

Manual do Markdown para GitHub

Quando começamos a produzir repositórios no GitHub, nosso principal objetivo é apenas subir nossos arquivos. Mas para começar a organizar nossos arquivos ou interagir com *issues* ou *pull requests*, chega uma hora que precisamos aprender um tal de **markdown**. Mas o que é isso? Vamos começar a aprender como criar nossos conteúdos usando markdown. Vem comigo?



Você tem todo o direito de usar esse material para seu próprio aprendizado. Professores também podem ter acesso a todo o conteúdo e usá-los com seus alunos. Porém todos o que usarem esse material - seja para qual for a finalidade - deverão manter a referência ao material original, criado pelo **Prof. Gustavo Guanabara** e disponível no endereço do seu repositório público <https://github.com/gustavoguanabara/>. Este material não poderá ser utilizado em nenhuma hipótese para ser replicada - integral ou parcialmente - por autores/editoras para criar livros ou apostilas, com finalidade de obter ganho financeiro com ele.



O que é markdown?



Você já deve ter ouvido falar que a HTML não é uma **linguagem de programação** e sim uma **linguagem de marcação**. Pois saiba que HTML não é a única linguagem da categoria. Existem outras, inclusive a linguagem **markdown** (representada como uma letra M, seguida de uma seta para baixo).



Criada em 2004 pelo designer/blogueiro/podcaster **John Gruber** (esquerda) e pelo programador/ativista **Aaron Swartz** (co-fundador do *Reddit*, envolvido na *Creative Commons* e dos *feeds RSS*), a linguagem markdown tem como objetivos principais facilitar a criação de conteúdo formatado e ser mais simples que a HTML.

O principal motivo de se criar uma outra linguagem de marcação é que, segundo Gruber, códigos HTML se tornam difíceis de ler (baixa *readability*), pois as marcas textuais como `<body>` fazem com que as palavras do texto original não possam ser compreendidas facilmente.

Basicamente, no lugar de marcas textuais, as formatações se resumem a símbolos simples digitados pelo teclado que marcam partes do texto e geram resultados visuais. Atualmente, é possível usar markdown até mesmo em editores como o Microsoft Office, o LibreOffice Writer, que são compatíveis com arquivos do tipo **.md** ou **.markdown**. Além disso, o **GitHub** também usa a linguagem para especificar os arquivos **README**, criados nas raízes dos repositórios, além de ser usada na gestão de **issues** e de **pull requests** na plataforma.

Outra grande preocupação dos criadores da linguagem markdown é que a conversão entre ela e a HTML fosse muito simples e fácil de fazer.



Para isso, a markdown tinha que ser padronizada e por isso foi criado o **CommonMark**, que hoje é adotado por vários softwares, sites e serviços online.



REFERÊNCIAS: Para uma referência simples e objetiva da linguagem markdown, acesse os sites oficiais da organização e a especificação criada pela GitHub (*todos em Inglês*):

<https://www.commonmark.org>

<https://www.markdownguide.org>

<https://guides.github.com/features/mastering-markdown/>

A dica é sempre praticar bastante o uso da linguagem para que todas as marcas possam ser facilmente lembrada com o tempo. Você pode fazer isso adicionando recursos aos seus arquivos README.md que sempre estão na raiz dos repositórios no GitHub.

Markdown simples

Vamos começar com algumas formatações básicas.

Itálico

Para colocar uma palavra em itálico, podemos colocá-la entre asteriscos simples ou entre underlines simples.

Ex:

```
CursoemVideo  
*CursoemVideo*
```

Resultado: *CursoemVideo*

Negrito

Para usar o negrito, basta colocar asteriscos duplos ou underlines duplos para delimitar o termo.

Ex:

```
CursoemVideo  
**CursoemVideo**
```

Resultado: **CursoemVideo**

Riscado (strike)

Para riscar um texto, coloque-o entre dois símbolos de til.

Ex:

```
~~CursoemVideo~~
```

Resultado: ~~CursoemVideo~~

Dá pra misturar?

É possível também juntar as formatações de forma hierárquica, como por exemplo:

Ex:

```
**CursoemVideo**
```

Resultado: ***CursoemVideo***



IMPORTANTE: Lembre-se de não colocar espaços entre os símbolos e os termos que eles vão formatar.

Listas

Listas numeradas

Para criar uma lista numerada, comece a linha com um número qualquer, seguido de um ponto. O primeiro número da lista vai indicar onde ela começa, os itens seguintes, se estiverem em linhas consecutivas, vão continuar a sequência.

