Certainly! Here's a detailed explanation of how to read data from CSV and TXT files in Python, along with suitable code examples:

Reading data from a CSV file:

1. Import the pandas library:

```
import pandas as pd
```

2. Use the `read_csv()` function of pandas to read the CSV file:

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

3. Optionally, you can specify additional parameters to customize the CSV reading process. For example:

```
df = pd.read_csv('filename.csv', delimiter=',', header=0)
```

4. You can now use the DataFrame `df` to work with the data read from the CSV file. For example, you can print the first few rows using `df.head()` or perform various data manipulations and analysis on the DataFrame.

Here's the **complete code** for reading data from a CSV file:

```
import pandas as pd

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

print(df.head())
```

Reading data from a TXT file:

1. Import the pandas library:

```
import pandas as pd
```

2. Use the `read_csv()` function of pandas to read the TXT file. However, specify the `delimiter` parameter as per the separator used in the TXT file:

```
df = pd.read_csv('filename.txt', delimiter='\t')
```

Replace `'filename.txt'` with the actual path or filename of your TXT file. Also, ensure to set the `delimiter` parameter correctly to match the separator used in your TXT file.

3. Optionally, you can specify additional parameters to customize the TXT reading process, similar to reading CSV files.

4. You can now use the DataFrame `df` to work with the data read from the TXT file.

Here's the **complete code** for reading data from a TXT file:

```
import pandas as pd
```

```
df = pd.read_csv('filename.txt', delimiter='\t')

print(df.head())
```

Ensure that the specified file paths are correct and accessible by the program. The provided code examples will read the file and store the data in a pandas DataFrame, which you can further analyze and manipulate using pandas' powerful functionalities.