



rich_

Texto enriquecido no terminal

Live de Python # 224



1. Texto enriquecido

Uma breve explicação do termo e exemplos práticos no terminal

2. Rich

Instalação e entendimento básico da biblioteca

3. API do Console

O core da coisa toda!

4. Coisas pra mostrar no console

Traceback, Markdown, Arvores e Tabelas

5. Desenhista de shell!

Entendendo o básico sobre displays e layouts



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Ademar Peixoto, Adilson Herculano, Adriana Cavalcanti, Adriano Ferraz, Alexandre Harano, Alexandre Lima, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Andre Azevedo, André Rafael, Aquiles Coutinho, Arnaldo Turque, Aurelio Costa, Bruno Batista, Bruno Divino, Bruno Freitas, Bruno Guizi, Bruno Lopes, Bruno Ramos, Caio Felix, Caio Nascimento, Carina Pereira, Christiano Moraes, Clara Battesini, Dandara Sousa, Daniel Freitas, Daniel Haas, Daniel Santos, Danilo Segura, David Couto, David Kwast, Delton Porfiro, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Diogo Paschoal, Douglas Bastos, Douglas Zickuhr, Eduardo Tolmasquim, Elton Silva, Emerson Rafael, Eneas Teles, Erick Ritir, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Everton Silva, Fabiano Tomita, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Fabricio Araujo, Felipe Rodrigues, Fernanda Prado, Fernando Florêncio, Firehouse Firehouse, Flávio Meira, Flavkaze Flavkaze, Gabriel Barbosa, Gabriel Mizuno, Gabriel Nascimento, Gabriel Simonetto, Geandreson Costa, Guilherme Cabrera, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Guilherme Silva, Gustavo Dettenborn, Gustavo Pereira, Gustavo Suto, Harold Gautschi, Heitor Fernandes, Henrique Junqueira, Hugo Cosme, Igor Taconi, Ismael Ventura, Italo Silva, Jairo Jesus, Jairo Lenfers, Janael Pinheiro, João Paulo, João Rodrigues, Joelson Sartori, Johnny Tardin, Jônatas Silva, José Gomes, Joseíto Júnior, Jose Mazolini, José Pedro, Juan Gutierrez, Juliana Machado, Júlio Gazeta, Júlio Pereira, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Miranda, Lengo Lengo, Leonardo Mello, Leonardo Nazareth, Leon Solon, L. Perciliano, Luancomputacao Roger, Luã Vacaro, Lucas Adorno, Lucas Carderelli, Lucas Dias, Lucas Mello, Lucas Mendes, Lucas Nascimento, Lucas Schneider, Lucas Simon, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Maicon Pantoja, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marcio Moises, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mateus Lisboa, Matheus Cortezi, Matheus Oliveira, Matheus Silva, Matheus Vian, Mauricio Fagundes, Mauricio Nunes, Mírian Batista, Mlevi Lsantos, Murilo Andrade, Murilocunha Murilocunha, Murilo Viana, Natan Cervinski, Nathan Branco, Nicolas Teodosio, Osvaldo Neto, Otávio Carneiro, Patricia Minamizawa, Patrick Felipe, Paulo D., Paulo Tadei, Pedro Henrique, Pedro Pereira, Pedro Silva, Peterson Santos, P Muniz, Priscila Santos, Rafael Lopes, Rafael Romão, Ramayana Menezes, Regis Santos, Rene Bastos, Ricarte Jr, Riverfount, Rjribeiro Rjribeiro, Robson Maciel, Rodrigo Alves, Rodrigo Cardoso, Rodrigo Freire, Rodrigo Quiles, Rodrigo Ribeiro, Rodrigo Vaccari, Rodrigo Vieira, Rogério Lima, Rogério Nogueira, Rogério Sousa, Ronaldo Silva, Ronaldo Silveira, Rui Jr, Samanta Cicilia, Sebastião Tolentino, Tay Turnner, Thaynara Pinto, Thiago Araujo, Thiago Borges, Thiago Curvelo, Thiago Moraes, Thiago Salgado, Thiago Souza, Tiago Minuzzi, Tony Dias, Valcilon Silva, Valdir Tegon, Vcwild Vcwild, Vinícius Bastos, Vinicius Stein, Vitor Luz, Vladimir Lemos, Walter Reis, Wesley Mendes, Willian Lopes, Wilson Duarte, Wilson Neto, Wilson Rocha, Xico Silvério, Yury Barros



Obrigado você



Uma aulinha de
história

rich
text

Textos enriquecidos



teste.rtf - LibreOffice Writer

Arquivo Editar Exibir Inserir Formatar Estilos Tabela Formulário Ferramentas Janela Ajuda

Conteúdo da tabela Ubuntu 21 pt N I S - S A A Z T - A -

Batatinhas fritas

Olá, esse texto é sobre batatas fritas.

