Akash • Python & Tech Enthusiast 🚀
@pycode.hubb

by Pycode.Hubb

PANDAS CHEAT SHEET

Data Handling Made Easy

python™

NEXT

Level 1: Creating DataFrames

```python
# Creating a DataFrame from a dictionary
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 35]}
df = pd.DataFrame(data)
print(df)
```

```
      Name  Age
0    Alice   25
1      Bob   30
2  Charlie   35
```

Level 2: Reading and Writing CSV Files

```python
# Reading from and writing to a CSV file
df = pd.read_csv('data.csv')
df.to_csv('output.csv', index=False)
```

Level 3: Viewing Data

```python
# Displaying basic information and data overview
print(df.head())      # First 5 rows
print(df.tail())      # Last 5 rows
print(df.info())      # Data types and non-null counts
```

```
      Name  Age    Major  GPA  Graduated
0     John   20       CS  3.5      False
1     Emma   22     Math  3.8       True
2   Sophia   21  Physics  3.6      False
3  Michael   23       CS  3.2       True
4   Olivia   22     Math  3.9      False
      Name  Age    Major  GPA  Graduated
1     Emma   22     Math  3.8       True
2   Sophia   21  Physics  3.6      False
3  Michael   23       CS  3.2       True
4   Olivia   22     Math  3.9      False
5   Daniel   21  Physics  3.7       True
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Name       6 non-null      object
 1   Age        6 non-null      int64
 2   Major      6 non-null      object
 3   GPA        6 non-null      float64
 4   Graduated  6 non-null      bool
dtypes: bool(1), float64(1), int64(1), object(2)
memory usage: 330.0+ bytes
None
```

Level 4: Selecting Data

```python
# Selecting specific columns and rows
print(df['Name'])          # Single column
print(df[['Name', 'Age']]) # Multiple columns
print(df.loc[0])           # Row by index label
print(df.iloc[0])          # Row by index position
```

```
0       John
1       Emma
2     Sophia
3    Michael
4     Olivia
5     Daniel
Name: Name, dtype: object
      Name  Age
0     John   20
1     Emma   22
2   Sophia   21
3  Michael   23
4   Olivia   22
5   Daniel   21
Name          John
Age             20
Major           CS
GPA            3.5
Graduated    False
Name: 0, dtype: object
Name          John
Age             20
Major           CS
GPA            3.5
Graduated    False
Name: 0, dtype: object
```

Level 5: Filtering Data

```python
# Filtering rows based on a condition
adults = df[df['Age'] > 30]
print(adults)
```

```
Empty DataFrame
Columns: [Name, Age, Major, GPA, Graduated]
Index: []
```

Level 6: Handling Missing Data

```python
# Checking and filling missing values
print(df.isnull())  # Checking for null values
df['Age'] = df['Age'].fillna(df['Age'].mean())
```

```
    Name    Age  Major    GPA  Graduated
0  False  False  False  False      False
1  False  False  False  False      False
2  False  False  False  False      False
3  False  False  False  False      False
4  False  False  False  False      False
5  False  False  False  False      False
```

Level 7: Grouping and Aggregation

```python
# Grouping data and calculating aggregates
grouped = df.groupby('Age').size()
print(grouped)
```

```
Age
20    1
21    2
22    2
23    1
dtype: int64
```

Level 8: Merging and Joining DataFrames

```python
# Merging two DataFrames
df1 = pd.DataFrame({'ID': [1, 2, 3], 'Name': ['Alice', 'Bob', 'Charlie']})
df2 = pd.DataFrame({'ID': [1, 2, 4], 'Score': [85, 90, 95]})

merged = pd.merge(df1, df2, on='ID', how='inner')
print(merged)
```

```
   ID   Name  Score
0   1  Alice     85
1   2    Bob     90
```

Level 9: Applying Functions

```python
# Applying a custom function to a column
df['Age after 5 years'] = df['Age'].apply(lambda x: x + 5)
print(df)
```

```
      Name  Age    Major  GPA  Graduated  Age after 5 years
0     John   20       CS  3.5      False                 25
1     Emma   22     Math  3.8       True                 27
2   Sophia   21  Physics  3.6      False                 26
3  Michael   23       CS  3.2       True                 28
4   Olivia   22     Math  3.9      False                 27
5   Daniel   21  Physics  3.7       True                 26
```

