS é bem forte. Como o HTML é uma linguagem de marcação (o alicerce de um site) e o CSS é focado no estilo (toda a estética de um site), eles andam juntos.

CSS não é tecnicamente uma necessidade, mas provavelmente você não gostaria de olhar para um site que usa apenas HTML, pois isso pareceria completamente abandonado.

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.

Veja outros trabalhos incríveis [neste site](#) e se inspire.

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>`.

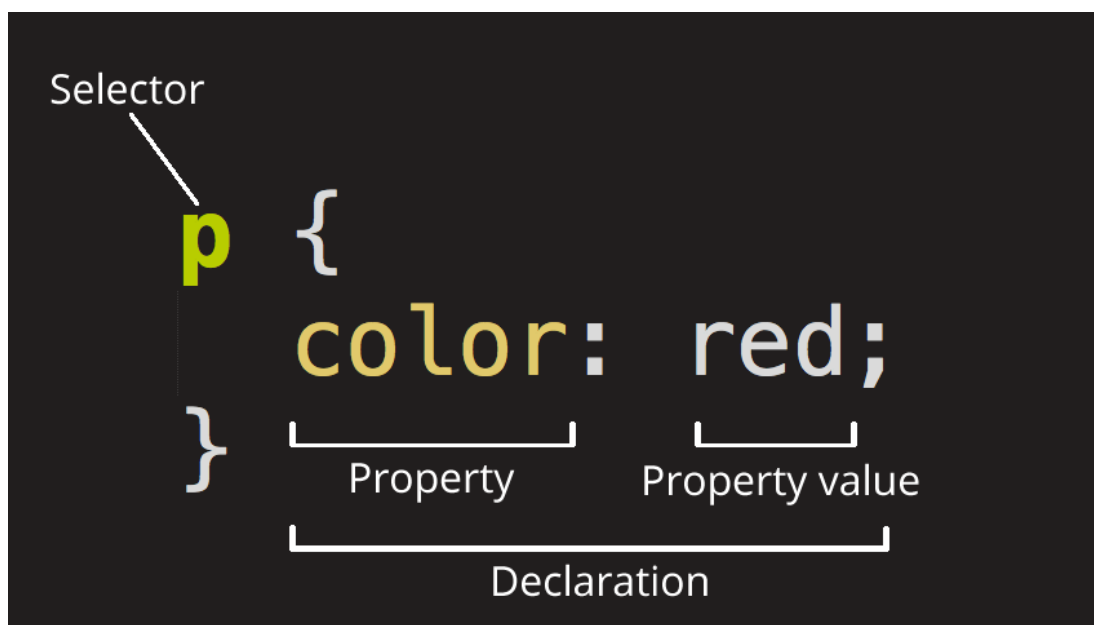
Mas como você estiliza o parágrafo? A estrutura da sintaxe CSS é bem simples. Tem um seletor e um bloco de declaração. Você seleciona um elemento e depois declara o que deseja fazer com ele. Bastante simples, certo?

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:



Anatomia de um comando CSS

O CSS estipula regras para o arquivo em html. Com cada regra é possível estilizar o conteúdo todo ou somente determinados elementos. Por isso entenda, um comando básico é composto por seletor e declarações, que contém propriedade e valor.

SELETOR {PROPRIEDADE: VALOR}

A sintaxe do CSS é muito simples de aprender. O seletor seleciona quais elementos em html receberão a propriedade. Pode ser p (parágrafo) ou body (corpo da sua página). Já a propriedade pode ser a cor ou algo mais específico como cor do fundo (background). E por último o valor, que determina o valor da propriedade.

Vamos simular um exemplo. Digamos que o objetivo é mudar a fonte de uma tag h1. Para isso podemos usar `h1 {font-size: 20px;}`

- h1 – é o seletor. Neste caso selecionamos o h1.
- font-size – é a declaração que contém a propriedade (font-size) e o valor é (20px).

Estilos CSS Interno, Externo e Inline (local)

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.

