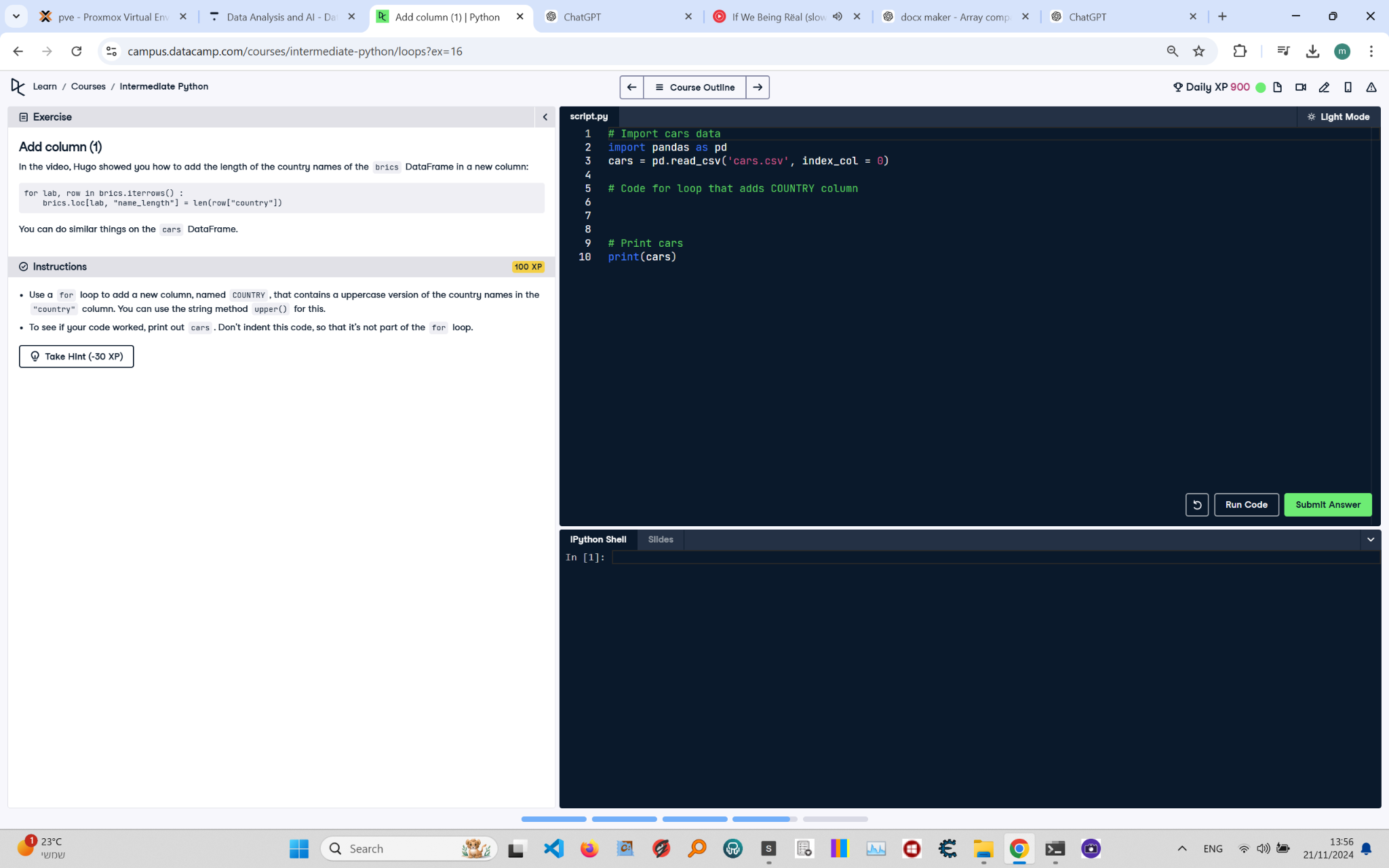
# Add Column to DataFrame in Python



\*\*Question:\*\*

In the video, Hugo showed you how to add the length of the country names of the `brics` DataFrame in a new column:  
  
```python  
for lab, row in brics.iterrows():  
 brics.loc[lab, 'name\_length'] = len(row['country'])  
```  
  
You can do similar things on the `cars` DataFrame.  
  
\*\*Instructions:\*\*  
1. Use a `for` loop to add a new column, named `COUNTRY`, that contains an uppercase version of the country names in the `country` column.  
 You can use the string method `upper()` for this.  
2. To see if your code worked, print out `cars`. Don't indent this code, so that it's not part of the `for` loop.

\*\*Answer:\*\*

Here is the Python code that solves the problem:

# Import pandas  
import pandas as pd  
  
# Load the cars DataFrame  
cars = pd.read\_csv('cars.csv', index\_col=0)  
  
# Add a new column 'COUNTRY' with uppercase country names  
for lab, row in cars.iterrows():  
 cars.loc[lab, 'COUNTRY'] = row['country'].upper()  
  
# Print the updated DataFrame  
print(cars)

\*\*Explanation:\*\*

1. \*\*Importing pandas\*\*: The `pandas` library is imported to handle the DataFrame.  
2. \*\*Loading the DataFrame\*\*: The `cars` DataFrame is loaded from a CSV file (`cars.csv`) with the first column set as the index using `index\_col=0`.  
3. \*\*Adding a new column\*\*:  
 - The `iterrows()` method is used to iterate over the rows of the DataFrame.  
 - For each row, the `upper()` method is applied to the `country` value to create an uppercase version.  
 - The result is stored in a new column named `COUNTRY`.  
4. \*\*Printing the updated DataFrame\*\*: Finally, the updated DataFrame is printed to verify the addition of the new column.