

# Data structures & Control Flow

Sugarkhuu Radnaa

Py4Econ in Ulaanbaatar

*py4econ@gmail.com*

## Week 2: Learning objectives

Get to know:

- 1 Basic data types in Python
- 2 Data structures in **numpy** and **pandas**
- 3 **SQL** basics
- 4 **if, else and elif** conditions
- 5 **for and while** loop

# Data types in Python

- 1 String
- 2 Integer, Float,
- 3 Boolean
- 4 Datetime
- 5 List, tuple, set, dictionary

# Operations on data

- slice: `d[1:10]`
- update or add: `d[a] = b`
- remove or pop: `d[a] = []`
- join, concatenate: `f = [d,e]`

# Main data structures in Python

- Array (numpy)
- Dataframe (pandas)

# Numpy

Numpy and Scipy are extensively used for scientific computations and ML/AI algorithms

- random array and vector operations
- universal functions (sin, cos, max, argmax, ...)
- array handling (broadcast, tile, ravel, ...)
- matrix operations

# Pandas

Powerful dataframe package in Python

- Can read data from many sources (csv, excel, json, html, url)
- Can handle large size of data
- Can sync well with other packages

# Operations in Pandas

- 1 info, dtypes
- 2 head, tail
- 3 describe
- 4 `df['field'] = df.field`
- 5 loc, iloc
- 6 sort, filter
- 7 groupby, pivot table
- 8 type conversion
- 9 min, max, mean, sum
- 10 import/export dataframe from/to file (csv, json, xlsx, ...)



# SQL

SQL is the most popular database language. There are many different DBMS (database management system)s including SQL and NoSQL.

## Popular SQL databases:

- 1 Oracle SQL
- 2 Microsoft SQL
- 3 MySQL
- 4 PostgreSQL

## Popular NoSQL databases:

- 1 MongoDB
- 2 ElasticSearch
- 3 Casandra

# Sample database (SQL) design

Data architecture is a crucial in any organization. When due implemented, it can become a game changing factor. Efficient ETL, Datawarehouse and data centers are necessary for an efficient DBMS.



# SQL: W3SCHOOL examples

- ❶ `SELECT CustomerName, City FROM Customers;`
- ❷ `SELECT * FROM Customers WHERE Country='Mexico';`
- ❸ `SELECT * FROM Customers WHERE Country='Germany' AND (City='Berlin' OR City='München');`
- ❹ `UPDATE Customers SET ContactName = 'Alfred Schmidt', City= 'Frankfurt' WHERE CustomerID = 1;`
- ❺ `SELECT * FROM Customers WHERE ContactName LIKE 'a`
- ❻ `SELECT Customers.CustomerName, Orders.OrderID FROM Customers LEFT JOIN Orders ON Customers.CustomerID = Orders.CustomerID ORDER BY Customers.CustomerName;`
- ❼ `SELECT COUNT(CustomerID), Country FROM Customers GROUP BY Country ORDER BY COUNT(CustomerID) DESC;`

# Conditional statements

- if statement:
- else:
- elif (else if) statement:
- and, or, in, isinstance

# Loops (repeat an action)

- For - used only when we already knew the number of iterations
- While - used only when the number of iteration are not exactly known

Ways to exit loop:

- break
- continue

Use 'pass' when operation within the loop is not implemented yet

# Homework

- ① Task 1
  - ② Task 2
  - ③ Task 3
- Submit your result as a Github repository
  - Deadline: 1 week

**Note:** Create a github repo from the start and populate it with your results step by step.

# Task 1

- ① Add rows with arbitrary values to “data.xlsx” so that you have 50 rows in total
- ② Create a '.csv' file with data of only women older than 25 years old
  - ① filter by using only loops
  - ② filter by using conditions in pandas
- ③ Create a '.json' file with data of men under than 23 years old
  - ① filter by using only loops
  - ② filter by using conditions in pandas

## Task 2

- 1 What are the values in row 17 and column 2-5 of dataframe created from "data.xlsx"
- 2 What are the values row 25-28 and column 'firstName, age' of dataframe created from "data.xlsx"
- 3 Find the lowest, the highest and mean salary and age for men and women separately using 'groupby'
- 4 Find the lowest, the highest and mean salary and age for men and women separately using 'pivot table'



## Task 3

- 1 Populate a MySQL (or PostgreSQL) table by the data in “data.xlsx” (CREATE TABLE)
- 2 Select 'firstName' and 'lastName' of the first three rows ('LIMIT')
- 3 Select 'firstName' and 'age' of the last three rows ('ORDER BY, LIMIT')

**Tip:** Watch the following tutorial for MySQL (PostgreSQL)

- MySQL
- PostgreSQL

Thank you!