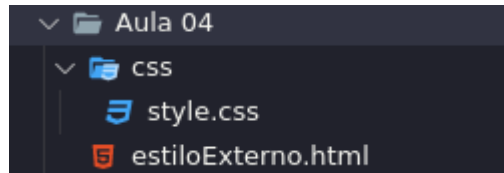
O método **externo** pode ser o mais conveniente. Tudo é feito externamente em um arquivo **.css**. Isso significa que você pode fazer todo o estilo em um arquivo separado e aplicar o CSS a qualquer página desejada. O estilo externo também pode melhorar o tempo de carregamento.

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. Mas pode ser útil, por exemplo, para alterar um único elemento, visualizar rapidamente as alterações ou se não tiver acesso aos arquivos CSS.

CSS externo

O recomendado é fazer a utilização do CSS externo, pois com ele você consegue manter a organização separando os comandos HTML e CSS em arquivos diferentes, mantendo organizado o código.

Para criar um arquivo CSS colocamos no final **.css** como o exemplo abaixo.



Após salvar o arquivo em nosso arquivo HTML, porque é ele quem vai carregar o CSS, o CSS sozinho não faz nada! 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 lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Estilo Externo</title>
9   <link rel="stylesheet" href="css/style.css">
10 </head>
11
12 <body>
13   <div class="verde">
14     <h3>Monetização</h3>
15     <p>configure a monetização</p>
16   </div>
17   <div class="azul">
18     <h3>Configurações </h3>
19     <p>configure seu sistema </p>
20   </div>
21   <div class="vermelho">
22     <h3>Artigos </h3>
23     <p>crie e edite artigos</p>
24   </div>
25 </body>
26
27 </html>
```

CSS Interno

Utilizamos a tag <style> dentro do <head> e aí colocamos todos os comandos que precisamos, ex:

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Testando CSS</title>
9      <style>
10         p {
11             color: #50f01f;
12             font-family: 'Courier New', Courier, monospace;
13         }
14
15         body {
16             background-color: #ddd;
17         }
18     </style>
19 </head>
20
21 <body>
22     <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quod reiciendis earum quia impedit cupiditate iure
23     officiis esse beatae quae, itaque ipsam voluptate ipsum repudiandae, modi labore aliquid laborum tenetur
24     repellendus.</p>
25 </body>
26
27 </html>
```

CSS Inline (Local)

O CSS inline é usado para dar estilo a um elemento HTML específico. Para este estilo de CSS você somente vai precisar adicionar o atributo style para cada tag HTML, sem usar os seletores.

Este tipo de CSS não é realmente recomendado, já que cada tag HTML precisa ser estilizada de maneira individual. Gerenciar o seu site pode se tornar uma tarefa bem difícil de você só usar o CSS inline.

Contudo, o CSS inline no HTML pode ser útil para algumas situações. Por exemplo, em casos onde você não tem acesso aos arquivos CSS ou precisa aplicar estilos para um único elemento.

Vamos dar uma olhada num exemplo. Aqui, nós adicionamos um CSS inline para as tags <p> e <h1>:

```
10
11 <body>
12
13     <h1 style="color:#001479;padding:30px;">Teste CSS inline</h1>
14     <p style="color:#000000;">Lorem ipsum dolor, sit amet consectetur adipisicing elit. Quibusdam accusantium doloribus
15     repellat enim id. Tenetur ipsam repudiandae accusantium laudantium necessitatibus? Obcaecati sint aperiam
16     suscipit ratione numquam sit ipsa vero nihil.</p>
17 </body>
```

Utilizando classes:

Classes são seletores CSS. Utilizaremos seletores CSS para estilizar elementos no HTML, usando o atributo "class", por exemplo: `<div class="blue"></div>` em referência à uma classe CSS blue.

