

# Trilha Fundamental da Rocketseat

## → HTML

### VISUAL STUDIO CODE

- É baixada uma extensão para ajudar a visualizar o .html (HTML Preview).
- Ctrl + Shift + P e depois seleciona o HTML: Open Preview.
- tudo o que é digitado no arquivo.html vai sendo mostrado no “visualizador de html”.
- Alt + Z quebra o texto do código, isso possibilita ver todo o texto em uma tela só, sem precisar arrastar para o lado.

### COMANDOS HTML

Se, antes de apertar o Enter, eu colocar **\*n°** depois do comando, ele se repete por n° vezes

Se digitar “../” antes do caminho de algum documento, ao invés de entrar no local, vai sair.

|   |  |
|---|--|
| <h1> <u>Título</u> </h1>  | se digitar só h1 + Enter, o VSC já completa a tag, vai de h1 até h6  |
| <p> <u>texto</u> </p>   | para fazer parágrafo   |
| <ol> <u>comandos li</u> </ol>   | 1. lista ordenada, com números   |
| <ul> <u>comandos li</u> </ul>   | ● lista não ordenada, com pontinhos  |
| - <li> <u>alguma coisa</u> </li>  | lista o que tiver dentro, tem que estar entre <u>ol</u> ou <u>ul</u>   |
| <dl> <u>dt e dd</u> </dl>   | lista de descrições  |
| - <dt> <u>título</u> </dt>  | título do que vai ser descrito   |
| - <dd> <u>descrição</u> </dd>   | descrição do título  |
| <!-- <u>comentário</u> -->  | os comentários não aparecem na página que está sendo desenvolvida  |
| <a href= <u>”link”</u> > <u>título do link em azul</u> </a>                               | link   |
| <a href= <u>”link”</u> target= <u>”_blank/_self”</u> > <u>título do link em azul</u> </a> | link normal, só que se tiver o <i>target= ”_blank”</i> , quando clicar no link ele abre uma nova guia e o <i>target= ”_self”</i> , quando clicar ele abre o site na mesma guia, que é o padrão |
| <em> <u>palavra ou frase</u> </em>  | vai deixar em <i>itálico</i>   |
| <strong> <u>palavra ou frase</u> </strong>  | vai deixar em <b>negrito</b>   |
| <div> <u>alguma coisa aqui</u> </div>   | serve para separar elementos em blocos   |
| <span> <u>alguma coisa aqui</u> </span>   | serve para separar elementos no código, mas aparece em linhas no site  |

|  |   |
|--|---|
| <code>&lt;br/&gt;</code> ou <code>&lt;br&gt;</code> ou <code>&lt;/br&gt;</code>            | <i>break</i> , quebra a linha, o que tá depois dele fica na linha de baixo  |
| lorem  | ele escreve um texto meio grandinho “aleatório”   |
| <code>&lt;img<br/>src=”https://source.unsplash.com/r<br/>andom” alt=””&gt;</code>          | imagem aleatória do Google  |
| <code>&lt;code&gt;alguma coisa&lt;/code&gt;</code>   | muda coloração e indica que ali tem um código simples   |
| <code>&lt;pre&gt;&lt;code&gt;alguma<br/>coisa&lt;/code&gt;&lt;/pre&gt;</code>              | com o <code>&lt;code&gt;</code> dentro do <code>&lt;pre&gt;</code> as tabulações são efetuadas, é melhor para códigos grandes e complexos |
| <code>&lt;blockquote<br/>cite=”link”&gt;citação&lt;/blockquote&gt;</code>                  | faz citação por aí  |
| <code>&lt;cite&gt;&lt;/cite&gt;</code>   | faz citação por aí, também  |
| <code>&lt;q&gt;&lt;/q&gt;</code>   | faz citação por aí, também  |
| <code>&lt;abbr title=”significado de uma<br/>abreviação”&gt;abreviação&lt;/abbr&gt;</code> | digita algo em “abreviação”, na parte do significado explica o que é, se deixar o mouse parado em cima dá para ver                        |
| <code>&lt;address&gt;contato&lt;/address&gt;</code>  | tipo, eu criei essa página, vou deixar meu nome e email aqui  |
| <code>&lt;button&gt;&lt;/button&gt;</code>   | cria um botão   |

→ Elementos Vazios (não tem esse fechamento `</comando>`)

|   |        |
|---|--------|
| <code>&lt;img src=”.png aqui” alt=”texto/título para a imagem”&gt;</code> | imagem |
| <code>&lt;input type=”text”&gt;</code>                                    |        |

