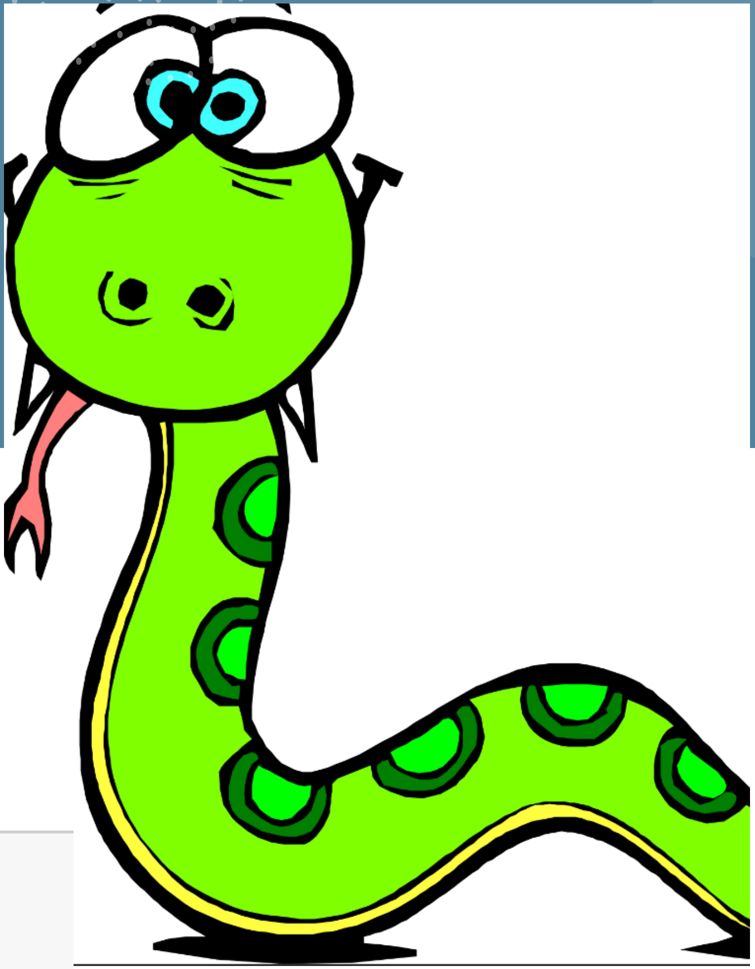
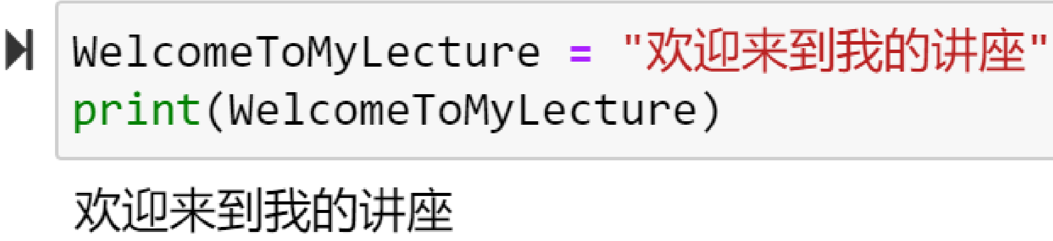
Welcome to my 11th Lecture 8th on Python

Lutz Plümer





```
WelcomeToMyLecture = "欢迎来到我的讲座"
print(WelcomeToMyLecture)
```

```
欢迎来到我的讲座
```

Overview – Schedule for today

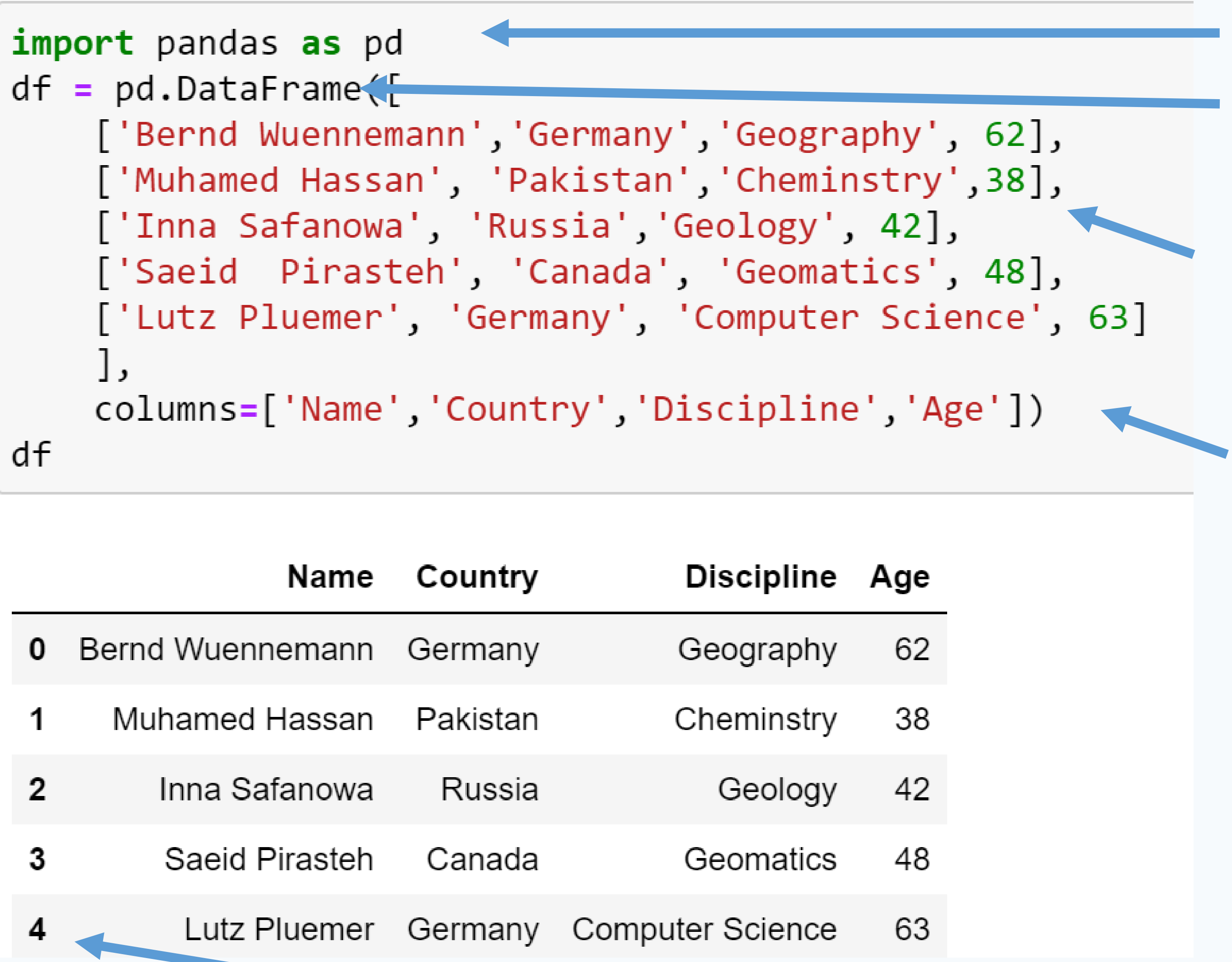
- This is my last Lecture before the Final Exam
- Today I will focus on the Panda Module
- Add some Rules for the Final Exam
-

Why on earth is the module called Pandas?

Pandas is derived from the term "**panel data**", an econometrics term for data sets that include **observations** over multiple **time periods** for the same **individuals**. — Wikipedia

What is Panda all about?

- Pandas is probably one of the most important, most **widely used** modules
- The focus is on the **processing** and **analysis** of **data**
- So it is all about **Data Science**
- Pandas has a strong connection to **databases**, can also handle very large amounts of data
- It also has a strong relationship with **Excel**
- At the center of Panda is the **Data Frame**
- Which is very similar to an **Excel sheet**



```python
import pandas as pd
df = pd.DataFrame([
    ['Bernd Wuennemann','Germany','Geography', 62],
    ['Muhamed Hassan', 'Pakistan','Cheminstry',38],
    ['Inna Safanowa', 'Russia','Geology', 42],
    ['Saeid  Pirasteh', 'Canada', 'Geomatics', 48],
    ['Lutz Pluemer', 'Germany', 'Computer Science', 63]
    ],
    columns=['Name','Country','Discipline','Age'])
df
```

	Name	Country	Discipline	Age
0	Bernd Wuennemann	Germany	Geography	62
1	Muhamed Hassan	Pakistan	Cheminstry	38
2	Inna Safanowa	Russia	Geology	42
3	Saeid Pirasteh	Canada	Geomatics	48
4	Lutz Pluemer	Germany	Computer Science	63