Level 10: Pivot Tables

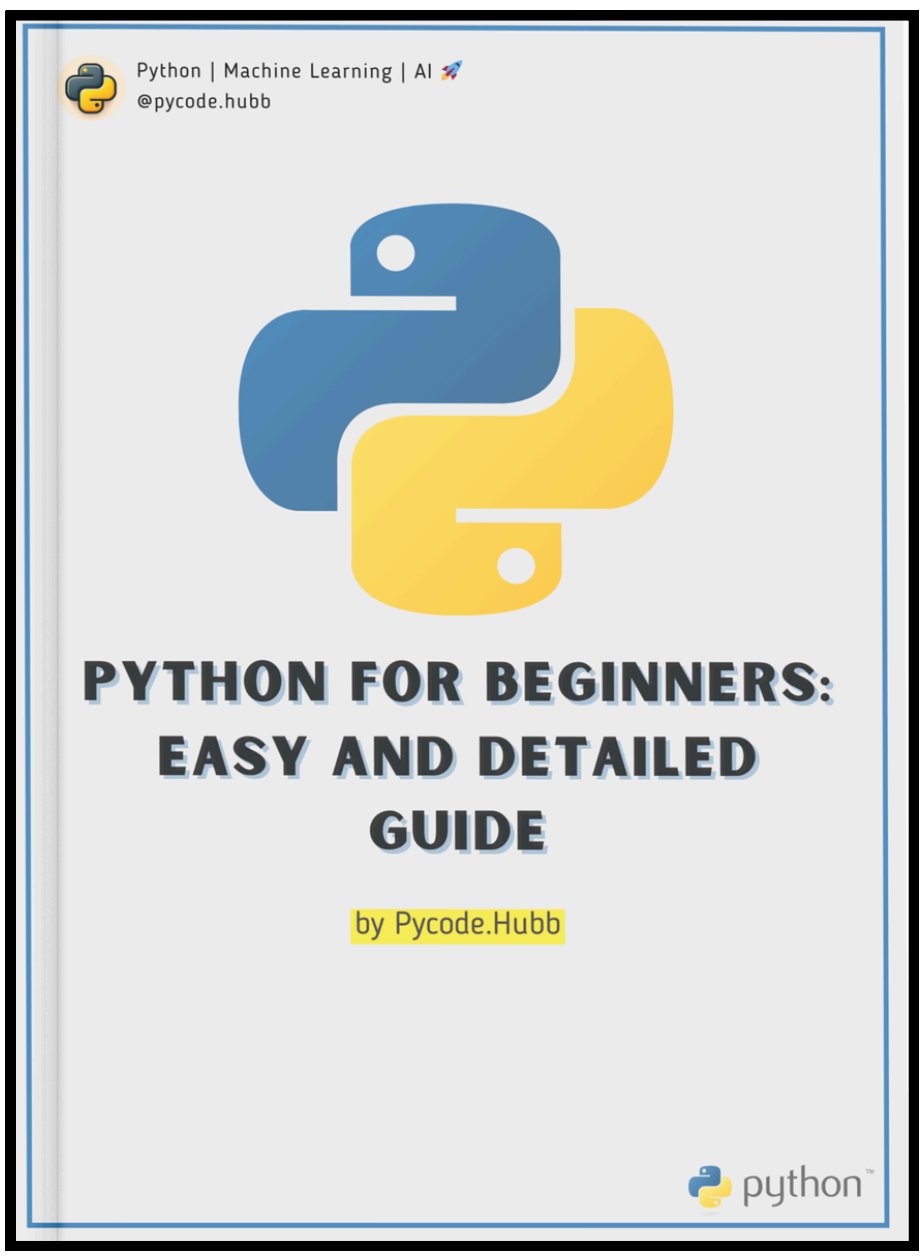
```python
# Creating a pivot table
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Alice', 'Bob'],
        'Sales': [200, 150, 300, 250, 100],
        'Year': [2021, 2021, 2021, 2022, 2022]}
df = pd.DataFrame(data)

pivot = df.pivot_table(values='Sales', index='Name', columns='Year', aggfunc='sum')
print(pivot)
```

```
Year       2021   2022
Name
Alice     200.0  250.0
Bob       150.0  100.0
Charlie   300.0    NaN
```

Akash • Python & Tech Enthusiast 🚀
@pycode.hubb

GRAB OUR ULTIMATE PYTHON BUNDLE!

Master Python from basic to advanced with detailed explanations and practical experience.

