- -> notebooks from this lecture: https://github.com/krishnatray/RDP-Reading-Data-with-Python-and-Pandas
- -> parsing data directly from a website
- -> the data needs to be public, so you can parse it
- -> we have a website with a table on it
- ->using NBA tables = html
- -> we are feeding the URL of the site with the data into the Jupyter notebook
- -> in this example, we are importing in the tables - but we can't see all of them
- -> in html the tables are formatted for people to read -> rather than for importing them into Python
- -> there are rows you will need to import into there
    - -> and ones to drop
- -> dr.drop, then giving it a range
- -> html pages are optimised for people not machines
- -> you can use Wikipedia pages to pull the data from
- -> you can also write data to CSV or html
- -> for the read data portion
- -> next <- data wrangling, grouping

- ***-> adding a final source of external data <- an Excel file***
    - -> you can import the Excel data into an ipynb file
    - -> Excel is not a text file
    - -> this requires external tools
    - -> there might be issues when importing from Excel
    - -> you can use the read Excel method for this

- ***-> reading the files***
    - -> we have in this example a products file, with three different sheets
    - -> reading Excel
    - -> we are reading the first sheet of the Excel file
    - -> you can change the way we parse headers
    - -> we can read different sheets of the Excel file

- **-> Excel file class**
    - -> this is another class where we use the Excel class, with the parameter being the file name
    - -> not using Excel to receive the contents of the file
    - -> we can parse certain Excel sheets
    - -> we can pass the product names into Excel
    - -> this works in a similar way to CSV
    - -> converting Excel into CSV
    - -> if the file is shifted -> rows into columns
    - -> we can use also use an Excel writer
    - -> reading and writing data from and to Excel files
    - -> it depends the libaries we have installed in the current environment
    - -> you can also refer to the documentation
    - -> there might be requirements for each of the files in the Pandas database - also depending on the operating system
- ***-> question***

What Python library has the .read_html() method we can use for parsing HTML documents and extracting tables?

- Options
    - BeautifierSoupy
    - WebReader

- HTTP-master
- Pandas <- This one