

# Estrutura de um arquivo Rmd

Prof. Frega

25-03-2024

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>O que é um arquivo Rmd</b>                            | <b>1</b> |
| <b>2</b> | <b>Inserindo código e texto</b>                          | <b>2</b> |
| 2.1      | Colocando um trecho ( <i>chunk</i> ) de código . . . . . | 3        |
| <b>3</b> | <b>Mudando o destino da geração</b>                      | <b>3</b> |

## 1 O que é um arquivo Rmd

Um arquivo `Rmd` é escrito em um formato que permite a geração de documentos de diferentes tipos, como `HTML` (*Hypertext Markup Language*, hipertexto diretamente publicável na internet), `PDF` (*Portable Document Format*) ou `docx` (formato Word).

Um arquivo `Rmd` é composto por diferentes peças que são interpretadas quando da montagem (ou costura) do documento. Ele inicia com um cabeçalho no formato `YAML` (*YAML Ain't Markup Language*). Neste cabeçalho é descrito em que formato o documento será gerado (`docx`, `PDF`, `HTML`) e as peculiaridades de geração em cada um desses formatos.

todo cabeçalho `YAML` precisa conter ao menos um título, podendo conter também um autor e uma data. Ele necessariamente inicia e termina com `---`. Na nossa disciplina estamos trabalhando com documentos `HTML`, então usaremos

```
---
title: "Estrutura de um arquivo Rmd"
author: "Prof. Frega"
date: "25-03-2024"
output:
  html_document:
---
```

Se desejamos seções numeradas, podemos usar

```
---
title: "Estrutura de um arquivo Rmd"
author: "Prof. Frega"
date: "25-03-2024"
output:
  html_document:
    number_sections: yes
---
```

Existem alguns estilos pré-fabricados que deixam o documento com uma formatação mais bonita, são eles: *default*, *bootstrap*, *cerulean*, *cosmo*, *darkly*, *flatly*, *journal*, *lumen*, *paper*, *readable*, *sandstone*, *simplex*, *spacelab*,

*united*, e *yeti*.

```
---
title: "Estrutura de um arquivo Rmd"
author: "Prof. Frega"
date: "25-03-2024"
output:
  html_document:
    number_sections: yes
    theme: yeti
---
```

Para arquivos no formato HTML, podemos inserir algumas formatações adicionais por meio de um arquivo de estilo, que tem o formato CSS (*Cascade Style Sheet*). Vamos criar um arquivo chamado **EstruturaRmd.css** e chamá-lo no cabeçalho YAML como

```
---
title: "Estrutura de um arquivo Rmd"
author: "Prof. Frega"
date: "25-03-2024"
output:
  html_document:
    number_sections: yes
    theme: cerulean
    css: EstruturaRmd.css
---
```

O arquivo **EstruturaRms.css** foi gerado com as seguintes implementações **css**

```
/*
Estes comandos fazem os textos dos parágrafos serem
escritos na cor azul e terem a sua fonte aumentada em 10%.
O texto será justificado às margens
*/

p {
  color: blue;
  font-size: 110%;
  text-align: justify;
}

/*
Aqui mandamos criar um espaço de 10 cm ao final do corpo de texto
e uma cor de fundo amarelo clara para melhorar a legibilidade
*/

body {
  margin-bottom: 10cm;
  background-color: lightyellow;
}
```

## 2 Inserindo código e texto

Tudo o que se escrever no arquivo Rmd é um texto, a menos que seja identificado por um conjunto de caracteres especiais. Por exemplo

```
# Identificador de capítulo
criará um capítulo no texto. Similarmente,

## identificador de seção
criará uma seção dentro do capítulo.
```

## 2.1 Colocando um trecho (*chunk*) de código

Um trecho de código é identificado com o formato

```
# Aqui serão inseridos os comandos
2+2
```

```
## [1] 4
```

Podem ser inseridos comandos em mais de uma linguagem, por exemplo, Python

```
# para executar isto aqui é necessário ter o Python instalado em sua máquina
2+2
```

```
## 4
```

## 3 Mudando o destino da geração

Simplesmente alterando a ordem do trecho YAML podemos mudar a geração do documento para qualquer dos três tipos mencionados no início deste texto.

```
---
title: "Estrutura de um arquivo Rmd"
author: "Prof. Frega"
date: "25-03-2024"
output:
  html_document:
    toc: yes
    number_sections: yes
    theme: cerulean
    css: EstruturaRmd.css
  pdf_document:
    toc: yes
    number_sections: yes
  word_document:
    toc: yes
---
```

Usando esse mesmo cabeçalho YAML foram gerados os três documentos que seguem anexos.