→ Atributos Globais Mais Utilizados

Tem mais atributos em: [developer.mozilla.org](https://developer.mozilla.org)

|                                      |  |
|--------------------------------------|--|
| <code>class”nome-classe”</code>      | cria uma classe, para criar estilo css, ou para acessar no js  |
| <code>contenteditable</code>         |  |
| <code>data-qualquer-nome-aqui</code> |  |
| <code>hidden</code>                  | esconde uma tag  |
| <code>id</code>                      | tipo subtópicos do class, se colocar <code>&lt;li&gt;&lt;a href=”#___”&gt;Título&lt;/a&gt;&lt;/li&gt;</code> no começo e depois um título com <code>&lt;h1 id=”___”&gt;Título&lt;/h1&gt;</code> e nos ___ colocar a mesma “etiqueta”, ele vira um “sumário interativo” |

|                      |  |
|----------------------|--|
| style                | dá para estilizar o texto  |
| tabindex="número"    | se apertar tab, ele segue na ordem dita pelo código                        |
| title="alguma coisa" | textinho que aparece quando deixa o cursor do mouse parado em cima de algo |

### → Conteúdos do Texto e Caracteres Reservados

|        |  |
|--------|--|
| &nbsp; | é como se fosse o TAB que a gente dá no texto                        |
| &lt;   | coloca < no texto, se colocar o caractere direto pode bugar o código |
| &gt;   | coloca > no texto, se colocar o caractere direto pode bugar o código |
| &amp;  | coloca & no texto, se colocar o caractere direto pode bugar o código |
| &quot; | coloca “ no texto, se colocar o caractere direto pode bugar o código |
| &apos; | coloca ‘ no texto, se colocar o caractere direto pode bugar o código |

### → Anatomia de um documento HTML

|   |   |
|---|---|
| <!DOCTYPE html>   | é necessário para que um navegador entenda que você está trabalhando com HTML                                 |
| <html lang="en">head e body</html>                                      | o lang="en" configura a linguagem para inglês, pt para português  |
| <head>__</head> <i>cabeça</i>   | contém configurações importantes para o site, mas que o usuário não vê  |
| - <meta charset="utf-8">  | comando que vai possibilitar que o site receba vários tipos de caracteres especiais                           |
| - <meta name="viewport" content="width=device-width,initial-scale=1.0"> | possibilita que o site seja visto tanto em celulares, quanto em computadores, etc.                            |
| - <meta name="author" content="nome">                                   | colocar o author da página no código  |
| - <title>Título</title>   | comando que dá o nome para a guia do Chrome, por exemplo  |
| - <link rel="icon" href="link ou caminho da imagem"/>                   | comando que possibilita colocar um favicon básico, aquela imagem pequena na guia do Chrome junto com o título |
| - <link rel="apple-touch-icon-precompose"                               | mesmo do de cima, mas para iPhone não-retina, iPod Touch e dispositivos 2.1+; tem várias outras opções e      |

|                                       |  |
|---------------------------------------|--|
| d" href="link ou caminho da imagem"/> | resoluções na parte “Trilha Fundamental > Guia Estelar de HTML > Cabeçalho > Favicon |
| <body></body> <i>corpo</i>            | aqui ficam as coisas que o usuário vai ver e ter acesso, conteúdo (<p>, <h1>)        |
| !                                     | no VSC ele já faz esse “cabeçalho” automaticamente                                   |

→ **Tabela**

|   |  |
|---|--|
| <table></table>                             | para começar a tabela  |
| <caption></caption>                         | título da tabela   |
| <colgroup></colgroup>                       | dá para estilizar a coluna   |
| <col>                                       | não acontece nada com a coluna   |
| <col style="background-color: <u>cor</u> "> | vai colorir o fundo da coluna (background) de acordo com a cor que estiver em <u>cor</u> (em inglês), pode colocar um <u>span=nº</u> depois de col, para pintar mais colunas |
| <tr></tr>                                   | para escrever na linha, mantém na mesma linha  |
| <thead></thead>                             | cabeçalho, todos os <th> são cabeçalhos, que descrevem o conteúdo da tabela  |
| <th></th>                                   | primeira linha, descreve os itens da tabela em colunas separadas, para os títulos  |
| <th rowspan="n"></th>                       | faz a coluna ocupar o tanto de linhas no nº  |
| <th colspan="n"></th>                       | faz a linha ocupar o tanto de colunas no nº  |
| <th colspan="n" scope="colgroup"></th>      | mesmo do de cima, mas o <u>scope</u> identifica que a coluna tem função de agrupar outras colunas, tipo em uma coluna tem mais duas colunas                                  |
| <th scope="col"></th>                       | no exemplo: uma coluna tem mais duas colunas, o <u>scope</u> serve para identificar esses dois cabeçalhos de colunas   |
| <th scope="row"></th>                       | para identificar que é um cabeçalho de linha   |
| <tbody></tbody>                             | corpo  |
| <td></td>                                   | escreve na linha da tabela, em colunas separadas, para os dados  |
| <tfoot></tfoot>                             | rodapé, usa <tr> e <td> também   |

