INTERMEDIATE PYTHON FOR DATA SCIENCE

Dictionaries, Part 1

List

```
In [1]: pop = [30.55, 2.77, 39.21]

In [2]: countries = ["afghanistan", "albania", "algeria"]

In [3]: ind_alb = countries.index("albania")

In [4]: ind_alb
Out[4]: 1

In [5]: pop[ind_alb]
Out[5]: 2.77
```

Not convenient
Not intuitive

Dictionary

```
In [1]: pop = [30.55, 2.77, 39.21]

In [2]: countries = ["afghanistan", "albania", "algeria"]

...

In [6]: world = {"afghanistan":30.55, "albania":2.77, "algeria":39.21}

In [7]: world["albania"]
Out[7]: 2.77
```

dict_name[key]

result: **value**

INTERMEDIATE PYTHON FOR DATA SCIENCE

Let's practice!

INTERMEDIATE PYTHON FOR DATA SCIENCE

Dictionaries, Part 2

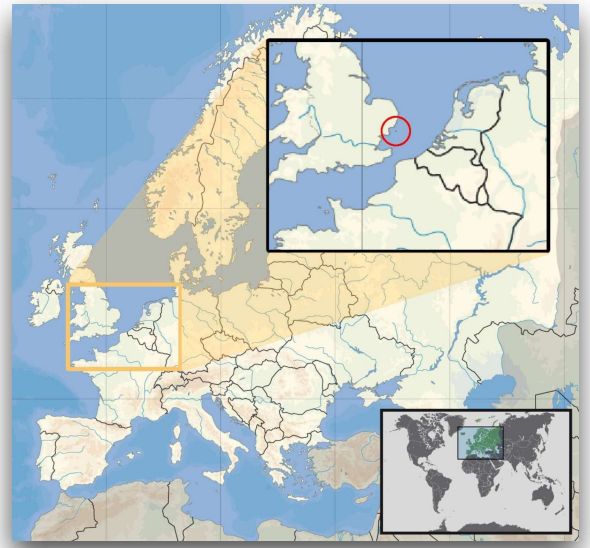
Recap

```
In [1]: world = {"afghanistan":30.55, "albania":2.77, "algeria":39.21}

In [2]: world["albania"]
Out[2]: 2.77

In [3]: world = {"afghanistan":30.55, "albania":2.77,
                 "algeria":39.21, "albania":2.81}

In [4]: world
Out[4]: {'afghanistan': 30.55, 'albania': 2.81, 'algeria': 39.21}
```

keys have to be "immutable" objects

```
In [5]: {0:"hello", True:"dear", "two":"world"}
Out[5]: {0: 'hello', True: 'dear', 'two': 'world'}

In [6]: {["just", "to", "test"]: "value"}
TypeError: unhashable type: 'list'
```

Principality of Sealand



Source: Wikipedia

Dictionary

```
In [8]: world["sealand"] = 0.000027

In [9]: world
Out[9]: {'afghanistan': 30.55, 'albania': 2.81,
                'algeria': 39.21, 'sealand': 2.7e-05}

In [10]: "sealand" in world
Out[10]: True

In [11]: world["sealand"] = 0.000028

In [12]: world
Out[12]: {'afghanistan': 30.55, 'albania': 2.81,
          'algeria': 39.21, 'sealand': 2.8e-05}

In [13]: del(world["sealand"])

In [14]: world
Out[14]: {'afghanistan': 30.55, 'albania': 2.81, 'algeria': 39.21}
```

List vs Dictionary

List	Dictionary
Select, update and remove: [ ]	Select, update and remove: [ ]
Indexed by range of numbers	Indexed by unique keys
Collection of values order matters select entire subsets	Lookup table with unique keys

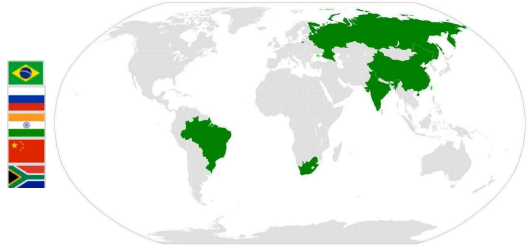
INTERMEDIATE PYTHON FOR DATA SCIENCE

Let's practice!