Import the module

DataFrame **Constructor**

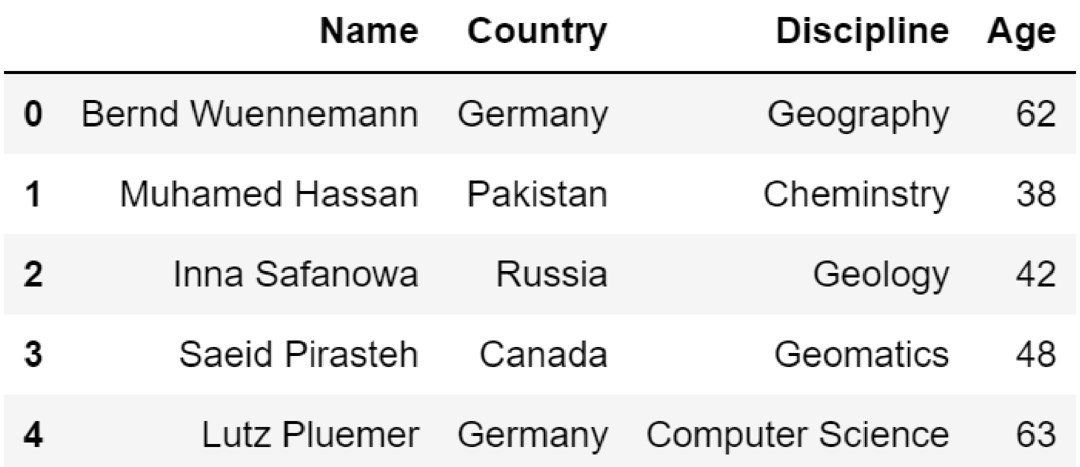
Dataframe Items as **Lists of Attributes**

Last argumen gives **the column names**

Nice printout of the resulting Dataframe

Default Names for the Rows

Date Frame, Excel Sheet



	Name	Country	Discipline	Age
0	Bernd Wuennemann	Germany	Geography	62
1	Muhamed Hassan	Pakistan	Cheminstry	38
2	Inna Safanowa	Russia	Geology	42
3	Saeid Pirasteh	Canada	Geomatics	48
4	Lutz Pluemer	Germany	Computer Science	63

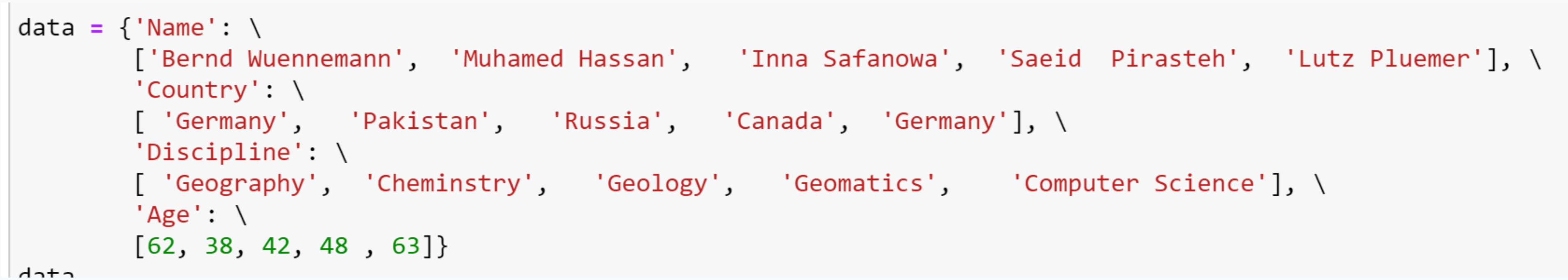
- A Panda Dataframe is just an ordinary table, with **items** as **rows**, attributes (features) as columns, and attribute names as header
- Very similar to an excel sheet
- Different **columns** have different **types**:
  name contry and discipline are text
  Age is Integer
  and the index on the left is integer as well

Date Frame from Dictionary

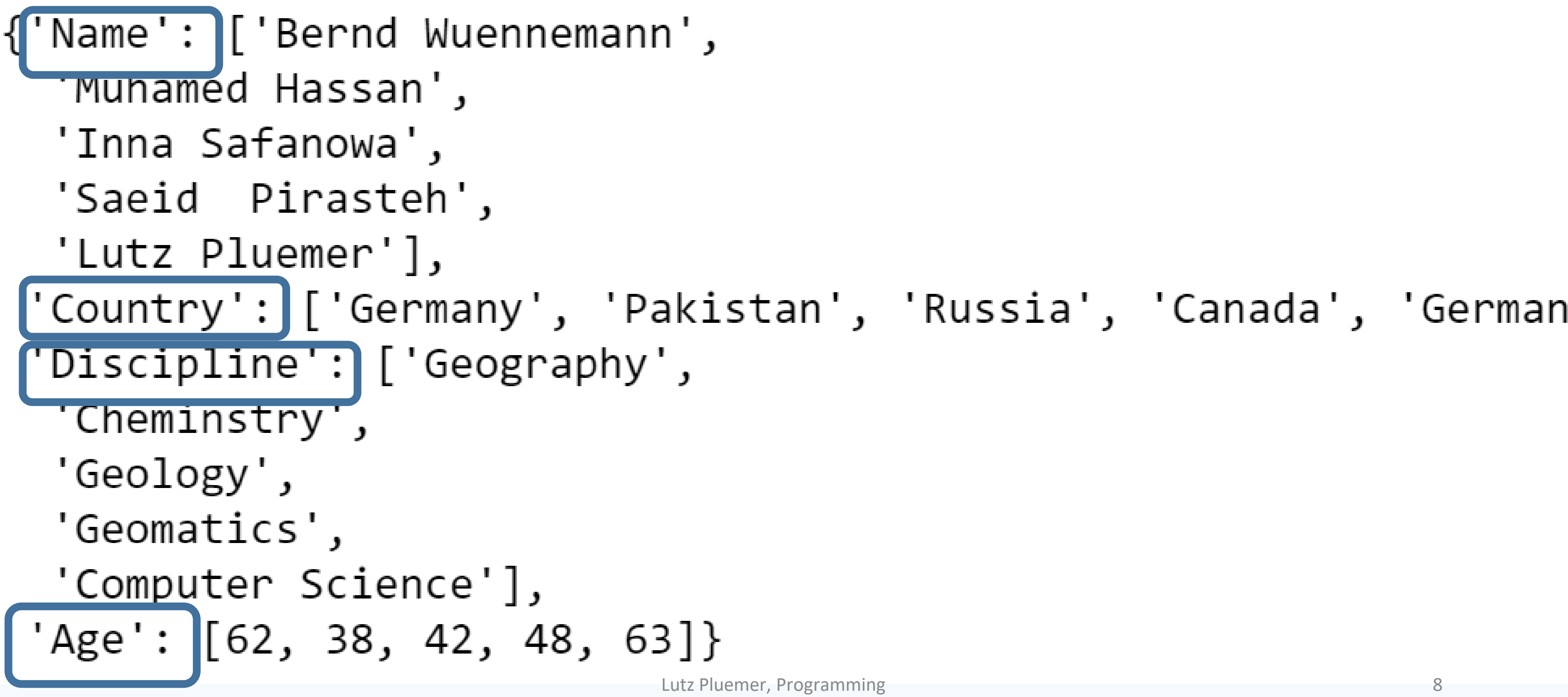
- There is another way to construct a data frame: from a dictionary
- You remember: dictionarys are key-value pair sets with a “:” in the midst
- Here they **keys** are the **attribute names**, and the values are the columns

```
{'Name': \
 {'Bernd Wuennemann',  'Muhamed Hassan',   'Inna Safanowa’, \
 'Saeid  Pirasteh',  'Lutz Pluemer'}, \
'Country': \
{ 'Germany',   'Pakistan',   'Russia',   'Canada',  'Germany'}, \
'Discipline': \
{ 'Geography',  'Chemistry',   'Geology',   'Geomatics',    'Computer Science'}, \
'Age': \
{62, 38, 42, 48 , 63}}
```

