1. Enter and run this code in a Jupyter cell to produce the dataframe weo:

```
In [9]: import pandas as pd
        data = {'BRA': [13.37, 13.30, 14.34, 15.07, 15.46, 15.98, 16.10],
        'JPN': [33.43, 31.83, 33.71, 34.29, 35.60, 36.79, 37.39],
        'USA': [48.30, 46.91, 48.31, 49.72, 51.41, 52.94, 54.60],
        'Year': [2008, 2009, 2010, 2011, 2012, 2013, 2014]}
        weo = pd.DataFrame(data)
```

```
In [10]: weo
```

Out[10]:

| | BRA | JPN | USA | Year |
|---|---|---|---|---|
| **0** | 13.37 | 33.43 | 48.30 | 2008 |
| **1** | 13.30 | 31.83 | 46.91 | 2009 |
| **2** | 14.34 | 33.71 | 48.31 | 2010 |
| **3** | 15.07 | 34.29 | 49.72 | 2011 |
| **4** | 15.46 | 35.60 | 51.41 | 2012 |
| **5** | 15.98 | 36.79 | 52.94 | 2013 |
| **6** | 16.10 | 37.39 | 54.60 | 2014 |

a. It imported the package "Panda" which is a specific Python package that adds data analysis/structuring functions to Python's fundamental built-in functions, and indicated that we'll refer to it/access it by typing "pd.[TAB]"

**b.**

```
In [11]: whos
Variable   Type         Data/Info
---------------------------------
data       dict         n=4
pd         module       <module 'pandas' from '/a<...>ages/pandas/__init__.py'>
weo        DataFrame          BRA    JPN    USA  Y<...>16.10  37.39  54.60  2014
```

"Data" is a dictionary that defines the data used in "weo"

c. "pd" makes it a panda function

**d. "weo" is a DataFrame made up of the data imported**

**e.**

```
In [15]: Count_Row=weo.shape[0]
         Count_Col=weo.shape[1]
```

```
In [17]: print(Count_Row)
         print(Count_Col)
7
4
```

There are 7 rows and 4 columns

**f.**

```
In [19]: weo.dtypes
Out[19]: BRA     float64
         JPN     float64
         USA     float64
         Year      int64
         dtype: object
```

This describes the variables in each column - those under BRA, JPN, and USA are floats and those under Year are integers

**g.**

In [27]:
```
weo['Year']
```

Out[27]:
```
0    2008
1    2009
2    2010
3    2011
4    2012
5    2013
6    2014
Name: Year, dtype: int64
```

In [28]:
```
type(weo['Year'])
```

Out[28]:
```
pandas.core.series.Series
```

This is a series that referes to the variables in the Year column

In [30]:
```
weo[['Year']]
```

Out[30]:

| | Year |
|---|---|
| **0** | 2008 |
| **1** | 2009 |
| **2** | 2010 |
| **3** | 2011 |
| **4** | 2012 |
| **5** | 2013 |
| **6** | 2014 |

In [26]:
```
type(weo[['Year']])
```

Out[26]:
```
pandas.core.frame.DataFrame
```

This essentially makes the column Year into a series and then turns that series into its own DataFrame

In [36]:
```
weo[[3]]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-36-5a76019d91f2> in <module>()
----> 1 weo[[3]]

/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in __getitem__(self, key)
   1956         if isinstance(key, (Series, np.ndarray, Index, list)):
   1957             # either boolean or fancy integer index
-> 1958             return self._getitem_array(key)
   1959         elif isinstance(key, DataFrame):
   1960             return self._getitem_frame(key)

/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in _getitem_array(self, key)
   2000             return self.take(indexer, axis=0, convert=False)
   2001         else:
-> 2002             indexer = self.loc._convert_to_indexer(key, axis=1)
   2003             return self.take(indexer, axis=1, convert=True)
   2004

/anaconda3/lib/python3.6/site-packages/pandas/core/indexing.py in _convert_to_indexer(self, obj, axis, is
_setter)
   1229                 mask = check == -1
   1230                 if mask.any():
-> 1231                     raise KeyError('%s not in index' % objarr[mask])
   1232
   1233                 return _values_from_object(indexer)

KeyError: '[3] not in index'
```

This doesn't work because there's no column labeled '3', i.e. 3 isn't an index

**h.**

In [37]:
```
weo['Year'].astype(float)
```

Out[37]:
```
0    2008.0
1    2009.0
2    2010.0
3    2011.0
4    2012.0
5    2013.0
6    2014.0
Name: Year, dtype: float64
```

**i.**

In [43]:
```
t = weo.tail(3)
type(t)
```

Out[43]:
```
pandas.core.frame.DataFrame
```

In [49]:
```
t
```

Out[49]:

| | BRA | JPN | USA | Year |
|---|---|---|---|---|
| **4** | 15.46 | 35.60 | 51.41 | 2012 |
| **5** | 15.98 | 36.79 | 52.94 | 2013 |
| **6** | 16.10 | 37.39 | 54.60 | 2014 |

In [51]:
```
type(t)
```

Out[51]:
```
pandas.core.frame.DataFrame
```

.tail returns the last n rows (in this case 3) of the data frame "weo" ; its type is a DataFrame

**j.**

In [81]:
```
t2 = weo.head(4)
t2
```

Out[81]:

| | BRA | JPN | USA | Year | c | gdp_ratio |
|---|---|---|---|---|---|---|
| **0** | 13.37 | 33.43 | 48.30 | 2008 | 0.399940 | 0.399940 |
| **1** | 13.30 | 31.83 | 46.91 | 2009 | 0.417845 | 0.417845 |
| **2** | 14.34 | 33.71 | 48.31 | 2010 | 0.425393 | 0.425393 |
| **3** | 15.07 | 34.29 | 49.72 | 2011 | 0.439487 | 0.439487 |

**k.**

In [82]:
```
weo['BRA']
```

Out[82]:
```
0    13.37
1    13.30
2    14.34
3    15.07
4    15.46
5    15.98
6    16.10
Name: BRA, dtype: float64
```

In [97]:
```
type(weo['BRA'])
```

Out[97]:
```
pandas.core.series.Series
```

**l.**

In [98]:
```
weo = weo.assign(gdp_ratio = weo['BRA'] / weo['JPN'])
weo
```

Out[98]:

| | BRA | JPN | USA | Year | c | gdp_ratio |
|---|---|---|---|---|---|---|
| **0** | 13.37 | 33.43 | 48.30 | 2008 | 0.399940 | 0.399940 |
| **1** | 13.30 | 31.83 | 46.91 | 2009 | 0.417845 | 0.417845 |
| **2** | 14.34 | 33.71 | 48.31 | 2010 | 0.425393 | 0.425393 |
| **3** | 15.07 | 34.29 | 49.72 | 2011 | 0.439487 | 0.439487 |
| **4** | 15.46 | 35.60 | 51.41 | 2012 | 0.434270 | 0.434270 |
| **5** | 15.98 | 36.79 | 52.94 | 2013 | 0.434357 | 0.434357 |
| **6** | 16.10 | 37.39 | 54.60 | 2014 | 0.430596 | 0.430596 |