INTERMEDIATE PYTHON FOR DATA SCIENCE

Pandas, Part 1

Tabular dataset examples



temperature	measured_at	location
76	2016-01-01 14:00:01	valve
86	2016-01-01 14:00:01	compressor
72	2016-01-01 15:00:01	valve
88	2016-01-01 15:00:01	compressor
68	2016-01-01 16:00:01	valve
78	2016-01-01 16:00:01	compressor

row = observations
column = variable

country	capital	area	population
Brazil	Brasilia	8.516	200.4
Russia	Moscow	17.10	143.5
India	New Delhi	3.286	1252
China	Beijing	9.597	1357
South Africa	Pretoria	1.221	52.98

Datasets in Python

- 2D Numpy array?
  - One data type

country	capital	area	population
Brazil	Brasilia	8.516	200.4
Russia	Moscow	17.10	143.5
India	New Delhi	3.286	1252
China	Beijing	9.597	1357
South Africa	Pretoria	1.221	52.98
`str`	`str`	`float`	`float`

- Pandas!
  - High level data manipulation tool
  - Wes McKinney
  - Built on Numpy
  - `DataFrame`

DataFrame

```
In [1]: brics
Out[1]:
         country    capital    area  population
BR        Brazil   Brasilia   8.516      200.40
RU        Russia     Moscow  17.100      143.50
IN         India  New Delhi   3.286     1252.00
CH         China    Beijing   9.597     1357.00
SA  South Africa   Pretoria   1.221       52.98
```

DataFrame

```
In [1]: brics
Out[1]:
          country    capital    area  population
BR         Brazil   Brasilia   8.516      200.40
RU         Russia     Moscow  17.100      143.50
IN          India  New Delhi   3.286     1252.00
CH          China    Beijing   9.597     1357.00
SA  South Africa    Pretoria   1.221       52.98
```

observations

DataFrame

```
In [1]: brics
Out[1]:
```

variables

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

DataFrame

```
In [1]: brics
Out[1]:
```

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

column labels

row labels

columns with different types

DataFrame from Dictionary

```
In [2]:  dict = {
    "country":["Brazil", "Russia", "India", "China", "South Africa"],
    "capital":["Brasilia", "Moscow", "New Delhi", "Beijing", "Pretoria"],
       "area":[8.516, 17.10, 3.286, 9.597, 1.221]
 "population":[200.4, 143.5, 1252, 1357, 52.98] }
```

keys (column labels) **values (data, column by column)**

```
In [3]: import pandas as pd

In [4]: brics = pd.DataFrame(dict)
```

DataFrame from Dictionary (2)

```
In [5]: brics
Out[5]:
      area    capital       country  population
0    8.516   Brasilia        Brazil      200.40
1   17.100     Moscow        Russia      143.50
2    3.286  New Delhi         India     1252.00
3    9.597    Beijing         China     1357.00
4    1.221   Pretoria  South Africa       52.98

In [6]: brics.index = ["BR", "RU", "IN", "CH", "SA"]
In [7]: brics
Out[7]:
      area    capital       country  population
BR   8.516   Brasilia        Brazil      200.40
RU  17.100     Moscow        Russia      143.50
IN   3.286  New Delhi         India     1252.00
CH   9.597    Beijing         China     1357.00
SA   1.221   Pretoria  South Africa       52.98
```

DataFrame from CSV file

brics.csv

```
,country,capital,area,population
BR,Brazil,Brasilia,8.516,200.4
RU,Russia,Moscow,17.10,143.5
IN,India,New Delhi,3.286,1252
CH,China,Beijing,9.597,1357
SA,South Africa,Pretoria,1.221,52.98
```