-

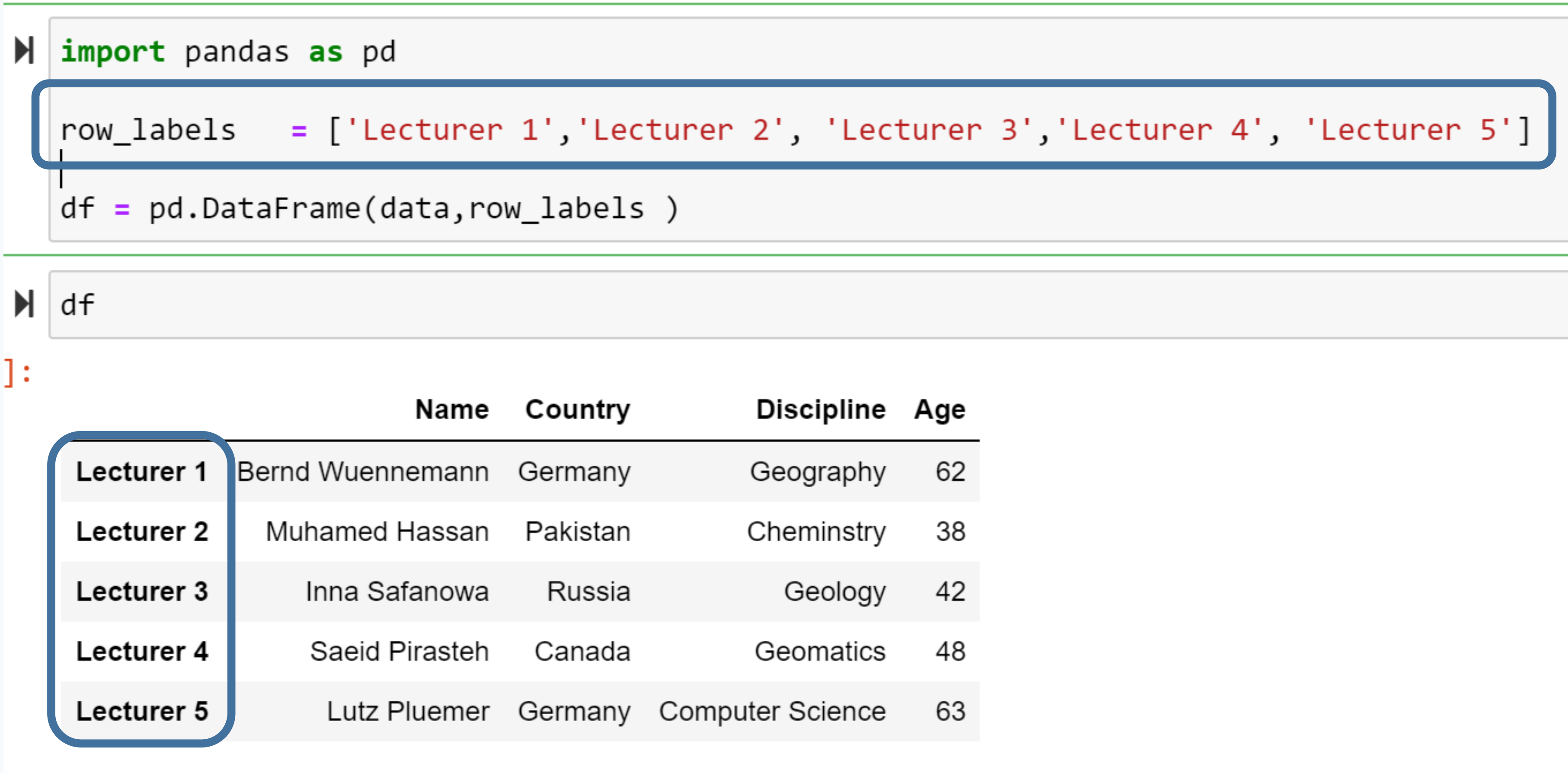


```
data = {'Name': \
        ['Bernd Wuennemann',  'Muhamed Hassan',   'Inna Safanowa',  'Saeid  Pirasteh',  'Lutz Pluemer'], \
        'Country': \
        [ 'Germany',   'Pakistan',   'Russia',   'Canada',  'Germany'], \
        'Discipline': \
        [ 'Geography',  'Cheminstry',   'Geology',   'Geomatics',    'Computer Science'], \
        'Age': \
        [62, 38, 42, 48 , 63]}
data
```



```
{'Name': ['Bernd Wuennemann',
  'Muhamed Hassan',
  'Inna Safanowa',
  'Saeid  Pirasteh',
  'Lutz Pluemer'],
 'Country': ['Germany', 'Pakistan', 'Russia', 'Canada', 'German
 'Discipline': ['Geography',
  'Cheminstry',
  'Geology',
  'Geomatics',
  'Computer Science'],
 'Age': [62, 38, 42, 48, 63]}
```

Dataframe with self defined row_labels



```
import pandas as pd

row_labels    = ['Lecturer 1','Lecturer 2', 'Lecturer 3','Lecturer 4', 'Lecturer 5']

df = pd.DataFrame(data,row_labels )
```

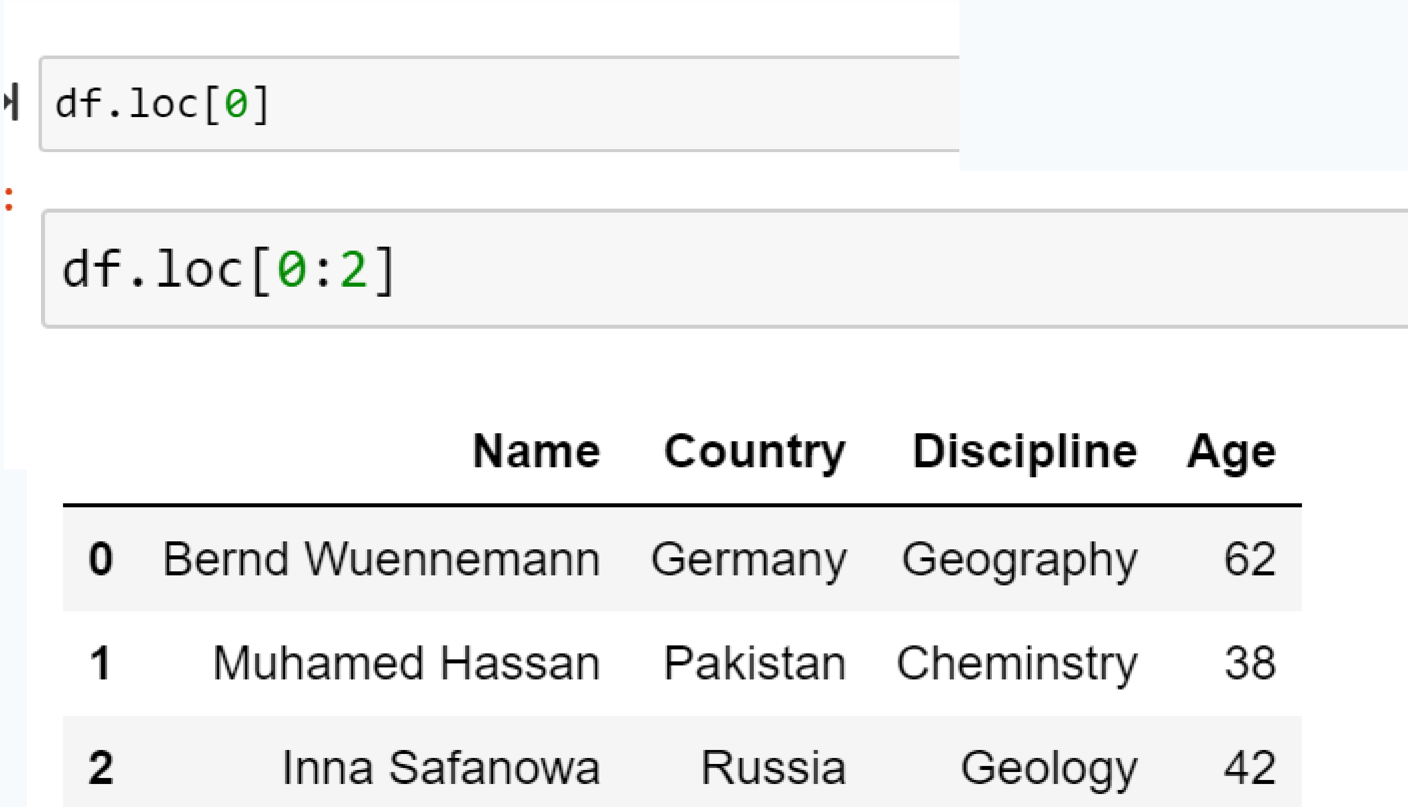
```
df
```

]:

	Name	Country	Discipline	Age
Lecturer 1	Bernd Wuennemann	Germany	Geography	62
Lecturer 2	Muhamed Hassan	Pakistan	Cheminstry	38
Lecturer 3	Inna Safanowa	Russia	Geology	42
Lecturer 4	Saeid Pirasteh	Canada	Geomatics	48
Lecturer 5	Lutz Pluemer	Germany	Computer Science	63

Selection of (a) single row(s)

- You can access single rows by df.loc[..]
  note that df is an object and loc is an attribute, so you access with [..] rather than (..)
- You can also write df.iloc[1]

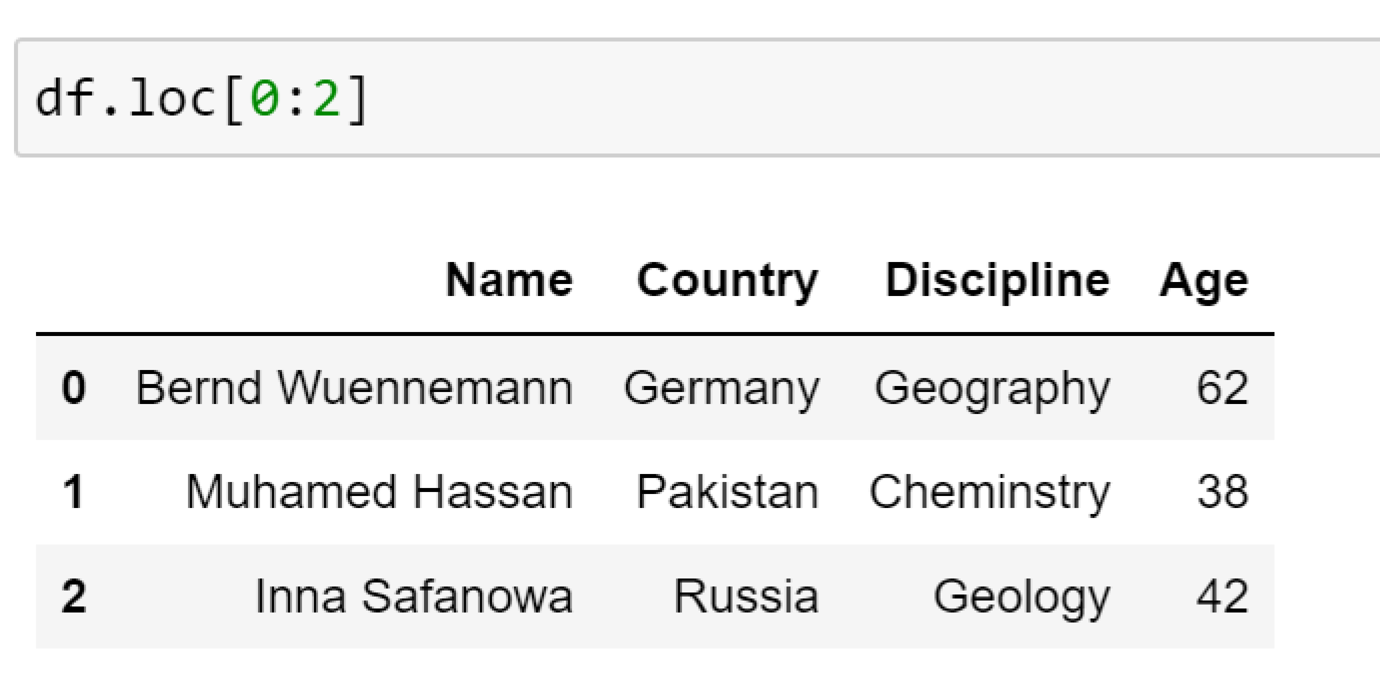


```
df.loc[0]
```

```
df.loc[0:2]
```

