

EASYANSI EM PYTHON //

Cores no terminal de forma simples

Prof. Gustavo Franz (Science/Biology)

Python Developer in Progress

Building Educational Solutions with Python

GitHub: github.com/GustaFranz

Instalação

```
pip install py-easyansi
```

```
# ou a partir do código-fonte:
```

```
pip install -e C:/caminho/para/EasyAnsi
```

Requisitos: Python 3.8+ e terminal com suporte ANSI (Windows 10+ já funciona).

Por que usar EasyAnsi?

EasyAnsi foi criada para quem quer cor no terminal sem aprender uma biblioteca pesada. Você escreve a cor dentro da string, como numa f-string.

EasyAnsi x bibliotecas mais complexas

Necessidade	EasyAnsi	Alternativas pesadas
Começar rápido	<code>from easyansi import eprint</code>	Configurar Console, markup
Colorir uma palavra	<code>//red/{valor}/red</code>	Montar objetos Text
Mensagem de sucesso	<code>success('Pronto')</code>	Criar renderizável
Log colorido	<code>setup_logging()</code>	Handlers, temas, plugins
Tag escrita errada	Vira texto normal	Pode quebrar o programa

Filosofia da biblioteca

- ✓ Menos conceitos, menos imports.
- ✓ Comandos fáceis de lembrar no dia a dia.
- ✓ Fail-safe: URLs e tags desconhecidas não quebram a saída.

Sintaxe `//cor/texto/cor`

Abertura e fechamento

Padrão	Significado	Exemplo
<code>//nome/</code>	Abre tag de estilo	<code>//red/</code>
<code>/nome</code>	Fecha tag explicitamente	<code>/red</code>
<code>//</code>	Fecha última tag aberta	<code>//green/texto//</code>

```
from easyansi import eprint

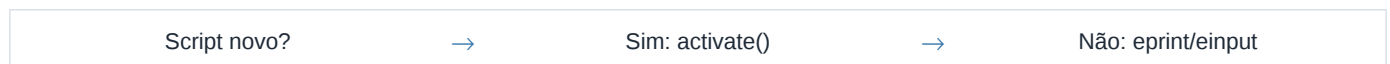
eprint('//green/Olá, mundo!/green')
eprint('Sobrou //red/1/red item no estoque')
eprint('Acesse https://exemplo.com') # URL intacta
```

Fail-safe

- ✓ Somente nomes conhecidos viram cor.
- ✓ `//erro/texto/erro` vira texto puro (não quebra).
- ✓ URLs com `//` permanecem literais.

activate() vs eprint/einput

Fluxo:



Com `activate()`, `print` e `input` normais passam a interpretar as tags EasyAnsi:

```
import easyansi
easyansi.activate()

print('//green/Tudo certo!/green')
nome = input('Digite //cyan/seu nome/cyan: ')
easyansi.deactivate() # restaura originais
```

Alternativa explícita (sem alterar `print` global):

```
from easyansi import eprint, einput, fmt

eprint('//green/Tudo certo!/green')
nome = einput('Digite //cyan/seu nome/cyan: ')
mensagem = fmt('//yellow/aviso/yellow')
```

Atalhos e mensagens de status

Atalhos de cor — ideais para autocomplete no editor:

```
from easyansi import red, green, yellow, blue, bold

print(red('erro'), green('ok'), yellow('aviso'))
print(bold(red('crítico'))) # encadeamento
```

Mensagens prontas para o dia a dia:

```
from easyansi import success, error, warning, info

success('Salvo com sucesso')
error('Falha ao conectar')
warning('Conexão lenta')
info('Rodando na porta 8080')
```

Referência rápida de atalhos

Função	Uso típico
<code>red()</code> , <code>green()</code> , ...	Destacar palavras isoladas
<code>bold()</code> , <code>italic()</code> , ...	Enfatizar títulos e notas
<code>success()</code> , <code>error()</code>	Feedback de operação
<code>style('bold-blue', txt)</code>	Combo personalizado