CSV = comma-separated values

DataFrame from CSV file

brics.csv

```
,country,capital,area,population
BR,Brazil,Brasilia,8.516,200.4
RU,Russia,Moscow,17.10,143.5
IN,India,New Delhi,3.286,1252
CH,China,Beijing,9.597,1357
SA,South Africa,Pretoria,1.221,52.98
```

```
In [8]: brics = pd.read_csv("path/to/brics.csv")
In [9]: brics
Out[9]:
  Unnamed: 0       country    capital    area  population
0         BR        Brazil   Brasilia   8.516      200.40
1         RU        Russia     Moscow  17.100      143.50
2         IN         India  New Delhi   3.286     1252.00
3         CH         China    Beijing   9.597     1357.00
4         SA  South Africa   Pretoria   1.221       52.98

In [6]: brics = pd.read_csv("path/to/brics.csv", index_col = 0)
In [7]: brics
Out[7]:
        country  population      area    capital
BR       Brazil         200   8515767   Brasilia
RU       Russia         144  17098242     Moscow
IN        India        1252   3287590  New Delhi
CH        China        1357   9596961    Beijing
SA  South Africa          55   1221037   Pretoria
```

INTERMEDIATE PYTHON FOR DATA SCIENCE

Let's practice!

INTERMEDIATE PYTHON FOR DATA SCIENCE

Pandas, Part 2

brics

```
In [1]: import pandas as pd

In [2]: brics = pd.read_csv("path/to/brics.csv", index_col = 0)

In [3]: brics
Out[3]:
         country    capital    area  population
BR        Brazil   Brasilia   8.516      200.40
RU        Russia     Moscow  17.100      143.50
IN         India  New Delhi   3.286     1252.00
CH         China    Beijing   9.597     1357.00
SA  South Africa   Pretoria   1.221       52.98
```

Index and Select Data

- Square brackets
- Advanced methods
  - `loc`
  - `iloc`

Column Access [ ]

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

```
In [4]: brics["country"]
Out[4]:
BR           Brazil
RU           Russia
IN            India
CH            China
SA     South Africa
Name: country, dtype: object

In [5]: type(brics["country"])
Out[5]: pandas.core.series.Series
```

1D labelled array

Column Access [ ]

```
In [6]: brics[["country"]]
Out[6]:
         country
BR        Brazil
RU        Russia
IN         India
CH         China
SA  South Africa

In [7]: type(brics[["country"]])
Out[7]: pandas.core.frame.DataFrame
```

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

Column Access [ ]

```
In [8]: brics[["country", "capital"]]
Out[8]:
         country    capital
BR        Brazil   Brasilia
RU        Russia     Moscow
IN         India  New Delhi
CH         China    Beijing
SA  South Africa   Pretoria
```

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

Row Access [ ]

```
In [9]: brics[1:4]
Out[9]:
   country    capital    area  population
RU  Russia     Moscow  17.100       143.5
IN   India  New Delhi   3.286      1252.0
CH   China    Beijing   9.597      1357.0
```

indexes

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

0
1
2
3
4

Discussion `[ ]`

- Square brackets: limited functionality
- Ideally
  - 2D Numpy arrays
  - `my_array[ rows , columns ]`
- Pandas
  - `loc` (label-based)
  - `iloc` (integer position-based)

Row Access `loc`

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

```
In [10]: brics.loc["RU"]
Out[10]:
country       Russia
capital       Moscow
area            17.1
population     143.5
Name: RU, dtype: object
```

Row as Pandas Series

```
In [11]: brics.loc[["RU"]]
Out[11]:
   country capital  area  population
RU  Russia  Moscow  17.1       143.5
```