	Name	Country	Discipline	Age
0	Bernd Wuennemann	Germany	Geography	62
1	Muhamed Hassan	Pakistan	Cheminstry	38
2	Inna Safanowa	Russia	Geology	42

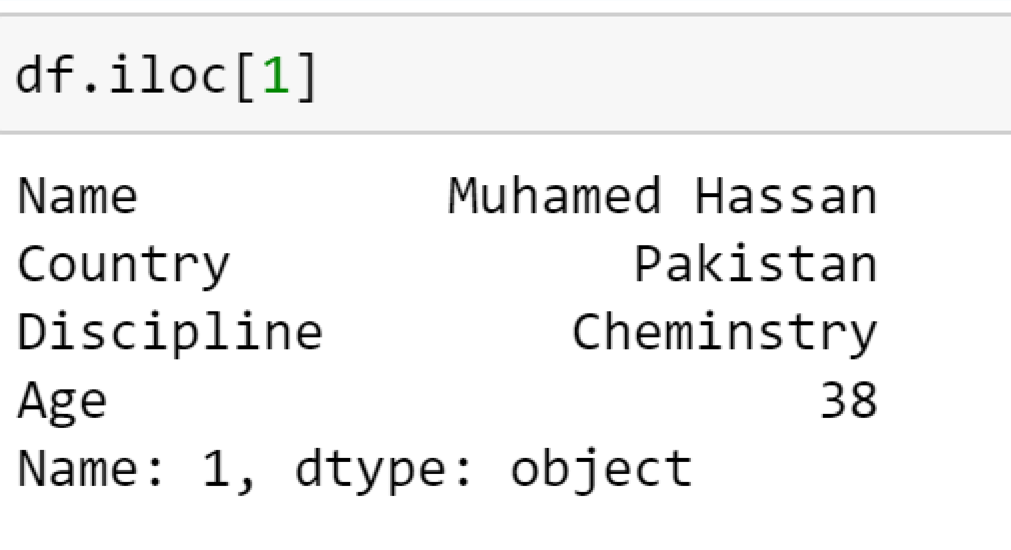
Slicing with 0:2



```
df.loc[0:2]
```

	Name	Country	Discipline	Age
0	Bernd Wuennemann	Germany	Geography	62
1	Muhamed Hassan	Pakistan	Cheminstry	38
2	Inna Safanowa	Russia	Geology	42

df.iloc

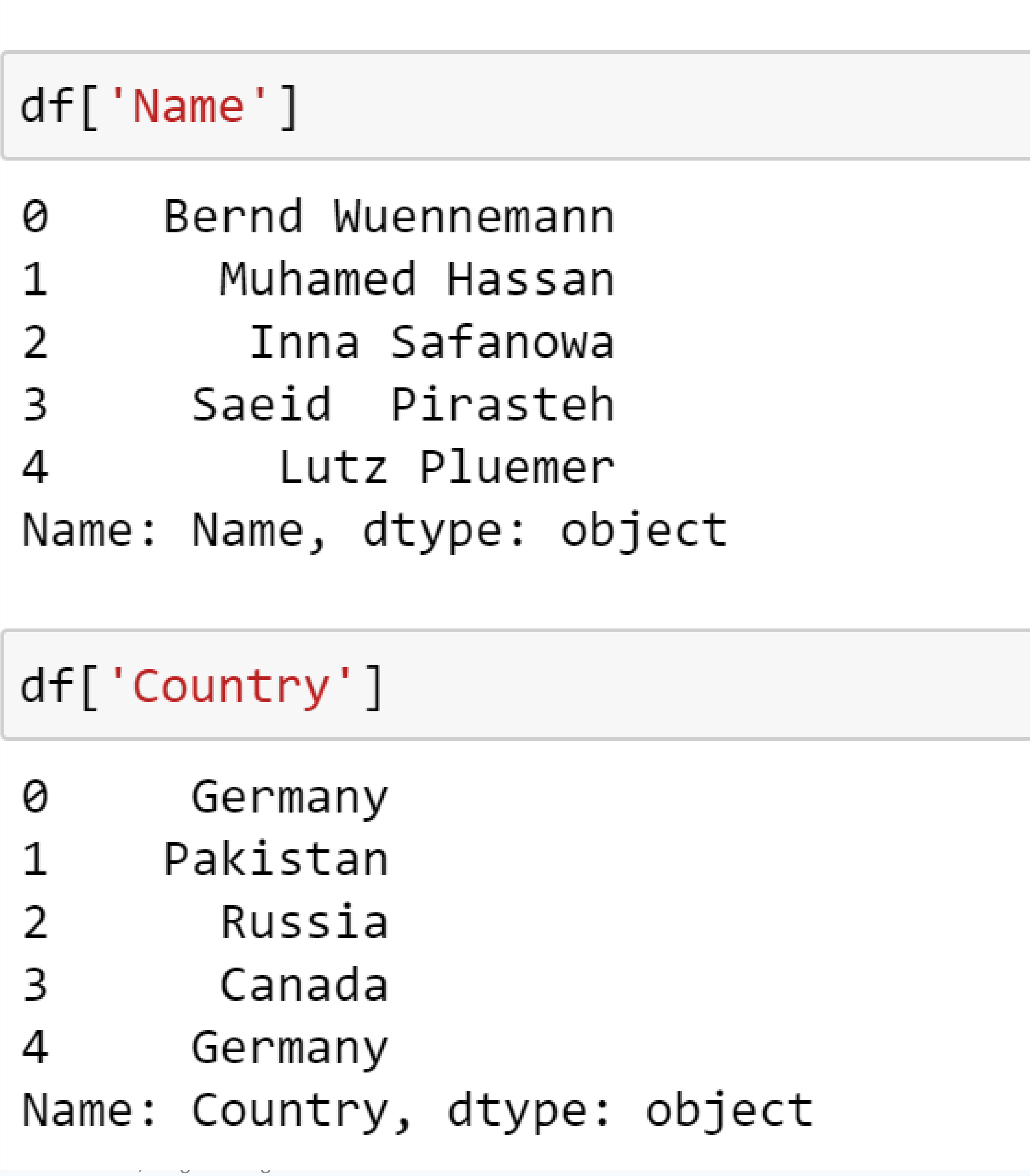


```
df.iloc[1]
```

```
Name          Muhamed Hassan
Country             Pakistan
Discipline        Cheminstry
Age                       38
Name: 1, dtype: object
```

Selection of Columns

- You can select columns with df[column_name], such as df[“Name”] or df[“Country”]



```
df['Name']
```

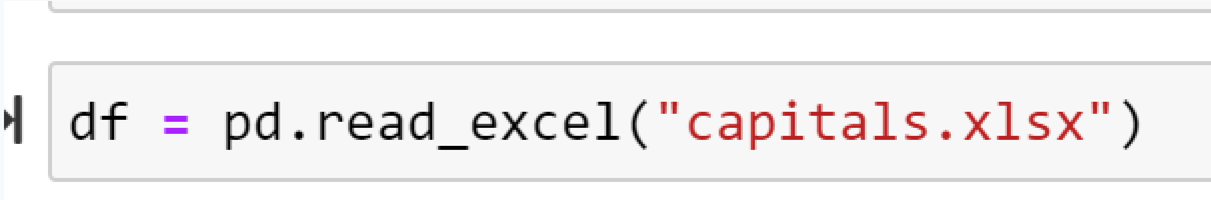
```
0    Bernd Wuennemann
1      Muhamed Hassan
2       Inna Safanowa
3     Saeid  Pirasteh
4        Lutz Pluemer
Name: Name, dtype: object
```

```
df['Country']
```

```
0     Germany
1    Pakistan
2      Russia
3      Canada
4     Germany
Name: Country, dtype: object
```