## → CSS

CSS = Cascading Style Sheet (Folha de Estilo em Cascata)

“HTML é a estrutura e o CSS é a beleza”

Não é linguagem de programação, é uma linguagem style sheet.

Pode fazer pelo VS Code e ir olhando pela página atualizada do Chrome, ou fazer direto pelo codepen.io

Para olhar o CSS por trás de um site bonito: F12 ou Fn+F12 aí vai em “Styles”.

|                         |                                 |
|-------------------------|---------------------------------|
| <code>/*lalala*/</code> | comentários não afetam o código |
|-------------------------|---------------------------------|

## → Anatomia CSS

`<div id=“random_name” class=“random_name_2”>` comandos aqui `</div>`

|  |  |  |   |
|--|--|--|---|
| <u>selector</u> {<br><u>property</u> : <u>value</u> ;<br>} | <u>* global selector</u><br><br>h1, h2, p, div<br><u>element/type selector</u><br><br>.red .m-4 .random_name<br><u>class selector</u><br><br>#box #container<br><u>ID selector</u> | <u>_0. global selector</u><br><u>_1. element/type selector</u><br><u>_10. classes e attribute selector</u><br><u>_100. ID selector</u><br><u>_1000. inline</u> | no <u>selector</u> coloca algum comando do HTML, aí o que colocar em <u>property</u> e <u>value</u> vai mudar em todos os comandos do HTML, <u>declaration</u> é tudo que está entre {} |
| - color: <u>cor</u>  | muda a cor do meu texto, por exemplo   |  |   |
| - font-size: n°px  | altera o tamanho da fonte, em px (pixels)  |  |   |
| - background: <u>cor</u>                                   | muda a cor do fundo de algo  |  |   |
| - width: n°px  | altera a largura de algo, em px  |  |   |
| - height: n°px   | altera a altura de algo, em px   |  |   |
| - margin: n°px   | altera a margem, espaçamento de algo, em px, é em volta da caixa, descola o texto do limite da pági<br>na  |  |   |
| - padding: n°px  | parece margin, só que é dentro da caixa  |  |   |
| - border: n°px <u>solid cor</u>                            | cria uma borda em volta do selector, n°px = largura da borda, solid = sólida   |  |   |

## → Inserir CSS no HTML

- por linha, inline, código por código, já no HTML
  - `<h1 style=“color: red”>Título</h1>`
  - **Título**
- no head (não precisa ser necessariamente nele)
  - `<style>h1 { color: red; }</style>`
  - `<h1>Título</h1>`

- **Título**
- no head, mas em outro arquivo
  - <link rel="stylesheet" href="style.css"> (nome do arquivo)
  - faz seu arquivo em HTML normal
  - no arquivo .css estiliza separadamente
- @import ' ' (só no CSS)
  - 
  - para incluir um CSS externo fonts.google.com, pega o link
  - mas deixa a página lenta
  - pega o font-family: \_\_\_\_\_ e blablabla

Comando **!important** as vezes não dá para estilizar um texto de outro modelo, aí usa ele depois da cor, por exemplo, mas não é uma boa prática e quebra o “sentido cascata”.

**@media** regras condicionais para dispositivos, tem haver com responsividade

**@font-face** para colocar fontes externas

**@keyframes** para animações

### → Shorthand

Junção ou resumo de um comando, vai disso:

```
background-color: #000;
background-image: url(images/bg.gif);
background-repeat: no-repeat;
background-position: left top;
```

Para isso:

```
background: #000 url(images/bg.gif) no-repeat left top;
```

Não importa a ordem debaixo, tem que escrever os dois, mas ele sempre vai pegar o que foi escrito por último; nem todas as propriedades aceitam o shorthand.