DataFrame

Row Access `loc`

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

```
In [10]: brics.loc["RU"]
Out[10]:
country       Russia
capital       Moscow
area            17.1
population     143.5
Name: RU, dtype: object

In [11]: brics.loc[["RU"]]
Out[11]:
   country capital  area  population
RU  Russia  Moscow  17.1       143.5

In [12]: brics.loc[["RU", "IN", "CH"]]
Out[12]:
   country    capital    area  population
RU  Russia     Moscow  17.100       143.5
IN   India  New Delhi   3.286      1252.0
CH   China    Beijing   9.597      1357.0
```

Row & Column `loc`

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

```
In [13]: brics.loc[["RU", "IN", "CH"], ["country", "capital"]]
Out[13]:
   country    capital
RU  Russia     Moscow
IN   India  New Delhi
CH   China    Beijing
```

Row & Column `loc`

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

```
In [13]: brics.loc[["RU", "IN", "CH"], ["country", "capital"]]
Out[13]:
   country    capital
RU  Russia     Moscow
IN   India  New Delhi
CH   China    Beijing

In [14]: brics.loc[:, ["country", "capital"]]
Out[14]:
         country    capital
BR        Brazil   Brasilia
RU        Russia     Moscow
IN         India  New Delhi
CH         China    Beijing
SA  South Africa   Pretoria
```

Recap

- Square brackets
  - Column access `brics[["country", "capital"]]`
  - Row access: only through slicing `brics[1:4]`
- `loc` (label-based)
  - Row access `brics.loc[["RU", "IN", "CH"]]`
  - Column access `brics.loc[:, ["country", "capital"]]`
  - Row & Column access `brics.loc[["RU", "IN", "CH"], ["country", "capital"]]`

Row Access iloc

```
In [15]: brics.loc[["RU"]]
Out[15]:
   country capital  area  population
RU  Russia  Moscow  17.1       143.5

In [16]: brics.iloc[[1]]
Out[16]:
   country capital  area  population
RU  Russia  Moscow  17.1       143.5
```

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

Row Access iloc

```
In [17]: brics.loc[["RU", "IN", "CH"]]
Out[17]:
   country    capital    area  population
RU  Russia     Moscow  17.100       143.5
IN   India  New Delhi   3.286      1252.0
CH   China    Beijing   9.597      1357.0

In [18]: brics.iloc[[1,2,3]]
Out[18]:
   country    capital    area  population
RU  Russia     Moscow  17.100       143.5
IN   India  New Delhi   3.286      1252.0
CH   China    Beijing   9.597      1357.0
```

	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

Row & Column iloc

	0	1	2	3
	country	capital	area	population
BR	Brazil	Brasilia	8.516	200.40
RU	Russia	Moscow	17.100	143.50
IN	India	New Delhi	3.286	1252.00
CH	China	Beijing	9.597	1357.00
SA	South Africa	Pretoria	1.221	52.98

```
In [19]: brics.loc[["RU", "IN", "CH"], ["country", "capital"]]
Out[19]:
   country    capital
RU  Russia     Moscow
IN   India  New Delhi
CH   China    Beijing

In [20]: brics.iloc[[1,2,3], [0, 1]]
Out[20]:
   country    capital
RU  Russia     Moscow
IN   India  New Delhi
CH   China    Beijing
```

Row & Column iloc

	0	1	2	3
	country	capital	area	population
0	BR	Brazil	Brasilia	8.516	200.40
1	RU	Russia	Moscow	17.100	143.50
2	IN	India	New Delhi	3.286	1252.00
3	CH	China	Beijing	9.597	1357.00
4	SA	South Africa	Pretoria	1.221	52.98

```
In [21]: brics.loc[:, ["country", "capital"]]
Out[21]:
         country    capital
BR        Brazil   Brasilia
RU        Russia     Moscow
IN         India  New Delhi
CH         China    Beijing
SA  South Africa   Pretoria

In [22]: brics.iloc[:, [0,1]]
Out[22]:
         country    capital
BR        Brazil   Brasilia
RU        Russia     Moscow
IN         India  New Delhi
CH         China    Beijing
SA  South Africa   Pretoria
```

INTERMEDIATE PYTHON FOR DATA SCIENCE

Let's practice!