```
# pandas is used to load and clean your data.
# LabelEncoder converts text (like country names) into numbers for machine learning.
import pandas as pd
from sklearn.preprocessing import LabelEncoder
```

```
# This loads your file raw_visa_data.csv into a table format (DataFrame).
# df.head() shows the first 5 rows to confirm the data loaded correctly.
df = pd.read_csv("raw_visa_data.csv")
df.head()
```

	application_id	submission_date	decision_date	country	visa_type	age	gender	processing_center
0	1	2022-10-11	2023-02-07	UK	Work	25	Male	USA
1	2	2023-05-07	2023-08-08	UK	Work	30	Female	USA
2	3	2023-08-25	2023-09-07	China	Work	27	Female	USA
3	4	2022-12-08	2023-04-04	UK	Work	23	Female	USA
4	5	2022-10-16	2023-01-28	UK	Work	25	Male	USA

Next steps: Generate code with df | New interactive sheet

```
# info() shows column types (object, int, date).
# isnull().sum() shows how many missing values exist in each column.
df.info()
df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   application_id     400 non-null    int64
 1   submission_date    400 non-null    object
 2   decision_date      400 non-null    object
 3   country            400 non-null    object
 4   visa_type          400 non-null    object
 5   age                400 non-null    int64
 6   gender             400 non-null    object
 7   processing_center  400 non-null    object
dtypes: int64(2), object(6)
memory usage: 25.1+ KB
```

	0
application_id	0
submission_date	0
decision_date	0
country	0
visa_type	0
age	0
gender	0
processing_center	0

dtype: int64

```
# Text columns missing values are replaced with a default "Unknown".
# Age is a number → we fill missing values using the median age.
df['country'] = df['country'].fillna("Unknown")
df['visa_type'] = df['visa_type'].fillna("Unknown")
df['processing_center'] = df['processing_center'].fillna("Unknown")
df['gender'] = df['gender'].fillna("Unknown")
df['age'] = df['age'].fillna(df['age'].median())

print(df.head().to_string())
```

```
   application_id submission_date decision_date country visa_type  age  gender processing_center  processing_days  country_e
0               1      2022-10-11    2023-02-07      UK      Work   25    Male              USA              119
1               2      2023-05-07    2023-08-08      UK      Work   30  Female              USA               93
2               3      2023-08-25    2023-09-07   China      Work   27  Female              USA               13
```

```
3              4      2022-12-08    2023-04-04      UK      Work   23  Female              USA             117
4              5      2022-10-16    2023-01-28      UK      Work   25    Male              USA             104
```

```
# This step converts them into actual date format, so we can calculate time differences.
df['submission_date'] = pd.to_datetime(df['submission_date'])
df['decision_date'] = pd.to_datetime(df['decision_date'])

print(df[['submission_date','decision_date']].head().to_string())
```

```
  submission_date decision_date
0      2022-10-11    2023-02-07
1      2023-05-07    2023-08-08
2      2023-08-25    2023-09-07
3      2022-12-08    2023-04-04
4      2022-10-16    2023-01-28
```

```
# This calculates how many days it took to process each visa application.
df['processing_days'] = (df['decision_date'] - df['submission_date']).dt.days

print(df[['submission_date','decision_date','processing_days']].head().to_string())
```

```
  submission_date decision_date  processing_days
0      2022-10-11    2023-02-07              119
1      2023-05-07    2023-08-08               93
2      2023-08-25    2023-09-07               13
3      2022-12-08    2023-04-04              117
4      2022-10-16    2023-01-28              104
```

```
le = LabelEncoder()

df['country_encoded'] = le.fit_transform(df['country'])
df['visa_type_encoded'] = le.fit_transform(df['visa_type'])
df['gender_encoded'] = le.fit_transform(df['gender'])
df['center_encoded'] = le.fit_transform(df['processing_center'])

print(df[['country', 'country_encoded',
          'visa_type', 'visa_type_encoded',
          'gender', 'gender_encoded',
          'processing_center', 'center_encoded']].head().to_string())
```

```
  country  country_encoded visa_type  visa_type_encoded  gender  gender_encoded processing_center  center_encoded
0      UK                5      Work                  2    Male               1              USA               7
1      UK                5      Work                  2  Female               0              USA               7
2   China                2      Work                  2  Female               0              USA               7
3      UK                5      Work                  2  Female               0              USA               7
4      UK                5      Work                  2    Male               1              USA               7
```

```
#remove undedded column
df_clean = df.drop(['country','visa_type','gender','processing_center','decision_date'], axis=1)
df_clean.head()
```

	application_id	submission_date	age	processing_days	country_encoded	visa_type_encoded	gender_encoded	center_encoded
0	1	2022-10-11	25	119	5	2	1	7
1	2	2023-05-07	30	93	5	2	0	7
2	3	2023-08-25	27	13	2	2	0	7
3	4	2022-12-08	23	117	5	2	0	7
4	5	2022-10-16	25	104	5	2	1	7

Next steps: Generate code with `df_clean` New interactive sheet

```
#displays cleaned dataset
df_clean.head()
```

	application_id	submission_date	age	processing_days	country_encoded	visa_type_encoded	gender_encoded	center_encoded
0	1	2022-10-11	25	119	5	2	1	7
1	2	2023-05-07	30	93	5	2	0	7
2	3	2023-08-25	27	13	2	2	0	7
3	4	2022-12-08	23	117	5	2	0	7
4	5	2022-10-16	25	104	5	2	1	7

Next steps: Generate code with `df_clean` New interactive sheet