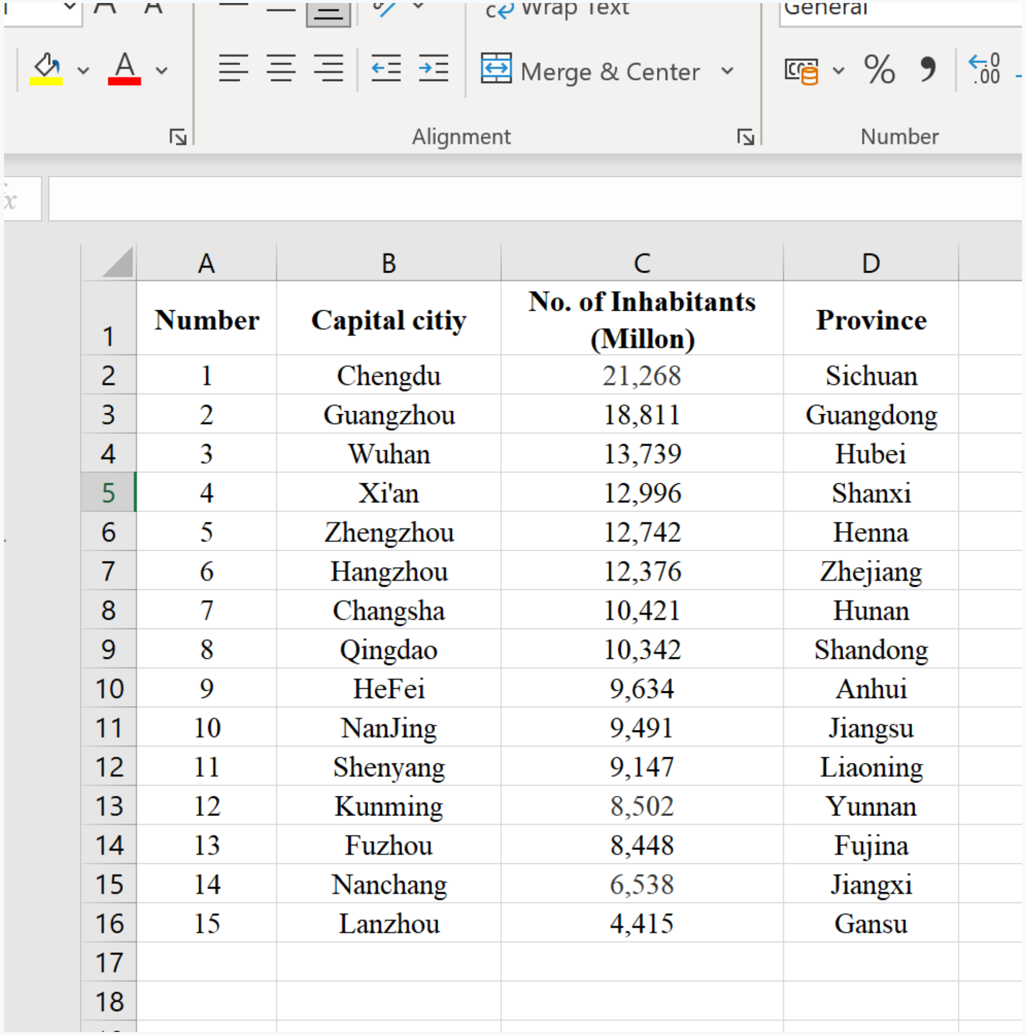
Pada Dataframes and Excel Sheets

- Panda Dataframe are very similar do Excel Data Sheets
- So you can easily import und export them
- Import is easy when the excel files has just one sheet, like here:



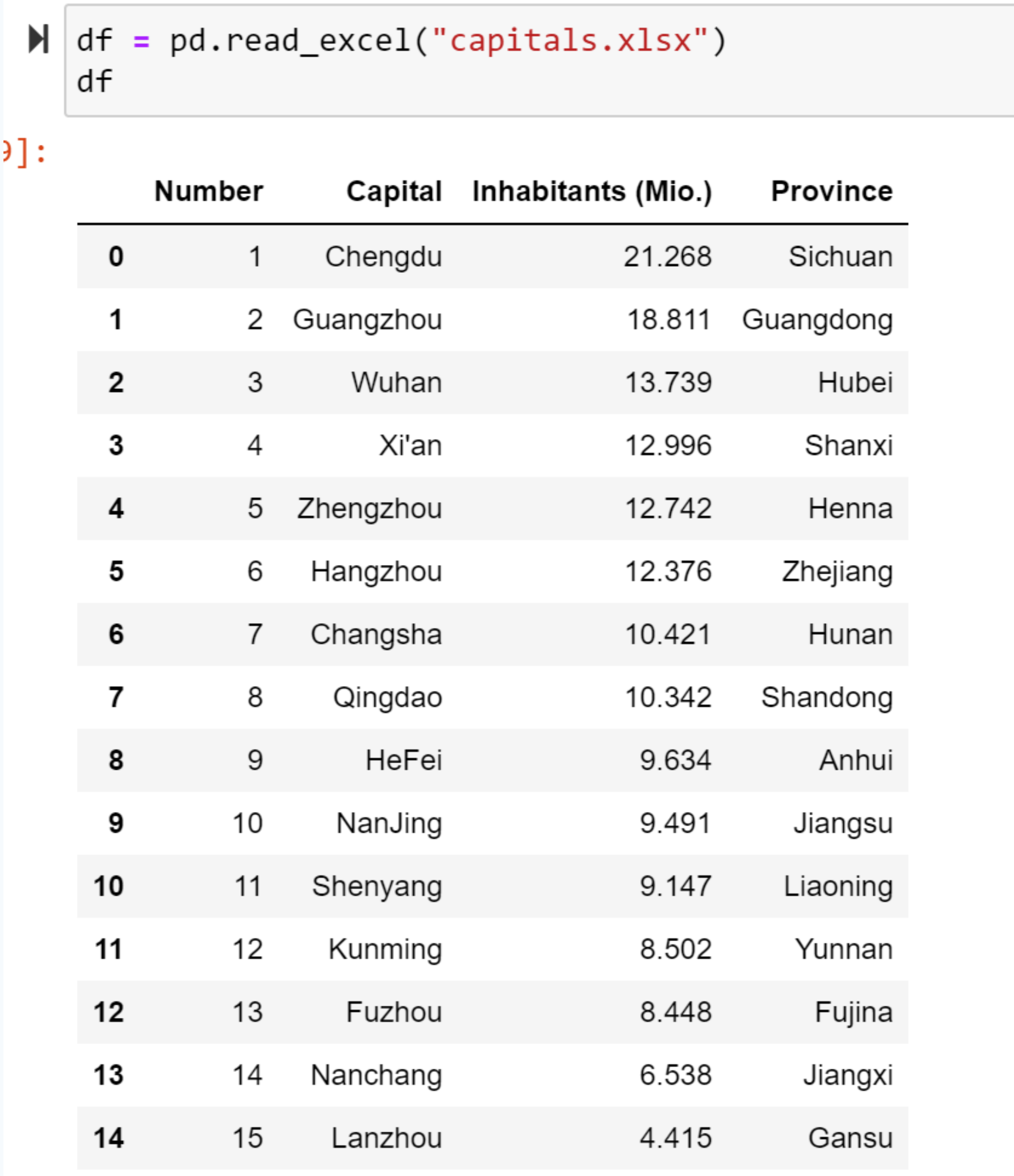
```
df = pd.read_excel("capitals.xlsx")
```

This is the Excel Sheet



	A	B	C	D
1	Number	Capital citiy	No. of Inhabitants (Millon)	Province
2	1	Chengdu	21,268	Sichuan
3	2	Guangzhou	18,811	Guangdong
4	3	Wuhan	13,739	Hubei
5	4	Xi'an	12,996	Shanxi
6	5	Zhengzhou	12,742	Henna
7	6	Hangzhou	12,376	Zhejiang
8	7	Changsha	10,421	Hunan
9	8	Qingdao	10,342	Shandong
10	9	HeFei	9,634	Anhui
11	10	NanJing	9,491	Jiangsu
12	11	Shenyang	9,147	Liaoning
13	12	Kunming	8,502	Yunnan
14	13	Fuzhou	8,448	Fujina
15	14	Nanchang	6,538	Jiangxi
16	15	Lanzhou	4,415	Gansu
17				
18				

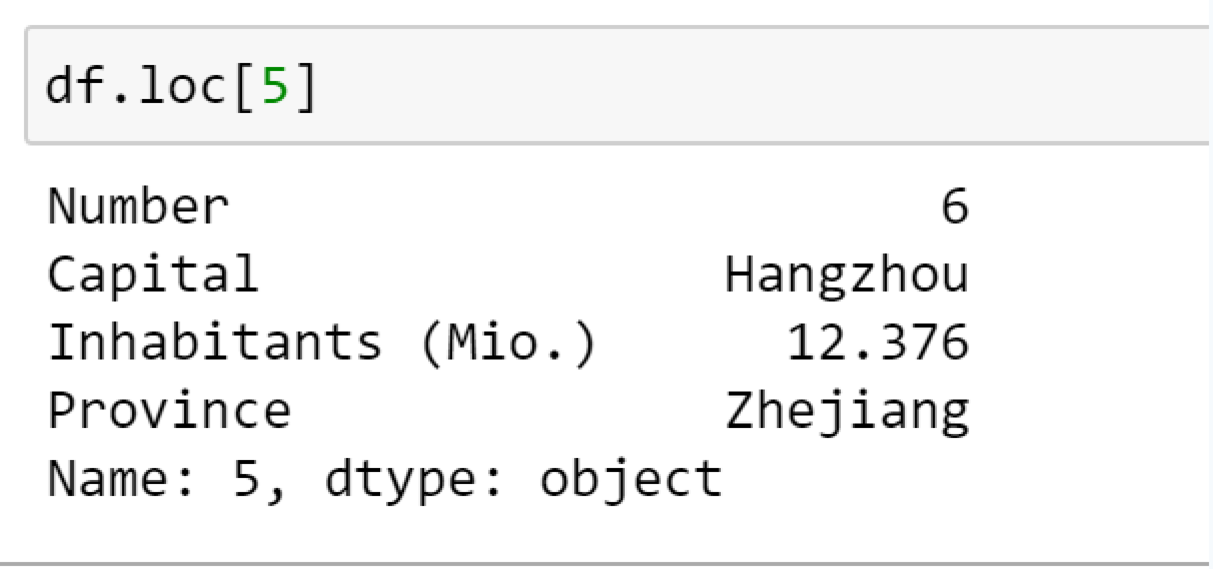
This is the imported Panda Data Frame



```
df = pd.read_excel("capitals.xlsx")
df
```