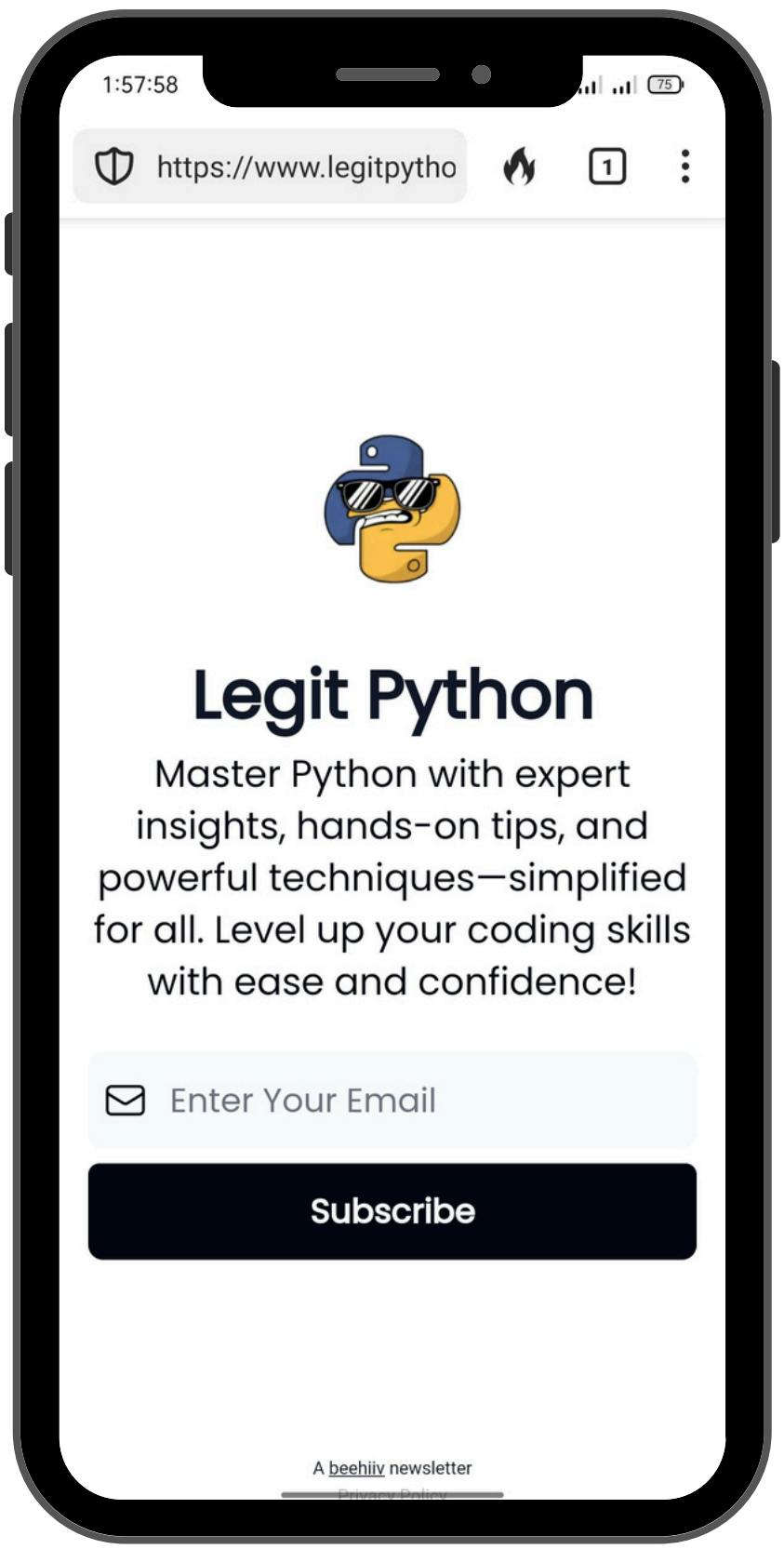


🔗 Link in the Bio 🔗

python™

NEXT

Akash • Python & Tech Enthusiast 🚀
@pycode.hubb



NEWSLETTER

Improve Python in 5 Minutes a Week.

- Stay ahead with our free Python newsletter.
- Discover tips, tools, & insights sent, straight to your inbox every Friday.
- Don't miss out—subscribe today for free.

🔗Link in Bio 🔗

python™