



PandaTrix

Pandas : Escolha a Pílula Vermelha da Análise de Dados

Explorando o Poder do Pandas: Principais Comandos, Funções e Métodos

RAFAEL SOARES

PANDATRIX

Explorando o Poder do Pandas: Principais Comandos, Funções e Métodos

Pandas é uma biblioteca poderosa do Python, amplamente utilizada para análise de dados. Vamos explorar os principais comandos, funções e métodos de Pandas com exemplos práticos para facilitar seu aprendizado.

Vamos mostrar as saídas esperadas para cada bloco de código. Neste contexto, suponhamos que temos um arquivo 'dados.csv' com o seguinte conteúdo:

```
nome,idade,departamento
Alice,29,TI
Bob,35,Financeiro
Charlie,22,Marketing
David,40,TI
Eva,NaN,Financeiro
```





01

Carregando Dados com Pandas

FUNÇÃO 'read_csv'

A função 'read_csv' é frequentemente usada para carregar dados de arquivos CSV.

python

```
import pandas as pd

# Carregar dados de um arquivo CSV
df = pd.read_csv('dados.csv')
print(df.head())
```

Saída:

markdown

	nome	idade	departamento
0	Alice	29.0	TI
1	Bob	35.0	Financeiro
2	Charlie	22.0	Marketing
3	David	40.0	TI
4	Eva	NaN	Financeiro



The background features a complex digital aesthetic. The upper portion is filled with vertical streams of binary code (0s and 1s) in white and light blue against a dark blue background. Below this, there are horizontal bands of glowing, cyan-colored, elongated, pill-shaped patterns that resemble data tracks or a stylized representation of a hard drive platters. The overall effect is one of high-tech data visualization.

02

Visualizando os Primeiros Registros

FUNÇÃO 'head'

A função 'head': permite visualizar as primeiras linhas do DataFrame. Por padrão, esta função mostra as primeiras 5 linhas do DataFrame. Caso queira visualizar mais linhas, declare o parâmetro na função. Exemplo: 10 linhas, `df.head(10)`.

python

```
# Mostrar as primeiras 5 linhas do DataFrame  
print(df.head())
```

Saída:

markdown

	nome	idade	departamento
0	Alice	29.0	TI
1	Bob	35.0	Financeiro
2	Charlie	22.0	Marketing
3	David	40.0	TI
4	Eva	NaN	Financeiro





03

Informações do DataFrame

MÉTODO 'info'

O método 'info' fornece um resumo conciso do Dataframe.

python

```
# Obter informações sobre o DataFrame  
print(df.info())
```

Saída:

kotlin

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 5 entries, 0 to 4  
Data columns (total 3 columns):  
#   Column          Non-Null Count  Dtype  
---  -  
0   nome            5 non-null     object  
1   idade           4 non-null     float64  
2   departamento    5 non-null     object  
dtypes: float64(1), object(2)  
memory usage: 248.0+ bytes  
None
```



04

Estatísticas Descritivas

FUNÇÃO 'describe'

A função 'describe' retorna estatísticas descritivas do DataFrame.

python

```
# Estatísticas descritivas do DataFrame  
print(df.describe())
```

Saída:

shell

	idade
count	4.000000
mean	31.500000
std	7.854757
min	22.000000
25%	27.250000
50%	32.000000
75%	36.250000
max	40.000000



The background of the slide is a dark blue/black field filled with vertical lines of glowing green and yellow dots, resembling a digital data stream or binary code. The lines vary in density and brightness, creating a sense of depth and movement.

05

Seleccionando Columnas Específicas

SELECIONANDO COLUNAS

Você pode selecionar colunas específicas do DataFrame facilmente.

python

```
# Selecionar a coluna 'idade'
idades = df['idade']
print(idades.head())
```

Saída:

yaml

