- -> notebooks from this lecture: https://github.com/ine-rmotr-curriculum/freecodecamp-intro-to-pandas
- -> dataframes
- ***-> it looks like an excel table***
    - -> it's common to create pandas tables out of CSV files
    - -> columns and rows of values
    - -> a dataframe column will be a series
- ***-> the info method <- methods for dataframes***
    - -> for information about the structure of the data frame
    - -> what columns you have, population, gdp, surface area, the continent
    - -> how many null values you have
    - -> the size / shape of the dataframe
- ***-> the describe method***
    - -> for summary statistics
    - -> for each numeric column, there are summary statistics for them
    - -> the minimum, maximum, standard deviations
- ***-> in the info method***
    - -> the columns have associated types
    - -> floats, integers
    - -> the correct type is recognised and automatically assigned to the columns
- ***-> to select data from series***
    - -> df.loc <- this attribute lets you select individual rows
        - -> selecting by index using the loc attribute
    - -> .iloc <- to select the row by sequential position
        - -> if you want to select the last row, for example
    - -> df. column_name <- give me the column name
    - -> one gives you an element by index, the other gives you the element by position
    - -> the results are all series that are being returned
- ***-> for the last example***
    - -> there are the elements returned
    - -> if you ask for a row, then the result will also be a series
    - -> but in the dataset you are asking for it from its a row and in the result returned we have a transposed column
    - -> if the index is numeric
- ***-> others***
    - -> df.size <- rows by columns
    - -> df.shape
    - -> df.describe <- summary statistics -> median, mean, standard deviation
    - -> df.types <- e.g the continent being an object
        - -> int64 is a type
    - -> you can also check value counts
    - -> you can also select the last row by index
- ***-> df['Population'] <- select the entire column called population***
    - -> .loc <- by index
    - -> .iloc <- by position
    - -> this method is the entire column
    - -> all the results are series
        - -> this is what we saw before
    - -> the result is a series
        - -> it extracts, for example, an entire row from the data frame
        - -> that as a series
    - -> you can use iloc to return the data from one row to another row, using :
    - -> then from icloc, you can use slicing sequentially -> for example df.iloc[1:3]

    - -> you can also select entire columns from the dataframe
- ***-> question***

What will the following code print out?

```
import pandas as pd

certificates_earned = pd.DataFrame({
    'Certificates': [8, 2, 5, 6],
    'Time (in months)': [16, 5, 9, 12]
})

certificates_earned.index = ['Tom', 'Kris', 'Ahmad', 'Beau']

print(certificates_earned.iloc[2])
```

```
Tom      16
Kris      5
Ahmad     9
Beau     12
Name: Time (in months), dtype: int64
```

```
Certificates          6
Time (in months)     12
Name: Beau, dtype: int64
```

```
Certificates         5
Time (in months)     9
Name: Ahmad, dtype: int64 <- This one
```