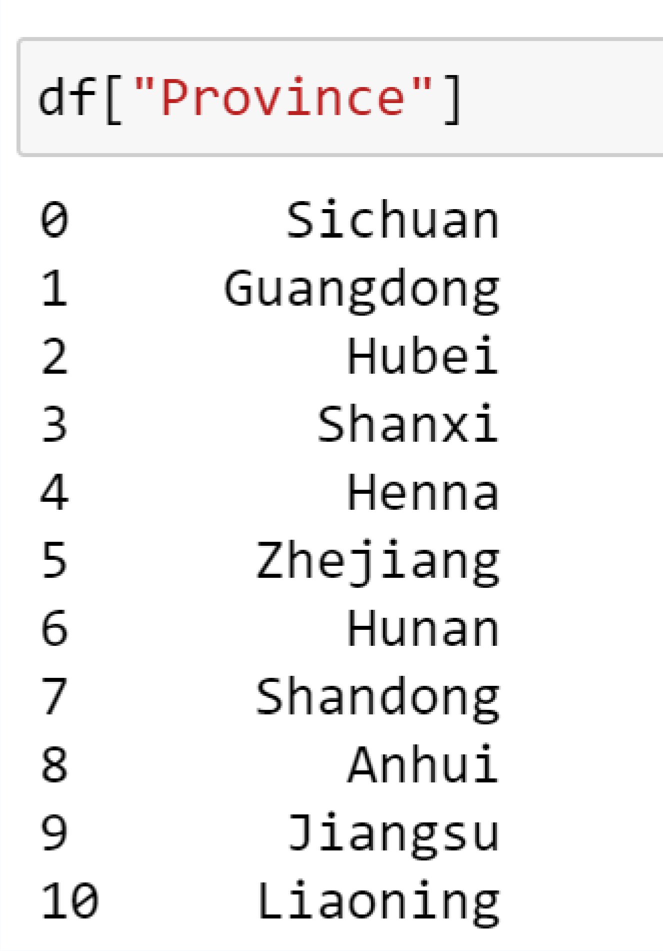
	Number	Capital	Inhabitants (Mio.)	Province
0	1	Chengdu	21.268	Sichuan
1	2	Guangzhou	18.811	Guangdong
2	3	Wuhan	13.739	Hubei
3	4	Xi'an	12.996	Shanxi
4	5	Zhengzhou	12.742	Henna
5	6	Hangzhou	12.376	Zhejiang
6	7	Changsha	10.421	Hunan
7	8	Qingdao	10.342	Shandong
8	9	HeFei	9.634	Anhui
9	10	NanJing	9.491	Jiangsu
10	11	Shenyang	9.147	Liaoning
11	12	Kunming	8.502	Yunnan
12	13	Fuzhou	8.448	Fujina
13	14	Nanchang	6.538	Jiangxi
14	15	Lanzhou	4.415	Gansu

Select Rows and Columns



```
df.loc[5]
```

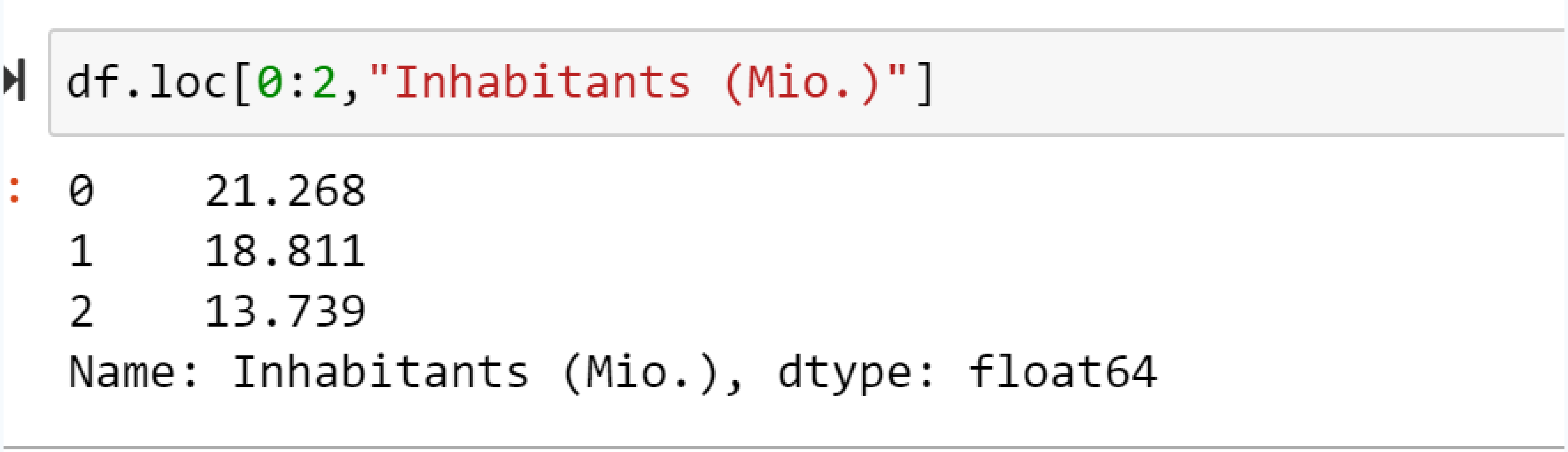
```
Number                        6
Capital                Hangzhou
Inhabitants (Mio.)       12.376
Province               Zhejiang
Name: 5, dtype: object
```



```
df["Province"]
```

```
0       Sichuan
1     Guangdong
2         Hubei
3        Shanxi
4         Henna
5      Zhejiang
6         Hunan
7      Shandong
8         Anhui
9       Jiangsu
10     Liaoning
```

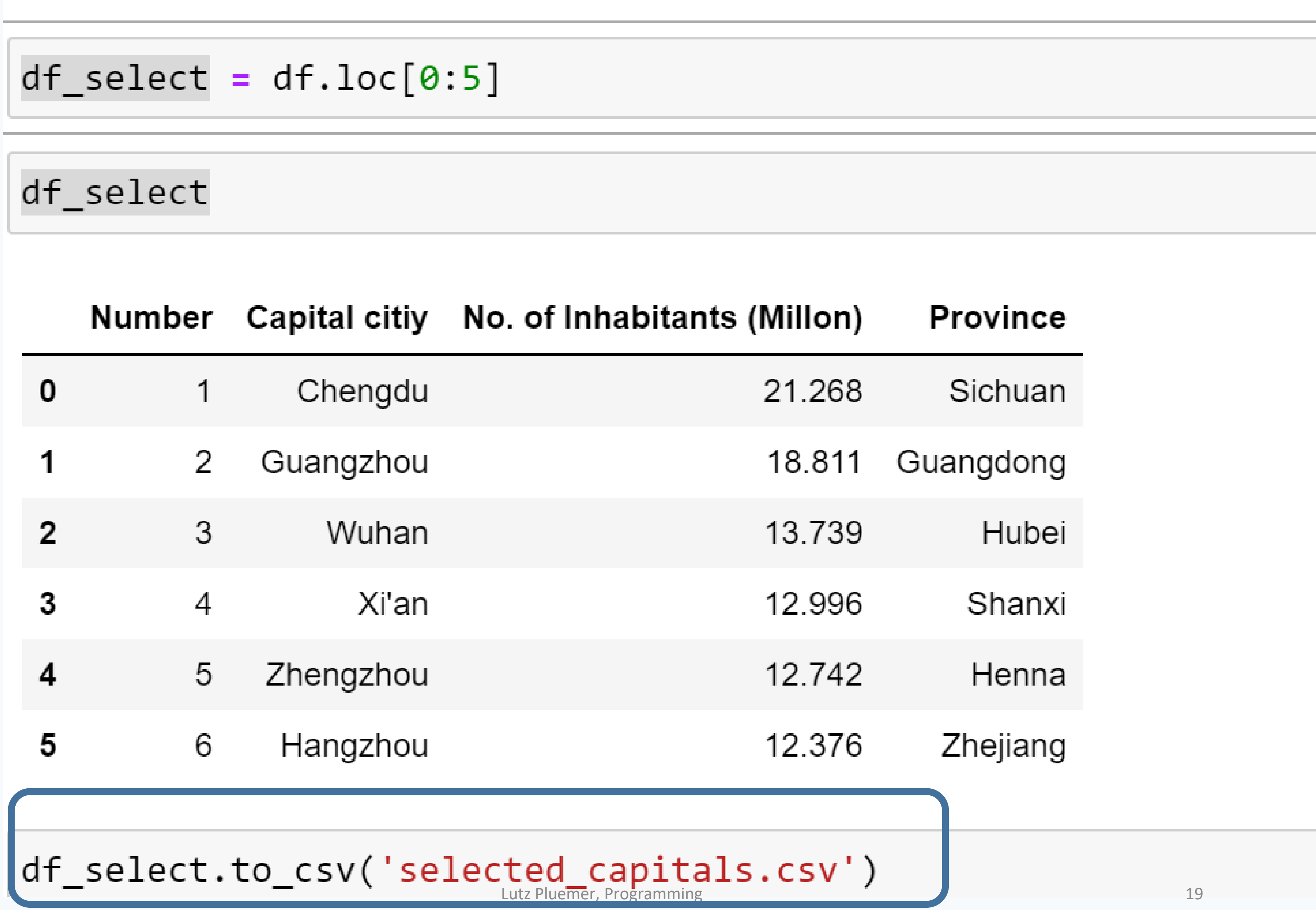
You can also combine via loc



```
df.loc[0:2,"Inhabitants (Mio.)"]
```

```
0    21.268
1    18.811
2    13.739
Name: Inhabitants (Mio.), dtype: float64
```