BATATAS FRITAS

| Tipo | Nota | Observação |
|--------|------|-------------------------|
| Assada | 5 | Bom, mas frita é melhor |
| Cozida | 5 | Bom, mas frita é melhor |
| Purê | 7 | Bom, mas frita é melhor |

Nada melhor do que **FRITA**, batata só é boa se for **FRITA**

Página 1 de 1 47 palavras, 230 caracteres Estilo de página Padrão Português (Brasil) Tabela1:C3 100%

formato .rtf (rich text format)



```
1 \formshade\paperh15840\paperw12240\margl1134\mgr1134\margt1134\margb1134\sectd\sbknone\pgndec\sftnnar\saftnnrlc
  \sectunlocked1\pgwsxn12240\pghsxn15840\marglsxn1134\mgrgsxn1134\margtsxn1134\mgrbsxn1134\ftnbj\ftnstart1
  \ftnrstcont\ftnnar\aeenddoc\aftnrstcont\aftnstart1\aftnnrlc
2 {\*\ftnsep\chftnsep}\pgndec\pard\plain \s0\rtlch\af8\afs24\alang1081 \ltrch\lang1046\langfe2052\hich\af3
  \loch\widctlpar\hyphpar0\ltrpar\cf0\fs24\lang1046\kerning1\dbch\af10\langfe2052\qc\ltrpar{\rtlch\afs96 \ltrch
  \hich\af6\loch\fs96\fs6\loch
3 Batatinhas fritas}
4 \par \pard\plain \s0\rtlch\af8\afs24\alang1081 \ltrch\lang1046\langfe2052\hich\af3\loch\widctlpar\hyphpar0\ltrpar
  \cf0\fs24\lang1046\kerning1\dbch\af10\langfe2052\qc\ltrpar\rtlch\afs42 \ltrch\hich\af4\loch\fs42\fs4\loch
5
6 \par \pard\plain \s0\rtlch\af8\afs24\alang1081 \ltrch\lang1046\langfe2052\hich\af3\loch\widctlpar\hyphpar0\ltrpar
  \cf0\fs24\lang1046\kerning1\dbch\af10\langfe2052\ql\ltrpar{\rtlch\afs42 \ltrch\hich\af6\loch\fs42\fs6\loch
7 0\u225'\3f, esse texto \u233'\3f sobre batatas fritas.}
8 \par \pard\plain \s0\rtlch\af8\afs24\alang1081 \ltrch\lang1046\langfe2052\hich\af3\loch\widctlpar\hyphpar0\ltrpar
  \cf0\fs24\lang1046\kerning1\dbch\af10\langfe2052\ql\ltrpar\rtlch\afs42 \ltrch\hich\af6\loch\fs42\fs6\loch
```

RTF



RTF é um formato criado pela Microsoft em 1987 para interoperabilidade de textos "estilizados".

A ideia é que o formato seja em texto "puro", qualquer editor de texto possa abrir. Mesmo que não seja um editor capaz de reproduzir o estilo.

Isso deu base para MUITOS outros formatos. Mas o nome **rich** permaneceu na história.

Formatos de textos ricos



Basicamente qualquer linguagem de "marcação" acaba tomando esse rótulo de textos ricos/enriquecidos

- HTML
- Markdown
- odt
- tex
- BBCode
- rtf
- entre outros...

A ideia principal



Escrever tags que representem uma forma visual em forma de texto.

