```
# === mapping.ipynb ===
# formatting

#%%
import pandas as pd


#%%
# Mapping lets you format an entire DataFrame
file = "Resources/sample_data.csv"
file_df = pd.read_csv(file)
file_df.head()


#%%
# Use Map to format all the columns
# ------------------------------------------------------------------------
# Format is almost akin to concatenating strings. Whatever is outside of the
# curly brackets is added before/after the initial value which is modified
# by whatever is contained within the curly brackets.
#
# "${:.2f}" converts values into a typical dollar format.  This places a dollar
# sign before the value which has been rounded to two decimal points.
# Likewise "{: ,}" will split a number up so that it uses comma notation.
# For example: the value 2000 would become 2,000 using this format string
#
# Formatting only really works once and - can also change the datatype of a
# column (to string => object) - will return errors if the same code is run
# multiple times without restarting the kernel. This is because, depending on
# what the value is being formatted to - i.e., it's impossible to apply a 2
# floating-point format to a string.
# For this reason apply formatting near the end of an application
# ------------------------------------------------------------------------
file_df["avg_cost"] = file_df["avg_cost"].map("${:.2f}".format)
file_df["population"] = file_df["population"].map("{:,}".format)
file_df["other"] = file_df["other"].map("{:.2f}".format)
file_df.head()


#%%
# Mapping has changed the datatypes of the columns to strings
file_df.dtypes


#%%
```