You can write an CSV File



```
df_select = df.loc[0:5]
```

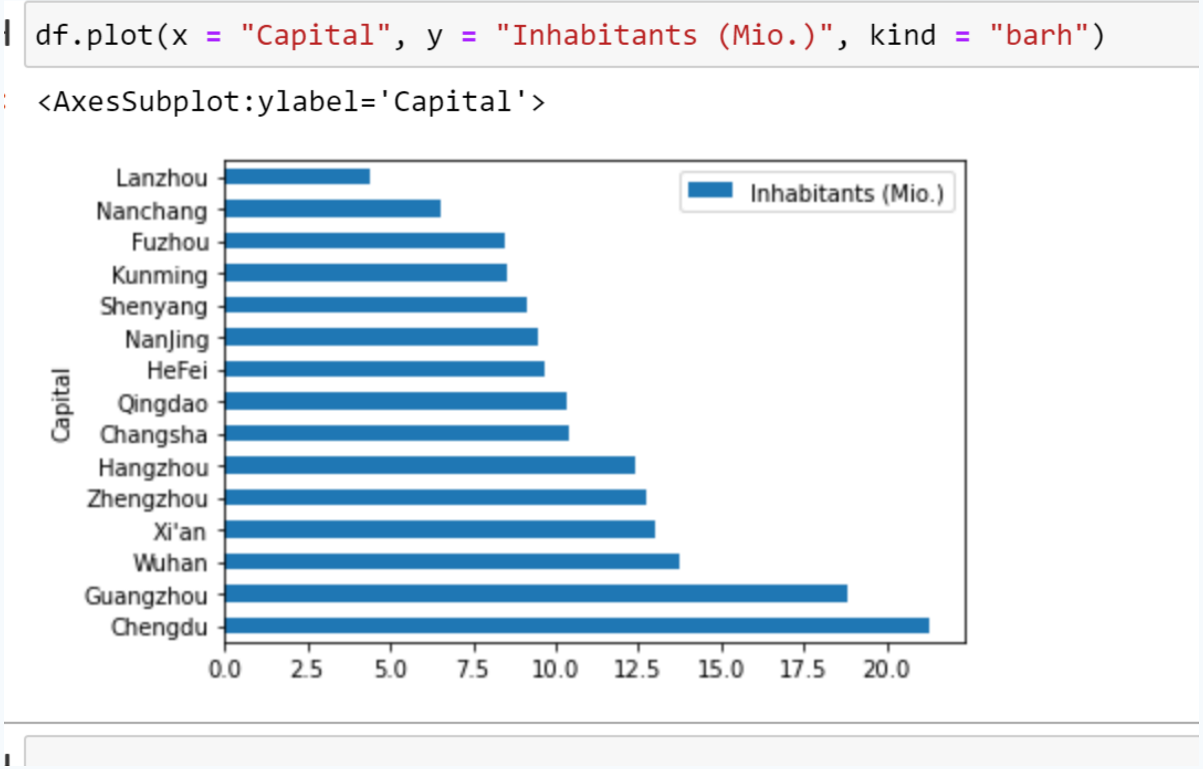
```
df_select
```

	Number	Capital citiy	No. of Inhabitants (Millon)	Province
0	1	Chengdu	21.268	Sichuan
1	2	Guangzhou	18.811	Guangdong
2	3	Wuhan	13.739	Hubei
3	4	Xi'an	12.996	Shanxi
4	5	Zhengzhou	12.742	Henna
5	6	Hangzhou	12.376	Zhejiang

```
df_select.to_csv('selected_capitals.csv')
```

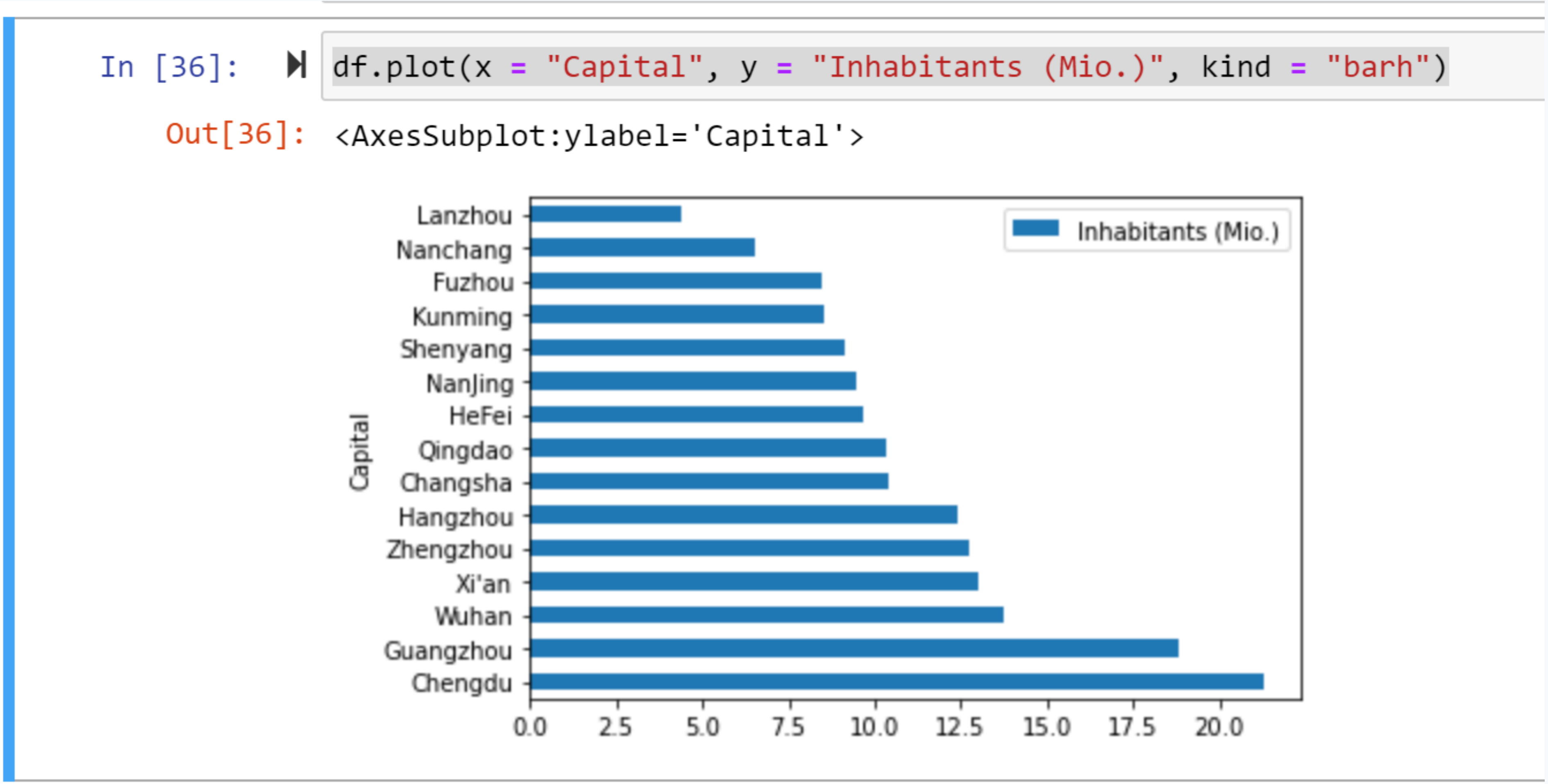
And you can make Charts

- You select the columns via x = … and y = … and select the chart with kind = …
  such as here
  df.plot(x = "Capital", y = "Inhabitants (Mio.)", kind = "barh")



