This notebook is an exercise in the [Data Cleaning (https://www.kaggle.com/learn/data-cleaning)](https://www.kaggle.com/learn/data-cleaning) course. You can reference the tutorial at [this link (https://www.kaggle.com/alexisbcook/character-encodings)](https://www.kaggle.com/alexisbcook/character-encodings).

In this exercise, you'll apply what you learned in the **Character encodings** tutorial.

Setup

The questions below will give you feedback on your work. Run the following cell to set up the feedback system.

In [1]:
```
from learntools.core import binder
binder.bind(globals())
from learntools.data_cleaning.ex4 import *
print("Setup Complete")
```

```
Setup Complete
```

Get our environment set up

The first thing we'll need to do is load in the libraries we'll be using.

In [2]:
```
# modules we'll use
import pandas as pd
import numpy as np

# helpful character encoding module
import charset_normalizer

# set seed for reproducibility
np.random.seed(0)
```

1) What are encodings?

You're working with a dataset composed of bytes. Run the code cell below to print a sample entry.

In [3]:
```
sample_entry = b'\xa7A\xa6n'
print(sample_entry)
print('data type:', type(sample_entry))
```

```
b'\xa7A\xa6n'
data type: <class 'bytes'>
```

You notice that it doesn't use the standard UTF-8 encoding.

Use the next code cell to create a variable `new_entry` that changes the encoding from `"big5-tw"` to `"utf-8"`. `new_entry` should have the bytes datatype.

In [11]:
```
before = sample_entry.decode("big5-tw")
new_entry = before.encode()

# Check your answer
q1.check()
```

Correct

In [5]:
```
# Lines below will give you a hint or solution code
#q1.hint()
#q1.solution()
```

2) Reading in files with encoding problems

Use the code cell below to read in this file at path `"../input/fatal-police-shootings-in-the-us/PoliceKillingsUS.csv"`.

Figure out what the correct encoding should be and read in the file to a DataFrame `police_killings`.

In [12]:
```
# TODO: Load in the DataFrame correctly.
police_killings = pd.read_csv("../input/fatal-police-shootings-in-the-us/Polic

# Check your answer
q2.check()
```

Correct

Feel free to use any additional code cells for supplemental work. To get credit for finishing this question, you'll need to run `q2.check()` and get a result of **Correct**.

In [7]:
```
# (Optional) Use this code cell for any additional work.
```

In [8]:
```
# Lines below will give you a hint or solution code
#q2.hint()
#q2.solution()
```

3) Saving your files with UTF-8 encoding

Save a version of the police killings dataset to CSV with UTF-8 encoding. Your answer will be marked correct after saving this file.

Note: When using the `to_csv()` method, supply only the name of the file (e.g., `"my_file.csv"`). This saves the file at the filepath `"/kaggle/working/my_file.csv"`.

In [9]:
```
# TODO: Save the police killings dataset to CSV
____

# Check your answer
q3.check()
```

Incorrect: Please save a CSV file and run this code cell again to get credit!

In [10]:
```
# Lines below will give you a hint or solution code
#q3.hint()
#q3.solution()
```

(Optional) More practice

Check out this dataset of files in different character encodings (https://www.kaggle.com/rtatman/character-encoding-examples). Can you read in all the files with their original encodings and them save them out as UTF-8 files?

If you have a file that's in UTF-8 but has just a couple of weird-looking characters in it, you can try out the ftfy module (https://ftfy.readthedocs.io/en/latest/#) and see if it helps.

Keep going

In the final lesson, learn how to **clean up inconsistent text entries** (https://www.kaggle.com/alexisbcook/inconsistent-data-entry) in your dataset.

Have questions or comments? Visit the course discussion forum (https://www.kaggle.com/learn/data-cleaning/discussion) to chat with other learners.