Títulos, paint e f-strings

```
from easyansi import title, t, paint, ask, activate

activate()
print(title('CADASTRO DE ALUNOS', '='))
print(t('MENU', '~', style='bold-blue'))
print(paint('Texto inteiro azul', 'bold-blue'))
nome = ask('Qual seu nome?', prompt_style='bold-blue')
```

Funciona dentro de f-strings com variáveis:

```
pontos = 42
print(f'Resultado: //green/{pontos}/green pontos')
print(f'Nota //bold-blue/{7.5}/bold-blue em Matemática')
```

Dica sobre input colorido

- ✓ O terminal não colore em tempo real o que você digita.
- ✓ `ask()` colore o prompt; a resposta aparece colorida ao imprimir depois.

Estilos combinados, fundos e hex

Combine estilos com hífen: `bold-blue`, `italic-underline-red`.

```
print('//bold-blue/cabeçalho/bold-blue')
print('//italic-underline-magenta/destaque/')
print('//bg-yellow/texto em fundo amarelo/bg-yellow')
print('//#ff8800/laranja exato/#ff8800')
print('//negrito-vermelho/erro em português/negrito-vermelho')
```

Variações de cor e estilo

Tipo	Tag	Exemplo
Estilo + cor	bold-blue	//bold-blue/texto/bold-blue
Fundo	bg-yellow	//bg-yellow/aviso/bg-yellow
Hex	#ff8800	//#ff8800/laranja/#ff8800
PT	negrito-vermelho	//negrito-vermelho/erro/

Logging colorido e preview()

```
import logging
from easyansi.logging import setup_logging

setup_logging(level=logging.INFO)
logging.info('Servidor iniciado')
logging.warning('Cache desatualizado')
logging.error('Falha ao conectar')
```

Cores por nível de log

Nível	Cor padrão
DEBUG	dim
INFO	cyan
WARNING	yellow
ERROR	red
CRITICAL	bold-red

Para ver todas as cores disponíveis no terminal:

```
import easyansi
easyansi.preview()
```

Comportamento inteligente da saída

Fluxo:

Terminal?	→	Pipe/redir?	→	NO_COLOR?	→	Aplicar cor
Condição	Comportamento					
Terminal interativo (TTY)	Cores ANSI aplicadas					

Condição	Comportamento
Saída em pipe / arquivo	Texto limpo (sem escape)
Variável NO_COLOR	Texto limpo
FORCE_COLOR definido	Cores forçadas

Exercício prático — menu de notas

```
import easyansi
easyansi.activate()

print(easyansi.title('NOTAS DO ALUNO', '='))
print('//cyan/1/cyan Listar //cyan/2/cyan Cadastrar //cyan/0/cyan Sair')
opcao = input('Escolha: ')
if opcao == '1': print('//green/Notas carregadas/green')
elif opcao == '2': print('//yellow/Cadastro em breve/yellow')
else: print('//dim/Encerrando/dim')
```

Cola rápida da API

fmt · eprint · einput · activate · title · t · paint · ask · preview · red/green/bold · success/error/warning/info ·
setup_logging

Prof. Gustavo Franz (Science/Biology)

Python Developer in Progress

Building Educational Solutions with Python

Professor de Ciencias e Biologia desde 2013 — atualmente estudando Programacao.

Eu crio estes resumos para organizar os assuntos e estudar de forma clara e objetiva. Estou compartilhando porque eles tambem podem ajudar outros desenvolvedores que estao começando. Bons estudos — vamos aprender juntos!

GitHub: github.com/GustaFranz

Exercicios Python: github.com/GustaFranz/python_exercises

Para quem quiser ver a resolucao dos meus exercicios em Python.

EasyAnsi: github.com/GustaFranz/easyansi

Para quem quiser codificar personalizando com cores e estilos de forma mais facil, pratica e intuitiva.