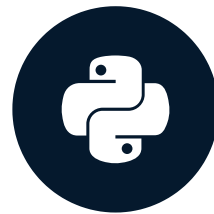


Dictionaries, Part 1

INTERMEDIATE PYTHON



Hugo Bowne-Anderson
Data Scientist at DataCamp

List

```
pop = [30.55, 2.77, 39.21]
countries = ["afghanistan", "albania", "algeria"]
ind_alb = countries.index("albania")
ind_alb
```

1

```
pop[ind_alb]
```

2.77

- Not convenient
- Not intuitive

Dictionary

```
pop = [30.55, 2.77, 39.21]  
countries = ["afghanistan", "albania", "algeria"]
```

```
...
```

```
{
```

```
}
```

Dictionary

```
pop = [30.55, 2.77, 39.21]
countries = ["afghanistan", "albania", "algeria"]

...

{"afghanistan":30.55,
```

Dictionary

```
pop = [30.55, 2.77, 39.21]
countries = ["afghanistan", "albania", "algeria"]

...

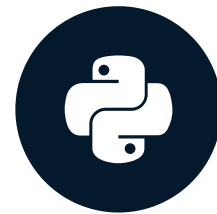
world = {"afghanistan":30.55, "albania":2.77, "algeria":39.21}
world["albania"]
```

2.77

Let's practice!
INTERMEDIATE PYTHON

Dictionaries, Part 2

INTERMEDIATE PYTHON



Hugo Bowne-Anderson
Data Scientist at DataCamp

Recap

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

```
2.77
```

```
world = {"afghanistan":30.55, "albania":2.77,  
         "algeria":39.21, "albania":2.81}  
world
```

```
{'afghanistan': 30.55, 'albania': 2.81, 'algeria': 39.21}
```


Recap

- Keys have to be "immutable" objects

```
{0:"hello", True:"dear", "two":"world"}
```

```
{0: 'hello', True: 'dear', 'two': 'world'}
```

```
{"just", "to", "test": "value"}
```

```
TypeError: unhashable type: 'list'
```

Principality of Sealand



¹ Source: Wikipedia

Dictionary

```
world["sealand"] = 0.000027  
world
```

```
{'afghanistan': 30.55, 'albania': 2.81,  
  'algeria': 39.21, 'sealand': 2.7e-05}
```

```
"sealand" in world
```

```
True
```

Dictionary

```
world["sealand"] = 0.000028
```

```
world
```

```
{'afghanistan': 30.55, 'albania': 2.81,  
  'algeria': 39.21, 'sealand': 2.8e-05}
```

```
del(world["sealand"])
```

```
world
```

```
{'afghanistan': 30.55, 'albania': 2.81, 'algeria': 39.21}
```

List vs. Dictionary

List vs. Dictionary

List	Dictionary
Select, update, and remove with <code>[]</code>	Select, update, and remove with <code>[]</code>

List vs. Dictionary

List	Dictionary
Select, update, and remove with <code>[]</code>	Select, update, and remove with <code>[]</code>

List vs. Dictionary

List	Dictionary
Select, update, and remove with <code>[]</code>	Select, update, and remove with <code>[]</code>
Indexed by range of numbers	

List vs. Dictionary

List	Dictionary
Select, update, and remove with <code>[]</code>	Select, update, and remove with <code>[]</code>
Indexed by range of numbers	Indexed by unique keys

List vs. Dictionary

List	Dictionary
Select, update, and remove with <code>[]</code>	Select, update, and remove with <code>[]</code>
Indexed by range of numbers	Indexed by unique keys
Collection of values — order matters, for selecting entire subsets	

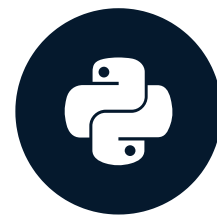
List vs. Dictionary

List	Dictionary
Select, update, and remove with <code>[]</code>	Select, update, and remove with <code>[]</code>
Indexed by range of numbers	Indexed by unique keys
Collection of values — order matters, for selecting entire subsets	Lookup table with unique keys