Exemplo de **negrito**:

- Markdown: ****negrito****
- Tex: **\textbf{negrito}**
- HTML: ** negrito **
- BBCode: **[bold] negrito [/bold]**

BBCode

O rich tem como inspiração o bbcode

| Example in HTML/CSS | BBCode | Output |
|---|--|-------------------------------|
| <code>bolded text</code> ^[2] , <code>bolded text</code> ^[1] or <code>bolded text</code> | <code>[b]bolded text[/b]</code> | bolded text |
| <code><i>italicized text</i></code> , <code>italicized text</code> or <code>italicized text</code> | <code>[i]italicized text[/i]</code> | <i>italicized text</i> |
| <code><ins>underlined text</ins></code> or <code>underlined text</code> | <code>[u]underlined text[/u]</code> | <u>underlined text</u> |
| <code>strikethrough text</code> or <code>strikethrough text</code> | <code>[s]strikethrough text[/s]</code> | strikethrough text |

Rich

estilizando tudo!



Rich é uma biblioteca para exibir textos "ricos" e formatados no terminal.

- Versão 12.6.0
- Primeira versão em 2019 (0.1.0)
- MIT
- Criado por Will McGugan (@willmcgugan)
- www.textualize.io

Rich virou base para diversos projetos incríveis:

- pip
- pytest
- typer (opcional)
- Textualize

rich também conta com um cli (rich-cli)

-bash

> python -m rich

Rich features

Colors

✓ 4-bit color

✓ 8-bit color

✓ Truecolor (16.7 million)

✓ Dumb terminals

✓ Automatic color conversion



Styles

All ansi styles: **bold**, *dim*, *italic*, underline, ~~strikethrough~~, reverse, and even blink.

Text

Word wrap text. Justify *left*, *center*, *right* or *full*.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque in metus sed sapien ultricies pretium a at justo. Maecenas luctus velit et auctor maximus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque in metus sed sapien ultricies pretium a at justo. Maecenas luctus velit et auctor maximus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque in metus sed sapien ultricies pretium a at justo. Maecenas luctus velit et auctor maximus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque in metus sed sapien ultricies pretium a at justo. Maecenas luctus velit et auctor maximus.

Asian language support

🇨🇳 该库支持中文，日文和韩文本！

🇯🇵 ライブラリは中国語、日本語、韓国語のテキストをサポートしています

🇰🇷 이 라이브러리는 중국어, 일본어 및 한국어 텍스트를 지원합니다

Markup

Rich supports a simple *bbcode* like markup for *color*, *style*, and emoji! 🍌🍏🐼🐶🍌🚗

Tables

| Date | Title | Production Budget | Box Office |
|--------------|-------------------------------------|-------------------|-----------------|
| Dec 20, 2019 | Star Wars: The Rise of Skywalker | \$275,000,000 | \$375,126,118 |
| May 25, 2018 | Solo: A Star Wars Story | \$275,000,000 | \$393,151,347 |
| Dec 15, 2017 | Star Wars Ep. VIII: The Last Jedi | \$262,000,000 | \$1,332,539,889 |
| May 19, 1999 | Star Wars Ep. I: The phantom Menace | \$115,000,000 | \$1,027,044,677 |

Syntax highlighting & pretty printing

```
1 def iter_last(values: Iterable[T]) -> Ite {
2     """Iterate and generate a tuple with
3     iter_values = iter(values)
4     try:
5         previous_value = next(iter_values)
6     except StopIteration:
7         return
8     for value in iter_values:
9         yield False, previous_value
10        previous_value = value
11    yield True, previous_value
}
```

```
'foo': [
    3.1427,
    (
        'Paul Atreides',
        'Vladimir Harkonnen',
        'Thufir Hawat'
    )
],
'atomic': (False, True, None)
```

Markdown

Markdown

Supports much of the **markdown**, *__syntax__*!

- Headers

- Basic formatting: ****bold****, **italic**, ``code``

- Block quotes

- Lists, and more...

Supports much of the *markdown*, *syntax*!

• Headers

• Basic formatting: ***bold***, *italic*, `code`

• Block quotes

• Lists, and more...

+more! Progress bars, columns, styled logging handler, tracebacks, etc...