Para aplicar a formatação é bem simples basta colocar um "."(ponto) e a palavra-chave da classe para aplicar a formatação do css nela.

```
9  <style>
10  .azul {
11      color: blue;
12  }
13  </style>
14  </head>
15
16  <body>
17      <div class="azul">Textano classes no css</div>
18  </body>
```

Aula 05

DIV <div> :

O **elemento de divisão** <div> é um container genérico para conteúdo de fluxo, que de certa forma não representa nada. Ele pode ser utilizado para agrupar elementos para fins de formatação

```
<div>
  <p>Qualquer tipo de conteúdo aqui.</p>
</div>
```

Linha de separação <hr> :

O elemento HTML <hr> representa uma quebra temática entre elementos de nível de parágrafo (por exemplo , uma mudança da cena de uma história, ou uma mudança de tema com uma seção). Nas versões anteriores do HTML, representava uma linha horizontal. Pode continuar sendo exibida como uma linha horizontal nos navegadores, mas agora está definida em termos semânticos, em vez de termos de apresentação.

```
<p>Este é o primeiro parágrafo do texto. Este é o primeiro parágrafo do texto.  
Este é o primeiro parágrafo do texto. Este é o primeiro parágrafo do texto.</p>  
<hr>  
  
<p>Este é o segundo parágrafo do texto. Este é o segundo parágrafo do texto.  
Este é o segundo parágrafo do texto. Este é o segundo parágrafo do texto.</p>
```

Atributos

align:

Define a regra de alinhamento na página. Se nenhum valor for especificado, o valor padrão é left.

color:

Define a regra da cor através do nome da cor ou um valor hexadecimal.

size:

Define a altura, em pixels.

width:

Define a regra de comprimento na página através de um valor em pixel ou porcentagem

Aula 06

Hyperlinks <a> :

O elemento <a> em HTML (ou elemento âncora), com o atributo href cria-se um hiperligação nas páginas web, arquivos, endereços de emails, ligações na mesma página ou endereços na URL. O conteúdo dentro de cada <a> precisará indicar o destino do link.

Atributo:

href:

A URL para a qual o hiperlink aponta. Links não se restringem a URLs baseadas no protocolo HTTP — eles podem utilizar qualquer tipo de URL suportado pelos browsers:

- Seções de página com fragmentos URL
- Pedacos de arquivos de mídia com fragmentos da própria mídia
- Números de telefone com tel: URLs
- Email addresses with(Endereço de email com) mailto: URLs

OUTROS ATRIBUTOS:

<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a#atributos>

Links internos:

Links internos são aqueles que apontam para uma página dentro do próprio site. São exemplos de links internos os paginas produtos, empresa, serviços, contato, localização, etc.

- Página de produtos
- Página de serviços

Links externos

Links externos são aqueles que apontam para fora do próprio site. Quando criamos um link para outros sites estamos criando links externos. Tecnicamente links internos e externos são iguais, mudando apenas o destino dos mesmos.

- `Uol`

No caso de links externos é importante usarmos o atributo `target="_blank"` que indica que a página será aberta em uma nova janela.

- `Terra`

É importante observar que ao fazermos um link externo, usamos o endereço completo da página destino. Isto é necessário pois caso informássemos apenas www.uol.com.br, por exemplo, ele iria tentar abrir o site do UOL dentro do nosso próprio site, portanto o `https://` indica ao link para fazer uma busca externa.

Links verticais

Links verticais são usados para apontarmos para uma região de uma página, podendo ser a própria página que contém o link. Imagine que você escreveu um artigo bem extenso e para facilitar a leitura você criou um índice colocando links neste índice. Quando um usuário clicar num link do índice ele será deslocado para o local do texto onde contém o texto, mas não mudou de página.

- `Capítulo 1`
- `Capítulo 2`

Observe que no atributo `href` não colocamos nenhum endereço de página, mas sim um termo que chamamos de âncora, ou seja, estamos apontando o nosso link para uma âncora chamada `capitulo1` e `capitulo2`, respectivamente.

Neste caso precisamos criar as âncoras também. Para isto insira o código abaixo no local onde inicia o primeiro e segundo capítulo, respectivamente.

- ``
- ``

Aula 07

Imagens :

Para colocar uma única imagem em uma página da web, usamos o elemento . Isso é um elemento vazio (quer dizer que não possui conteúdo de texto ou tag de fechamento) que requer no mínimo um atributo para ser útil — src (às vezes pronunciado como seu título completo, source). O atributo src contém um caminho apontando para a imagem que você deseja incorporar na página, que pode ser uma URL relativa ou absoluta, da mesma maneira que o valores de atributo href do elemento [<a>](#).