Let's practice!
INTERMEDIATE PYTHON

Pandas, Part 1

INTERMEDIATE PYTHON



Hugo Bowne-Anderson
Data Scientist at DataCamp

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

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

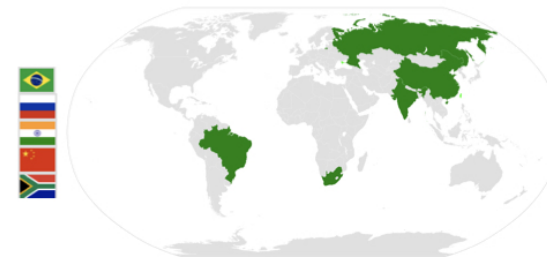
row = observations
column = variable

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	Pretoria	1.221	52.98



Datasets in Python

- 2D NumPy array?
 - One data type

Datasets in Python

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	Pretoria	1.221	52.98
		float	float

Datasets in Python

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	Pretoria	1.221	52.98
str	str	float	float

- pandas!
 - High level data manipulation tool
 - Wes McKinney
 - Built on NumPy
 - DataFrame

DataFrame

```
brics
```

	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 from Dictionary

```
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)

```
import pandas as pd  
brics = pd.DataFrame(dict)
```

DataFrame from Dictionary (2)

```
brics
```

```
   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
```

```
brics.index = ["BR", "RU", "IN", "CH", "SA"]
```

```
brics
```

```
   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
```

```
brics = pd.read_csv("path/to/brics.csv")
brics
```

```
   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
```


DataFrame from CSV file

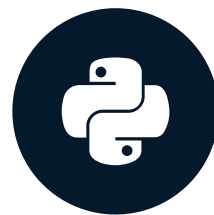
```
brics = pd.read_csv("path/to/brics.csv", index_col = 0)
brics
```

	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

Let's practice!
INTERMEDIATE PYTHON

Pandas, Part 2

INTERMEDIATE PYTHON



Hugo Bowne-Anderson
Data Scientist at DataCamp

brics

```
import pandas as pd  
brics = pd.read_csv("path/to/brics.csv", index_col = 0)  
brics
```

	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

```
brics["country"]
```

```
BR      Brazil
RU      Russia
IN      India
CH      China
SA      South Africa
Name: country, dtype: object
```

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

```
type(brics["country"])
```

```
pandas.core.series.Series
```

- 1D labelled array

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

```
brics[["country"]]
```

	country
BR	Brazil
RU	Russia
IN	India
CH	China
SA	South Africa

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

```
type(brics[["country"]])
```

```
pandas.core.frame.DataFrame
```

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

```
brics[["country", "capital"]]
```

	country	capital
BR	Brazil	Brasilia
RU	Russia	Moscow
IN	India	New Delhi
CH	China	Beijing
SA	South Africa	Pretoria

Row 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

```
brics[1:4]
```

	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 Access []

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

```
brics[1:4]
```

	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

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

```
brics.loc["RU"]
```

```
country      Russia
capital      Moscow
area          17.1
population    143.5
Name: RU, dtype: object
```

- Row as pandas Series

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

```
brics.loc[["RU"]]
```

	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

```
brics.loc[["RU", "IN", "CH"]]
```

	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

```
brics.loc[["RU", "IN", "CH"], ["country", "capital"]]
```

	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

```
brics.loc[:, ["country", "capital"]]
```

	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`

	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

```
brics.loc[["RU"]]
```

	country	capital	area	population
RU	Russia	Moscow	17.1	143.5

```
brics.iloc[[1]]
```

	country	capital	area	population
RU	Russia	Moscow	17.1	143.5

Row Access `iloc`

	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

```
brics.loc[["RU", "IN", "CH"]]
```

	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 Access `iloc`

	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

```
brics.iloc[[1,2,3]]
```

	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 iloc

	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

```
brics.loc[["RU", "IN", "CH"], ["country", "capital"]]
```

	country	capital
RU	Russia	Moscow
IN	India	New Delhi
CH	China	Beijing

Row & Column iloc

	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

```
brics.iloc[[1,2,3], [0, 1]]
```

	country	capital
RU	Russia	Moscow
IN	India	New Delhi
CH	China	Beijing

Row & Column iloc

	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

```
brics.loc[:, ["country", "capital"]]
```

	country	capital
BR	Brazil	Brasilia
RU	Russia	Moscow
IN	India	New Delhi
CH	China	Beijing
SA	South Africa	Pretoria

Row & Column iloc

	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

```
brics.iloc[:, [0,1]]
```

	country	capital
BR	Brazil	Brasilia
RU	Russia	Moscow
IN	India	New Delhi
CH	China	Beijing
SA	South Africa	Pretoria

Let's practice!
INTERMEDIATE PYTHON