`pip install rich`



Instalação



Exemplos na linha de comando



Abra seu terminal e tente um por um!

```
python -m rich
```

```
python -m rich.table
```

```
python -m rich.emoji
```

```
python -m rich.status
```

```
python -m rich.text
```

```
python -m rich.box
```

```
python -m rich.panel
```

```
python -m rich.layout
```


pip install rich

pip install rich-cli



Instalação



Rich-cli (mostrando o que é possível)



Abra seu terminal e tente um por um!

```
rich exemplos_cli/exemplo.json
```

```
rich exemplos_cli/exemplo.csv
```

```
rich exemplos_cli/exemplo.py
```

```
rich -g exemplos_cli/exemplo.py
```

```
rich -g -n exemplos_cli/exemplo.py
```

```
rich -g -n --theme monokai exemplos_cli/exemplo.py
```

rich.print



O rich conta com uma função de **print** que recebe os mesmos parâmetros da função de print do python. Porém, com super poderes de riqueza xD

```
1  # exemplo_00.py
2  import rich
3
4  l = [1, 2, 3, 4]
5  d = {'nome': 'Eduard', 'idade': 29, 'hobbies': ['Fazer Lives']}
6
7  print(l)
8  print(d)
9
10 rich.print(l)
11 rich.print(d)
```

Formatadores



| Símbolos | Abreviado | Uso | Resultado |
|-----------|-----------|-------------------|----------------|
| bold | b | [b] ola [/b] | ola |
| underline | u | [u] ola [/u] | <u>ola</u> |
| italico | i | [i] ola [/i] | <i>ola</i> |
| strike | s | [s] ola [/s] | ola |
| reverse | r | [r] ola [/r] | ola |
| conceal | | [conceal] ola [/] | |
| blink | | [blink] ola [/] | ola |

Formatadores









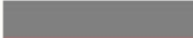





```
1  # exemplo_01.py
2  from rich import print
3
4  texto = '''
5  [i]Boas [s]vindas[/i][[/s] [d]a[/d] [b]Live de [u]Python[/b][[/u]!
6
7  [r]Live [blink]Rich[/blink][[/]
8  '''
9
10 print(texto)
```

Boas vindas a **Live de Python**!

Live Rich

Cores



| Color | Number | Name | Hex | RGB |
|---|--------|------------------|-----|-----|
|  | 0 | "black" | | |
|  | 1 | "red" | | |
|  | 2 | "green" | | |
|  | 3 | "yellow" | | |
|  | 4 | "blue" | | |
|  | 5 | "magenta" | | |
|  | 6 | "cyan" | | |
|  | 7 | "white" | | |
|  | 8 | "bright_black" | | |
|  | 9 | "bright_red" | | |
|  | 10 | "bright_green" | | |
|  | 11 | "bright_yellow" | | |
|  | 12 | "bright_blue" | | |
|  | 13 | "bright_magenta" | | |
|  | 14 | "bright_cyan" | | |
|  | 15 | "bright_white" | | |

<https://rich.readthedocs.io/en/latest/appendix/colors.html>

Usando as cores

```
1  # exemplo_02.py
2  from rich import print
3
4  texto = '''
5  [yellow on red]Boas vindas[/] a [green]Live[/] de [red][u][b]Python[/u][b][/]!
6
7  [bold cyan r]Live Rich[/]
8  '''
9
10 print(texto)
```

o → python exemplo_02.py

Boas vindas a Live de Python!

Live Rich

Emojis

```
1  # exemplo_03.py
2  from rich import print
3
4  texto1 = ':x: :heart: :warning: :snake: :woman:'
5  texto2 = ':cool: :taxi: :file_folder:'
6
7  print(texto1) # ❌ ❤️ ⚠️ 🐍 👩
8  print(texto2) # 🧊 🚕 📁
```


Tudo Junto

```
1  # exemplo_04.py
2  from rich import print
3
4  texto = '''
5  :snake: [i]Boas vindas[/i] a [red b]Live de [u]Python[/u][/]! :snake:
6
7  :clock10: Todas as segundas às [r]22h[/] :clock10:
8
9  :fireworks: [d]Se increva no canal[/] :fireworks:
10
11  :thumbsup: [blink r]Deixa um joinha[/] :thumbsup:
12  '''
13
14  print(texto)
```

Tudo Junto

```
1  # exemplo_04.py
2  from rich import print
3
4  texto = '''
5  :snake: [i]Boas vindas[/i] a [red b]Live de [u]Python[/u][/]! :snake:
6
7  :clock10: Todas as segundas às [r]22h[/] :clock10:
8
9  :fireworks: [d]Se increva no canal[/] :fireworks:
10
11  :thumbsup: [blink r]Deixa um joinha[/]
12  '''
13
14  print(texto)
```

o → python exemplo_04.py

🐍 Boas vindas a **Live de Python!** 🐍

⬇ Todas as segundas às **22h** ⬇

🎆 Se increva no canal 🎆

👍 **Deixa um joinha** 👍

Agora tudo fica
mais legal

Console

Console



Diferente da função de print. O console oferece uma API mais robusta para podermos fazer diversas coisas diferentes. Ter o controle total da área do shell. Não somente uma linha por vez.

```
1  # exemplo_05.py
2  from rich.console import Console
3
4  c = Console()
5
6  # Formas de saída
7  c.print('[b red]Live[/]')
8  c.log('[b red]Live[/]')
9  c.out('[b red]Live[/]')
10 c.print_json('[false, true, null, "foo"]')
```

Alinhamento e estilo



```
1  # exemplo_05.py
2  from rich.console import Console
3
4  c = Console()
5
6  # Alinhamento e estilo
7  c.print(t, justify='center', style='red on white')
8  c.print(t, justify='left', style='bold')
9  c.print(t, justify='right', style='u b')
```

Rule / Input / Status

```
1  # exemplo_05.py
2  from rich.console import Console
3
4  c = Console()
5
6  # Rule
7  c.rule('Boas vindas :warning:')
8
9  # Inputs
10 c.input('Qual seu nome [b]mesmo[/]? ')
11
12 # Status
13 with c.status(
14     'Esperando o lado certo da lua...',
15     spinner='moon'
16 ):
17     sleep(3)
```

Export



```
1  # exemplo_05.py
2  from rich.console import Console
3
4  c = Console(record=True)
5
6  c.rule('Boas vindas :warning:')
7  c.print('[bold red reverse] Um teste[/]')
8  c.rule('Adeus :atm:')
9
10 c.save_html('resultado.html')
```

Conso
le +1

Explorando o poder!

Traceback



Melhores mensagens de erro!

```
1  # exemplo_06.py
2  from rich.console import Console
3  console = Console()
4
5  try:
6      1 + 'a'
7  except Exception:
8      console.print_exception(show_locals=True)
```

Traceback 2



O rich consegue melhorar todas as mensagens de erro do terminal!
Mesmo sem ser invocado diretamente

```
1  # exemplo_06.py
2  from rich.traceback import install
3  install(show_locals=True)
4
5  1 + 'a'
```

Markdown

Rich consegue representar de forma rica textos em markdown

```
1  # exemplo_07.py
2  from rich.console import Console
3  from rich.markdown import Markdown
4  console = Console()
5
6  texto = '''
7  # Título do texto
8
9  Boas memórias da live de Python
10
11  ## Passos a seguir
12
13  1. **Curta** o vídeo
14  2. *Se inscreva no canal*'''
15
16  console.print(Markdown(texto))
```

Markdown

Rich consegue representar de forma rica textos em markdown

```
1  # exemplo_07.py
2  from rich.console import Console
3  from rich.markdown import Markdown
4  console = Console()
5
6  texto = ''
7  # Título do texto
```

o → python exemplo_07.py

Título do texto

Boas memórias da live de Python

Passos a seguir

- 1 **Curta** o vídeo
- 2 *Se inscreva no canal*

```
10 console.print(Markdown(texto))
```

Trees (Arvores)



```
1 # exemplo_08.py
2 from rich.console import Console
3 from rich.tree import Tree
4
5 c = Console()
6 t = Tree(label=':red_circle: - Arvore')
7
8 g1 = t.add('Galho 1')
9 g2 = t.add('Galho 2')
10 g3 = t.add('Galho 3')
11
12 g1.add('1')
13 g2.add('4')
14 g3.add('6')
15
16 c.print(t)
```

Trees (Arvores)



```
1 # exemplo_08.py
2 from rich.console import Console
3 from rich.tree import Tree
4
5 c = Console()
6 t = Tree(label=':red_circle: - Arvore')
7
8 g1 = t.add('Galho 1')
9 g2 = t.add('Galho 2')
10 g3 = t.add('Galho 3')
11
12 g1.add('1')
13 g2.add('4')
14 g3.add('6')
15
16 c.print(t)
```

```
python exemplo_08.py
- Arvore
├── Galho 1
│   ├── 1
│   ├── 2
│   └── 3
├── Galho 2
│   └── 4
└── Galho 3
    ├── 5
    └── 6
```

Tabelas

```
1  from rich.console import Console
2  from rich.table import Table
3
4  table = Table(title="Star Wars Movies")
5
6  table.add_column("Released", justify="right", style="cyan", no_wrap=True)
7  table.add_column("Title", style="magenta")
8  table.add_column("Box Office", justify="right", style="green")
9
10 table.add_row("Dec 20, 2019", "Star Wars: The Rise of Skywalker", "$952,110,690")
11 table.add_row("May 25, 2018", "Solo: A Star Wars Story", "$393,151,347")
12 table.add_row("Dec 15, 2017", "Star Wars Ep. V111: The Last Jedi", "$1,332,539,889")
13 table.add_row("Dec 16, 2016", "Rogue One: A Star Wars Story", "$1,332,439,889")
14
15 console = Console()
16 console.print(table)
```

Tabelas

```
1 from rich.console import Console
2 from rich.table import Table
3
4 table = Table(title="Star Wars Movies")
5
6 table.add_column("Released", justify="right", style="cyan", no_wrap=True)
7 table.add_column("Title", style="magenta")
8 table.add_column("Box Office", justify="right", style="green")
9
```

Star Wars Movies

| Released | Title | Box Office |
|--------------|-----------------------------------|-----------------|
| Dec 20, 2019 | Star Wars: The Rise of Skywalker | \$952,110,690 |
| May 25, 2018 | Solo: A Star Wars Story | \$393,151,347 |
| Dec 15, 2017 | Star Wars Ep. V111: The Last Jedi | \$1,332,539,889 |
| Dec 16, 2016 | Rogue One: A Star Wars Story | \$1,332,439,889 |

Desen
hando

Artista de shell

Grupos



Grupos fazem os elementos serem renderizados de uma vez só no console

```
1  from rich.console import Console, Group
2  console = Console()
3  # ...
4  g = Group(
5      '[red on yellow b] Batatinhas[/]',
6      '[r blink] fritas[/]',
7      table,
8  )
9
10 console.print(g)
```

Painéis



Painéis são formas de desenhar caixas no terminal

```
1  # exemplo_10.py
2  from rich.panel import Panel
3  from rich.console import Console, Group
4  from rich import box
5
6  c = Console()
7
8  g = Group(
9      Panel('Live', border_style='red on white'),
10     Panel('de', box=box.ASCII2),
11     Panel('[red]Python![/]')
12 )
13
14 c.print(g)
```

Painéis



Painéis são formas de desenhar caixas no terminal

```
1 # exemplo_10.py
2 from rich.panel import Panel
3 from rich.console import Console, Group
4 from rich import box
5
6 c = Console()
7
8 g = Group(
9     Panel('Live', border_
10     Panel('de', box=box.A
11     Panel('[red]Python![/
12 )
13
14 c.print(g)
```

python exemplo_10.py

Live

de

Python!

Text



Formatando o texto dentro de outros elementos

```
1  # exemplo_11.py
2  from rich import print
3  from rich.text import Text
4  from rich.panel import Panel
5
6  text = Text('Batatinha')
7  text.justify = 'center'
8  text.stylize('red b u reverse')
9
10 print(Panel(text))
```

Text



```
1  # exemplo_11.py
2  from rich import print
3  from rich.text import Text
4  from rich.panel import Panel
5
6  text = Text('Batatinha')
7  text.justify = 'center'
8  text.stylize('red b u reverse')
9
10 print(Panel(text))
```

```
o → python exemplo_11.py
```

Batatinha

Layouts

A arte de dividir a tela do shell e mais de uma

```
1  # exemplo_12.py
2  from rich import print
3  from rich.layout import Layout
4  from rich.panel import Panel
5
6  l = Layout()
7  l.split_row(
8      Panel('Esquerdo'),
9      Panel('Centro'),
10     Panel('Direito'),
11 )
12
13 print(l)
```

Layouts

A arte de dividir a tela do shell e mais de uma

```
1  # exemplo_12.py
2  from rich import print
3  from rich.layout import Layout
4  from rich.panel import Panel
5
6  l = Layout()
7  l.split_row()
```

➤ `python exemplo_12.py`

Esquerdo

Centro

Direito



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3