Por exemplo, se sua imagem for chamada dinossauro.jpg, e está no mesmo diretório de sua página HTML, você poderia inserir a imagem assim:

```

```

Se a imagem estivesse em um subdiretório de imagens, que estivesse dentro do mesmo diretório da página HTML (que o Google recomenda para fins de indexação/SEO), então você a incorporaria da seguinte maneira:

```

```

Atributos:

alt:

Este atributo define um texto alternativo que descreve a imagem. Os Usuários irão ver o texto se a URL da imagem estiver errada, a imagem não está em um dos formatos suportados ou até a imagem ser baixada .

height:

A altura da imagem em pixels ou em porcentagem.

width:

Largura da imagem em pixels ou porcentagem

src:

URL da imagem, este atributo é obrigatório para o elemento .

Propriedades de css:**Imagens arredondadas:**

Use a propriedade **border-radius** para criar imagens arredondadas:

```
10 <style>
11   .img1 {
12     border-radius: 8px;
13   }
14 </style>
15
16 <body>
17   
18 </body>
```

Imagem transparente:

A propriedade de opacidade pode ter um valor de 0.0 a 1.0 ou 0% a 100%. Quanto menor o valor, mais transparente:

```
10 <style>
11   .img {
12     opacity: 0.5;
13   }
14 </style>
15
16 <body>
17   
18 </body>
```

Filtros de imagens:

As funções previstas nas especificações para aplicação dos efeitos de filtragem são listadas a seguir:

- blur()
- brightness()
- contrast()
- grayscale()
- invert()
- saturate()
- sepia()
- hue-rotate()
- drop-shadow()

link site : <https://www.maujor.com/tutorial/css3-filters.php>

Aula 08

Tabela <table> :

O que são tabelas?

As tabelas são listas de dados em duas dimensões e são compostas por linhas e colunas. Portanto, são muito utilizadas para apresentar dados de uma forma organizada. Um grande exemplo de como as tabelas são importantes é o crescente uso do programa de planilha de cálculo, requisito obrigatório de seleção para muitas empresas. Mas e para apresentar uma tabela no seu site, como funciona? Para isso, existe a possibilidade de criar tabelas através do próprio HTML, de forma rápida e simples, por meio de uma tabela HTML.

A estrutura básica de uma tabela HTML

As tags que vão formar a estrutura básica de uma tabela em HTML são as tags <tr> e <td>. A tag <tr> representa uma linha e a tag <td> representa uma célula. Desta forma, a criação de colunas em uma tabela HTML é feita automaticamente através da quantidade de células incluídas dentro de uma linha.

Célula de título da tabela HTML

Caso queira incluir uma ou mais células que representam títulos, ganhando um destaque em relação às demais células, pode utilizar a tag <th>.

Tags estruturais da tabela

Anteriormente, vimos que a estrutura básica de uma tabela consiste nas tags <tr>, <td> e as células de título <th>. Porém, além dessas, também há tags que servem para estruturar melhor a tabela. São elas:

- <thead></thead> – Esta tag representa o cabeçalho da tabela, geralmente composta por células título;
- <tbody></tbody> – Essa tag representa o corpo da tabela;

- `<tfoot></tfoot>` – Essa tag representa o rodapé da tabela.

Portanto, essas tags tem como principal efeito manter uma semântica adequada para suas tabelas.

Células que abrangem mais que uma linha/coluna

Em algumas situações, temos a necessidade de utilizar uma mesclagem de célula, ou seja, criar uma célula que abrange mais do que uma linha ou coluna. Dessa forma, podemos utilizar os atributos `colspan` e `rowspan`. O `colspan="x"` vai fazer uma mesclagem de colunas, e o `rowspan="x"` vai mesclar linhas, bastando substituir o x pelo número de colunas ou linhas que deseja ocupar. Além disso, podemos também mesclar os dois em uma mesma célula. Com isso, vamos ver os exemplos a seguir: