

Estudo Autodidata Git e GitHub



Aula: 8

Tema: Linguagem Markdown

O que é Markdown?

Markdown é uma linguagem de marcação leve criada por John Gruber com a ajuda de Aaron Swartz. Seu objetivo é permitir que as pessoas escrevam usando um formato de texto simples e legível, que possa ser convertido facilmente para HTML.

Por que Markdown é importante no GitHub?

O GitHub usa Markdown extensivamente:

- README.md: página inicial dos repositórios.
- Issues e Pull Requests: para descrever problemas, propor mudanças com formatação.
- Wikis: documentação interna de projetos.
- Arquivos .md (como CONTRIBUTING.md, CODE_OF_CONDUCT.md, etc.)

GitHub usa uma variação chamada **GitHub Flavored Markdown (GFM)**, que estende o Markdown padrão com funcionalidades extras.

Sintaxe Markdown (com exemplos)

Cabeçalhos

Título H1

Título H2

Título H3

Título H4

Título H5

Título H6

Ênfase

itálico ou itálico

****negrito**** ou negrito

~~riscado~~

Listas

1. Lista não ordenada

- Item 1

- Item 2

- Subitem

- * Outro item

2. Lista ordenada

1. Primeiro
2. Segundo
1. Subitem

Links

[Texto do link](<https://github.com>)

Imagens

![Texto alternativo](URL da imagem)

Citação

> Isso é uma citação.

>> Citação aninhada

Bloco de código

Use o comando `npm install` em um arquivo .md

Linguagem para destaque de sintaxe

```
'''js function hello() { console.log("Olá, mundo!"); } '''
```

```
'''
```

```
###
```

```
'''
```

Tabelas

Nome	Idade	Cidade
------	-------	--------

-----	-----	-----
-------	-------	-------

Davi	22	São Paulo
------	----	-----------

Angela	22	Lima
--------	----	------

✓ Checklists

- [x] Tarefa feita

- [] Tarefa pendente

🔗 Referência de links

[Google][1]

[1]: <https://www.google.com>

Recursos específicos do GitHub Flavored Markdown (GFM)

1. Fechamento automático de issues/pull requests

Na descrição de um commit ou PR:

Fixes #42

Isso fecha automaticamente a *issue 42* ao fazer merge.

🔗 Situação:

Você tem uma *issue* aberta com o número **#42**, com o título:

#42 — Corrigir bug no login

Você então faz um commit com essa mensagem:

`" git commit -m "Corrigido bug no login. Fixes #42 " "`

Ou, na descrição de um Pull Request, você escreve:

`" Este PR corrige o problema de autenticação.`

Fixes #42 "

✓ Resultado:

Assim que esse **commit ou PR for feito merge (integrado ao branch principal)**, o GitHub **automaticamente fecha a issue #42** e coloca na issue o seguinte comentário:

"Closed by #43" (onde #43 seria o número do Pull Request, por exemplo)

Outras palavras-chave que funcionam:

Você pode usar:

- Fixes #n
- Closes #n
- Resolves #n

Uma curiosidade sobre o comando Fixes #n é que pode ser utilizado mais de uma vez em diferentes Commit's

Fixes #42, closes #51, resolves #53

2. Referência cruzada

Digamos que você esteja escrevendo um comentário na **issue #58**, e queira mencionar a **issue #42** porque ela está relacionada.

“Esse problema é muito parecido com o que foi relatado em **#42**. Podemos aplicar uma solução semelhante.”

3. Menções

@usuario

4. Emojis

:smile: :rocket: :bug:

Lista completa de emojis suportados: [GitHub Emoji Cheat Sheet](https://github.com/ikatyang/emoji-cheat-sheet)
(<https://github.com/ikatyang/emoji-cheat-sheet>)

Exemplos de uso no GitHub

Meu Projeto

Este projeto é incrível 😁

Tecnologias

- JavaScript
- Node.js
- Markdown

Como usar

```

bash

git clone https://github.com/usuario/repositorio.git

cd repositorio

npm install

npm start

```