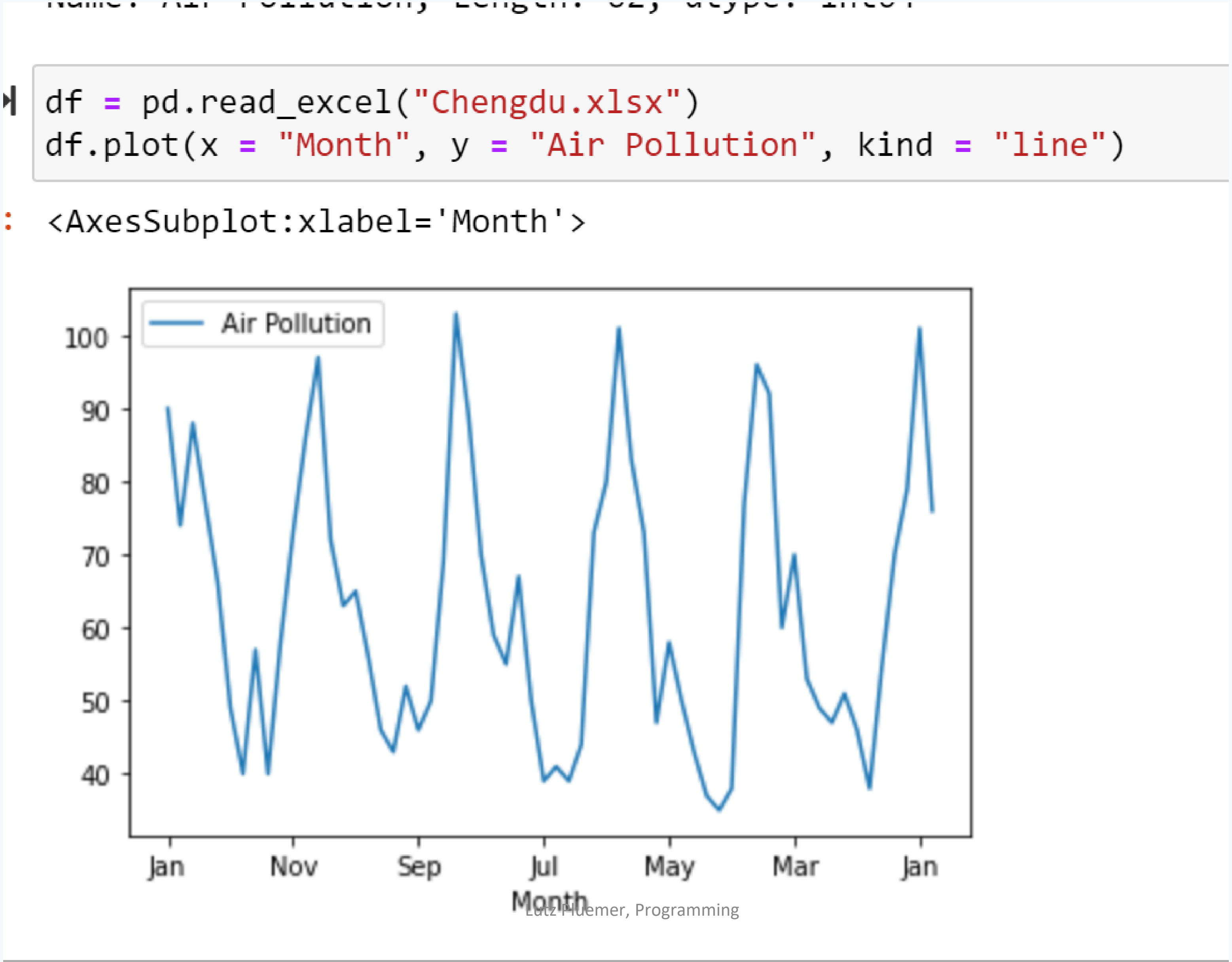
```
df.plot(x = "Capital", y = "Inhabitants (Mio.)", kind = "barh")
```

```
<AxesSubplot:ylabel='Capital'>
```



```
In [36]:  df.plot(x = "Capital", y = "Inhabitants (Mio.)", kind = "barh")
Out[36]: <AxesSubplot:ylabel='Capital'>
```

Line Chart of Chengdu Air Pollution Data



```
Name: Air Pollution, Length: 62, dtype: int64
```

```python
df = pd.read_excel("Chengdu.xlsx")
df.plot(x = "Month", y = "Air Pollution", kind = "line")
```

```
: <AxesSubplot:xlabel='Month'>
```

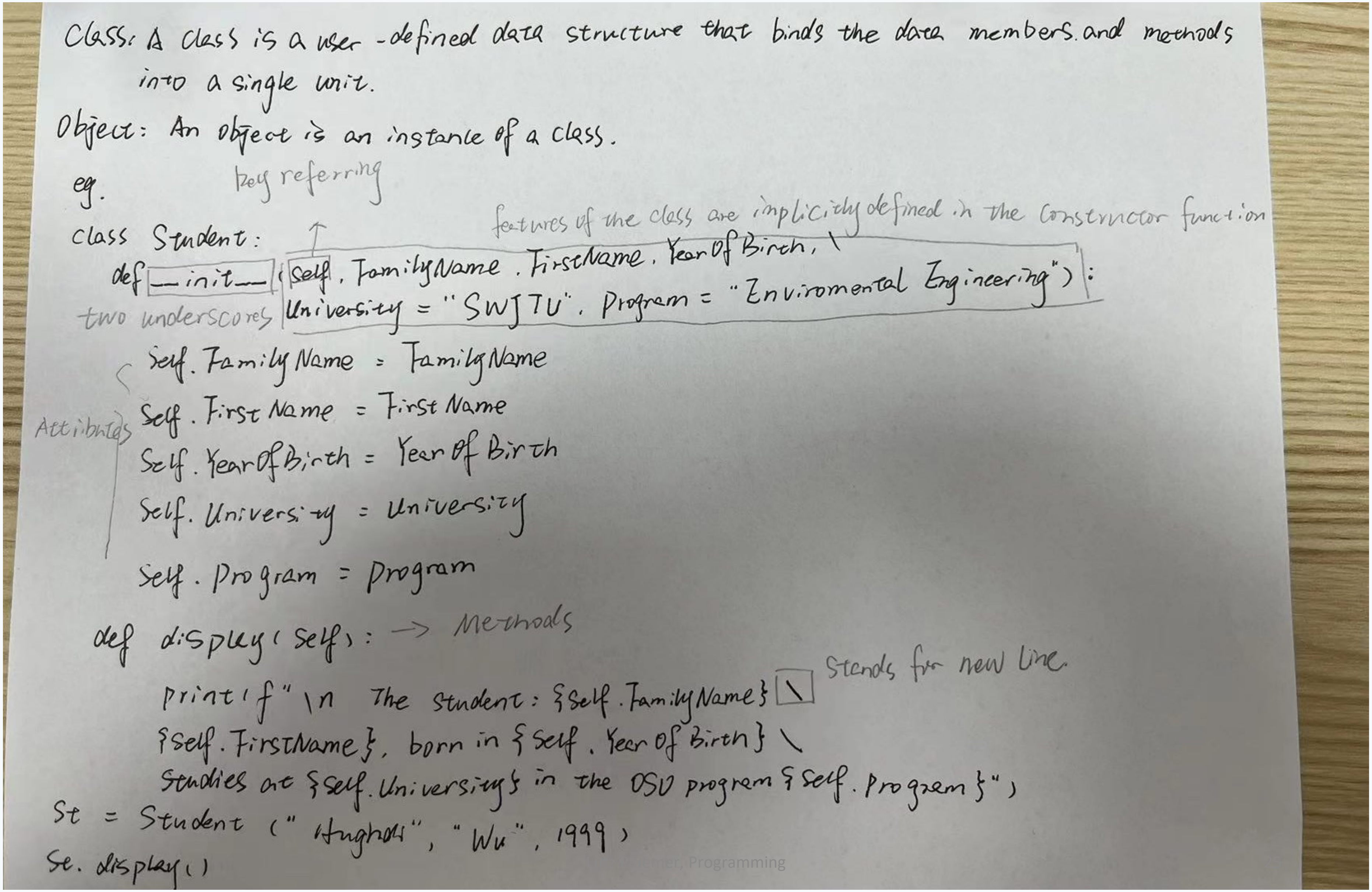
Classroom Exercise

- Code the Teachers Frame as described above, in tow ways, by items and by columns
- Select single items
- Select singe Columns
- Import the Capitals dataset from the excel sheet (put it in the same directory as your Jupyter files), display, select single rows and columns
- Generate the Bar Chart
- Import the Chengdu weather dataset and produce the Line Chart

Homework

- No special homework on Panda
- Prepare the Final Exam:
- Prepare your handwritten Notes
- Go through all the Panda exercises and review the ppt-slides in order to understand better

Another Example for Notes (Classes and Objects) by Lianlin Wu



Class: A class is a user-defined data structure that binds the data members. and methods into a single unit.

Object: An object is an instance of a class.

eg.

key referring

features of the class are implicitly defined in the constructor function

```
class Student:
   def __init__(self, FamilyName, FirstName, YearOfBirth, \
                University = "SWJTU", Program = "Enviromental Engineering"):
      self.FamilyName = FamilyName
      self.FirstName = First Name
      self.YearOfBirth = Year of Birth
      self.University = University
      self.Program = program
   def display(self):
      print(f"\n The student: {self.FamilyName} \
      {self.FirstName}, born in {self.Year of Birth} \
      Studies at {self.University} in the OSU program {self.Program}")
St = Student("Hughos", "Wu", 1999)
Se. display()
```

two underscores

Attributes

-> Methods

Stands for new line.

Final Exam, Rules

- **Permitted** Aids:
  - your own **computer**
  - **Jupyter**, but only the notebook with the Final Exam Exercises. Make sure you have access to Matplablotilb, Numpy and Panda.
  - Excel is **not** needed
  - **Notes** on 2 sheets of paper handwritten on both sides by your own hand
  - You may bring your cell phone, but it must be turned off before the exam begins.
- **Not allowed**
  - Hotspots
  - Access to the internet, except when downloading your results. You will the Wifi password then.
  - Access to the **ppt slides** of the lecture and to **old** exercises
  - Any form of **communication** and **exchange** with each other or with outsiders, be it wechat, qq or whatever.