### → Funções

valor(alguma\_coisa) — exemplo:

```
{
  color: rgb(225, 0, 100);
  width: calc(100% - 10px);
}
```

As funções, parênteses, recebem argumentos, não podem estar vazias.

### → Vendor Prefixes

```
p {
  -webkit-background-clip: text;      /* Chrome, Safari, iOS e Android */
  -moz-background-clip: text;         /* Mozilla (Firefox) */
  -ms-background-clip: text;          /* Internet Explorer */
  -o-background-clip: text;           /* Opera */
}
```

pode procurar mais em: <https://ireade.github.io/which-vendor-prefix>

pode consultar quais comandos precisam do vendor prefix: <https://caniuse.com>

### → **Nem tudo são Píxeis**

Se tiver `<color>` ou `<length>` são ‘*values*’ ou ‘*data types*’.

### → **Tipos numéricos**

`<integer>` número inteiro, positivos e negativos

`<number>` número decimal, inteiro com vírgula

`<dimension>` number com unidade, tipo 10.5px

`<percentagem>` fração de algum número, 50% de algo

### → **Unidades Comuns**

`<length>` valor de distância: px, em, vw, vh

`<angle>` ângulo: deg, rad, turn

`<time>` tempo: s, ms

`<resolution>` resoluções para dispositivos: dpi

`<position>` posições, conjunto de coordenadas 2D: top, right, bottom, left e center (não confundir com a propriedade)

px - pixel = 1/96in

cm - centímetro = 96px/2,54

in - inches (polegadas) = 2,54cm = 96px

em - tamanho da font pai

rem - tamanho da font do elemento raiz (root/html)

vw - 1% da viewport width, parte branca da página (vertical)

vh - 1% da viewport height, parte branca da página (horizontal)

### → **Funções**

rgb( ) - para cor

hsl( ) - para cor

url( ) - url

calc( ) -

## → Uma caixa dentro da outra

Box Model - caixa retangular com propriedades 2D, cada elemento na página é considerado uma caixa.

|   |   |
|---|---|
| Tamanho: largura / altura   | width / height**  |
| Conteúdo  | content   |
| Bordas<br>px   em   | border***<br>→ border-top   border-right   border-bottom   border-left<br>→ <border-style>   <border-color><br>→ style: solid   dotted   dashed   double   groove   ridge   inset   outset<br><a href="#">Documentação do MDN</a> |
| Preenchimento interno (margem dentro da caixa)<br>px   %   em     | padding*<br>→ padding-top   padding-right   padding-bottom   padding-left   |
| Espaços fora da caixa<br>px   %   auto                            | margin<br>→ margin-top   margin-right   margin-bottom   margin-left<br>→ margin: auto → tenta fazer um cálculo automático <b>nas laterais</b>   |
| outline<br>px   em  | outline****   |
| Adicionar um padding sem alterar a width ou o height              | box-sizing: border-box;   |
| transformar display inline em block                               | display: block;   |
| transformar display block em inline                               | display: inline;  |
| display block   | display inline  |
| ocupa toda a linha, colocando o próximo elemento abaixo desse     | elemento ao lado do outro   |
| width e height são respeitados                                    | width e height não funcionam  |
| padding, margin e border irão funcionar normalmente               | somente valores horizontais de margin, padding e border   |
| Ex.: `<p> <div> <section>`, todos os headings `<h1> <h2> <h3>...` | Ex.: `<a> <strong> <span> <em>`   |

\*padding – ele altera o tamanho da caixa

\*\*width / height – a largura ou a altura escolhidas vão afetar o content, mas não o padding



**\*\*border** – ela é aplicada fora da caixa, e o tamanho de width e height ficam “intactos”, para aplicar o border dentro da caixa é ``box-sizing: border-box``

**\*\*\*\*outline** – não modifica o tamanho da caixa por não ser parte de Box Model, pode ser diferente de retangular, não permite ajustes individuais, muito usada pelo user agent para acessibilidade

**margin-collapsing** – se a margin-bottom de uma caixa for um  $x$  e o elemento abaixo dele tiver uma margin-top  $y$ , elas se juntam e ao invés de ficar  $x+y$ , ele considera apenas o maior valor. Quando um elemento está um ao lado do outro isso não acontece, os valores são somados normalmente.