Python Pandas Cheat sheet

DECODING DATA SCIENCE

This covers some of the most commonly used functions and operations in Pandas:

Importing and Exporting Data

Here is a quick Python Pandas cheatsheet that covers some of the most common functions and operations you will use when working with Pandas:

Importing Pandas

To use Pandas, you will first need to import the library:

```
import pandas as pd
```

Reading a CSV file

You can read a CSV file into a Pandas DataFrame using the read_csv function:

```
df = pd.read_csv('filename.csv')
```

Displaying the DataFrame

To view the data in a DataFrame, you can use the head function to display the first few rows:

```
df.head()
```

You can also use the tail function to display the last few rows:

```
df.tail()
```

To display the entire DataFrame, you can simply print it:

```
print(df)
```

Selecting Columns

You can select a single column of a DataFrame by using the [] operator and the column name:

```
df['column_name']
```

You can also select multiple columns by passing a list of column names:

```
df[['column_1', 'column_2']]
```

Filtering Rows

You can filter the rows of a DataFrame using a boolean expression. For example, to select all rows where the value in the 'age' column is greater than 30:

```
df[df['age'] > 30]
```

Sorting Data

You can sort the rows of a DataFrame by one or more columns using the sort_values function. For example, to sort the DataFrame by the 'age' column in ascending order:

```
df.sort_values(by='age')
```

To sort in descending order, set the ascending parameter to False:

```
df.sort_values(by='age', ascending=False)
```

Grouping Data

You can group a DataFrame by one or more columns and apply a function to each group using the groupby function. For example, to group the DataFrame by the 'gender' column and compute the mean of each group:

```
df.groupby('gender').mean()
```

Joining DataFrames

You can join two DataFrames using the merge function. For example, to join two DataFrames on the 'user_id' column:

```
df1.merge(df2, on='user_id')
```

Pivot Tables

You can create a pivot table from a DataFrame using the pivot_table function. For example, to create a pivot table with the 'gender' column as the rows, the 'country' column as the columns, and the 'age' column as the values:

```
df.pivot_table(index='gender', columns='country', values='age')
```

Handling Missing Values

Pandas includes functions for handling missing values. To drop rows with missing values:

```
df.dropna()
```

To fill missing values with a specific value, you can use the fillna function:

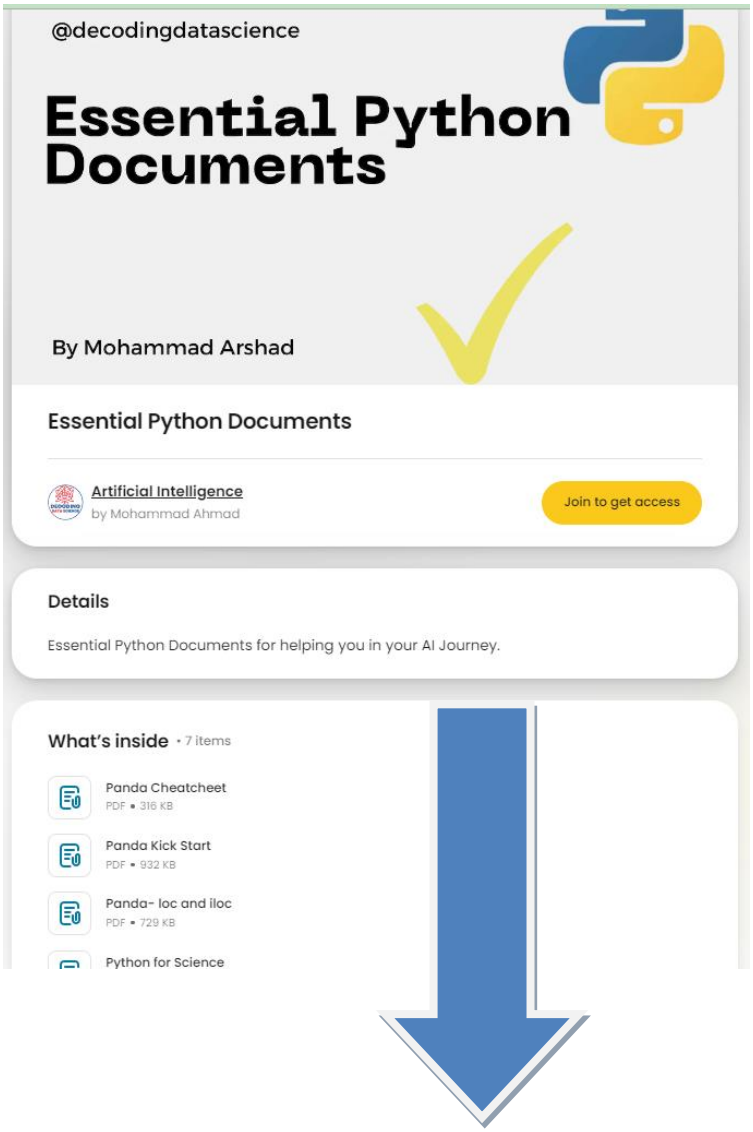
```
df.fillna(value=0)
```

You can also fill missing values with the mean of the column using the fillna function and the mean function:

```
df.fillna(df.mean())
```

Converting Data Types

Get the complete cheat sheet and much more here: https://nas.io/artificial-intelligence-4/hpro



https://nas.io/artificial-intelligence-4/hpro