Ex:

```
1. Brasil
1. Alemanha
5. França
0. Bélgica
```

Resultado:

```
1. Brasil
2. Alemanha
3. França
4. Bélgica
```



IMPORTANTE: Se você inserir três ou mais espaços antes do marcador, ele será colocado em outro nível.

Listas com marcadores

Basta usar um asterisco seguido de um espaço ou um traço seguido de espaço no início de uma linha para que ele seja parte de uma lista demarcada.

Ex:

```
* Opção 1
* Opção 2
  * Opção 3
* Opção 4
```

Resultado:

- Opção 1
- Opção 2
 - Opção 3
- Opção 4

Listas de tarefas

Colocando um sinal - [] ou - [x] antes de cada item, criamos listas de tarefa com itens marcados ou desmarcados.

Ex:

- [] pão
- [x] leite
- [] manteiga

Resultado:

- ☐ pão
- ☒ leite
- ☐ manteiga

Títulos e linhas horizontais

Para criar títulos, podemos usar de uma # até seis ##### para definir o nível do título para que o conteúdo fique organizado corretamente.

Para criar linhas horizontais, podemos usar três traços --- ou três asteriscos ***

Ex:

Texto principal

Teste

Resultado:

Texto principal

Teste

Imagens

Para inserir uma imagem, vamos usar o símbolo ![]() onde a descrição fica entre colchetes e o endereço da imagem entre parênteses.

Ex:

![CursoemVideo](https://cursoemvideo.com/logo.png)

Resultado:



Links

Para inserir links, a simbologia é semelhante à usada com as imagens, apenas removemos o sinal de exclamação.

Ex:

[Meu GitHub](https://gustavoguanabara.github.io)

Resultado:

[Meu GitHub](https://gustavoguanabara.github.io)

Trechos de código

Como o **GitHub** é uma rede social para programadores, nada mais interessante do que aprendermos a demarcar nossos códigos e comandos. Podemos usar dois tipos de marcação:

Comando isolado

Se for citar um comando apenas, basta colocá-lo entre crases.

Ex:

Gostaria de maiores informações sobre o comando `window.document.querySelector()` da linguagem JavaScript.

Resultado:

Gostaria de maiores informações sobre o comando `window.document.querySelector()` da linguagem JavaScript.



O QUE MUDOU? Todo código delimitado em Markdown ficará com uma fonte monoespaçada e com fundo acinzentado para dar destaque. Para códigos com várias linhas, a *indentação* é realizada através de quatro espaços por nível desejado.

Trecho de código

Para compartilhar um código com mais linhas, basta colocar todas elas dentro de um grupo delimitado entre três crases consecutivas (sem espaço entre elas).

Ex:

Olha só o programa que criei em ****Python****:

```
```\nnum = int(input('Digite um número: '))\nif num % 2 == 0:\n    print(f'O valor {num} é PAR')\nelse:\n    print(f'O valor {num} é ÍMPAR')\nprint('Fim do Programa')\n```
```

Resultado:

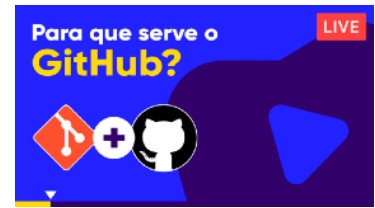
Olha só o programa que criei em **Python**:

```
num = int(input('Digite um número: '))\nif num % 2 == 0:\n print(f'O valor {num} é PAR')\nelse:\n print(f'O valor {num} é ÍMPAR')\nprint('Fim do Programa')
```

Viu como o código acima fica bem destacado e mais fácil de ser lido e compreendido?

# Eu já falei sobre isso no YouTube?

Eu sei que às vezes as pessoas gostam mais de assistir vídeos do que ler livros, e é por isso que eu lanço há anos materiais no canal Curso em Vídeo no YouTube. O link que vou compartilhar contigo tem o conteúdo explicado como você leu aqui, só que de forma mais ilustrada. Reserve um tempo dos seus estudos para assistir esse vídeo todo.



[https://www.youtube.com/playlist?list=PLHz\\_AreHm4dm7ZULPAmadvNhH6vk9oNZA](https://www.youtube.com/playlist?list=PLHz_AreHm4dm7ZULPAmadvNhH6vk9oNZA)