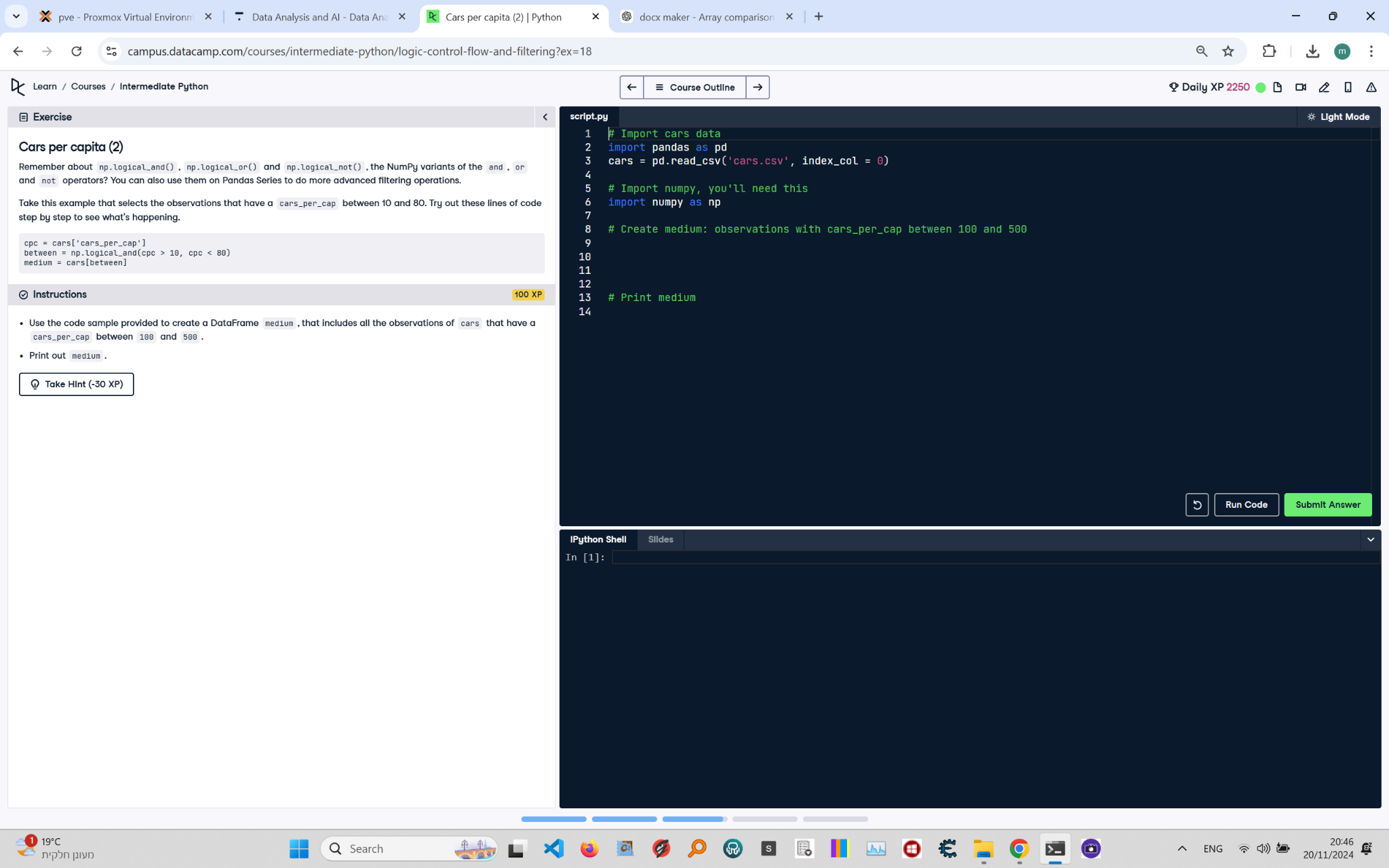
# Cars per Capita (2) in Python



\*\*Question:\*\*

Remember about `np.logical\_and()`, `np.logical\_or()` and `np.logical\_not()`, the NumPy variants of the `and`, `or`, and `not` operators? You can also use them on Pandas Series to do more advanced filtering operations.  
Use the code sample provided to create a DataFrame `medium`, that includes all the observations of `cars` that have a `cars\_per\_cap` between 100 and 500.  
Print out `medium`.

\*\*Answer:\*\*

Here is the Python code that solves the problem:

# Import cars data  
import pandas as pd  
import numpy as np  
  
# Read the dataset  
cars = pd.read\_csv('cars.csv', index\_col=0)  
  
# Create medium: observations with cars\_per\_cap between 100 and 500  
between = np.logical\_and(cars['cars\_per\_cap'] > 100, cars['cars\_per\_cap'] < 500)  
medium = cars[between]  
  
# Print medium  
print(medium)

\*\*Explanation:\*\*

1. \*\*Import libraries\*\*: Both the Pandas and NumPy libraries are imported to handle tabular data and logical operations.  
2. \*\*Read the dataset\*\*: The `pd.read\_csv()` function reads the dataset from a CSV file into a Pandas DataFrame. The `index\_col=0` parameter sets the first column as the index.  
3. \*\*Create boolean Series\*\*: The `np.logical\_and()` function checks if `cars\_per\_cap` is greater than 100 and less than 500 simultaneously. The result is a boolean Series stored in `between`.  
4. \*\*Subset DataFrame\*\*: The `cars[between]` expression uses the boolean Series to filter the rows where `cars\_per\_cap` is between 100 and 500. The result is stored in `medium`.  
5. \*\*Print the result\*\*: The `print(medium)` statement displays the subsetted DataFrame.