```
0      29.0
1      35.0
2      22.0
3      40.0
4       NaN
Name: idade, dtype: float64
```



The background of the slide is a dark blue/black field filled with vertical lines of glowing green and yellow dots, resembling a digital data stream or binary code. The lines vary in density and brightness, creating a sense of depth and movement.

06

Filtrando Dados com Condições

FILTRANDO DADOS

Filtrar dados baseado em condições específicas é simples com Pandas.

python

```
# Filtrar linhas onde a idade é maior que 30
filtro_idade = df[df['idade'] > 30]
print(filtro_idade.head())
```

Saída:

 Copiar código

	nome	idade	departamento
1	Bob	35.0	Financeiro
3	David	40.0	TI





07

Agrupando Dados

FUNÇÃO 'groupby'

A função 'groupby' permite agrupar dados e aplicar funções de agregação.

python

```
# Agrupar dados por 'departamento' e calcular a média das idades
media_idade = df.groupby('departamento')['idade'].mean()
print(media_idade)
```

Saída:

yaml

```
departamento
Financeiro    35.0
Marketing     22.0
TI            34.5
Name: idade, dtype: float64
```





08

Tratando Valores Nulos

MÉTODO 'fillna()'

Pandas oferece várias funções para lidar com valores nulos com o método 'fillna()'.

python

```
# Preencher valores nulos com a média da coluna
df['idade'].fillna(df['idade'].mean(), inplace=True)
print(df.head())
```

Saída:

markdown

	nome	idade	departamento
0	Alice	29.000000	TI
1	Bob	35.000000	Financeiro
2	Charlie	22.000000	Marketing
3	David	40.000000	TI
4	Eva	31.500000	Financeiro



The background is a dark, abstract digital landscape. The upper half features a dense field of vertical lines, each composed of small, glowing circles in shades of blue, green, and orange, resembling a stylized binary code or data stream. The lower half is dominated by a series of bright, glowing blue lines that curve and flow across the frame, creating a sense of dynamic movement and data processing.

09

Ordenando Dados

MÉTODO 'sort_values'

Ordenar dados com Pandas é simples e direto com o método 'sort_values()'.

python

```
# Ordenar DataFrame pela coluna 'idade'
df_sorted = df.sort_values(by='idade')
print(df_sorted.head())
```

Saída:

markdown

	nome	idade	departamento
2	Charlie	22.000000	Marketing
0	Alice	29.000000	TI
4	Eva	31.500000	Financeiro
1	Bob	35.000000	Financeiro
3	David	40.000000	TI



The background is a dark, abstract digital landscape. The upper half features a dense field of vertical lines, each composed of small, glowing circles in shades of blue, green, and yellow, resembling a digital rain or data stream. The lower half shows a series of bright, glowing blue lines that curve and flow across the frame, creating a sense of motion and depth. The overall aesthetic is futuristic and tech-oriented.

10

Aplicando Funções

MÉTODO 'apply'

O método 'apply' permite aplicar funções em colunas ou linhas.

python

```
# Aplicar uma função que converte idades em décadas
df['decada'] = df['idade'].apply(lambda x: (x // 10) * 10)
print(df.head())
```

Saída:

markdown

	nome	idade	departamento	decada
0	Alice	29.000000	TI	20.0
1	Bob	35.000000	Financeiro	30.0
2	Charlie	22.000000	Marketing	20.0
3	David	40.000000	TI	40.0
4	Eva	31.500000	Financeiro	30.0





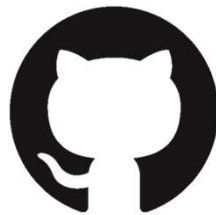
AGRADECIMENTOS

OBRIGADO POR TER ESCOLHIDO A PÍLULA VERMELHA

Esse Ebook foi gerado por IA, e diagramado por humano.
O passo a passo se encontra no meu Github.

•

Esse conteúdo foi gerado com fins didáticos de construção,
não foi realizado uma validação cuidadosa humana no
conteúdo e pode conter erros gerados por uma IA.



<https://github.com/Rafaunderoath/prompts-recipe-to-create-a-ebook>

