



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ



ФАКУЛЬТЕТ ГЕОГРАФИИ
И ГЕОИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Курс «Моделировании климата городов», лекция №3


Современные подходы к анализу данных


Михаил Иванович Варенцов





mvarentsov@hse.ru







Репозитории Git – что это и зачем?

Urban-climate-modelling4HSE / Practice / 

 mvarentsov PW1 first version

Name	Last commit message
 ..	
 .ipynb_checkpoints	PW1 first version
 PW1_data	PW1 first version
 PW1_example1.ipynb	PW1 first version

<https://github.com/mvarentsov/Urban-climate-modelling4HSE/tree/main/Practice>

 README  License  

License CC BY-NC-SA 4.0

BUCSS22

Course materials from the 2022 Bochum Urban Climate Summer School (Germany).

Context

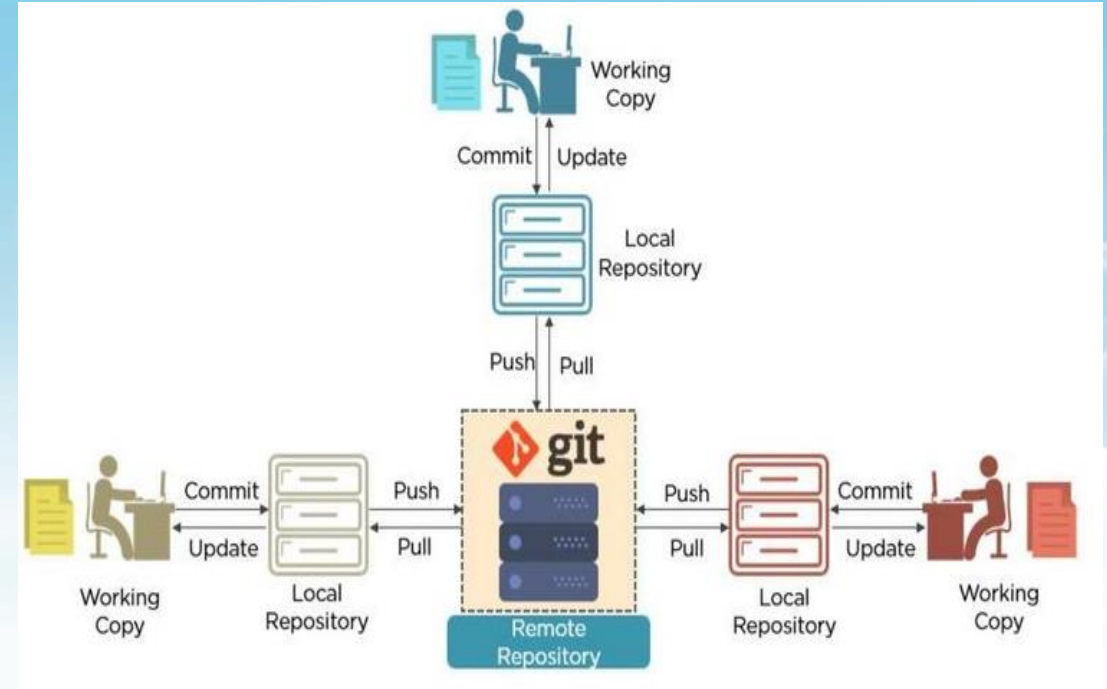
The first three editions of BUCSS took place in Bucharest (Romania) (see e.g. [here](#) for 2019). This 4th edition took place from 26 to 29 September 2022 at [Ruhr-University Bochum \(Germany\)](#), and aimed at 1) providing a general introduction to different facets of urban climatology with a special focus on urban climate informatics, 2) providing structured information and skill-building capabilities related to urban climate monitoring, remote sensing and modelling, and 3) strengthening an active pool of young scientists to tackle the major urban sustainability challenges of future generations.

Official website: <https://www.climate.ruhr-uni-bochum.de/bucss/>

<https://github.com/RUBclim/BUCSS22>

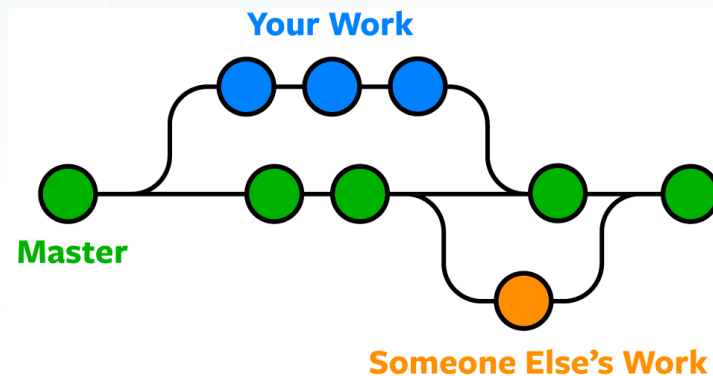
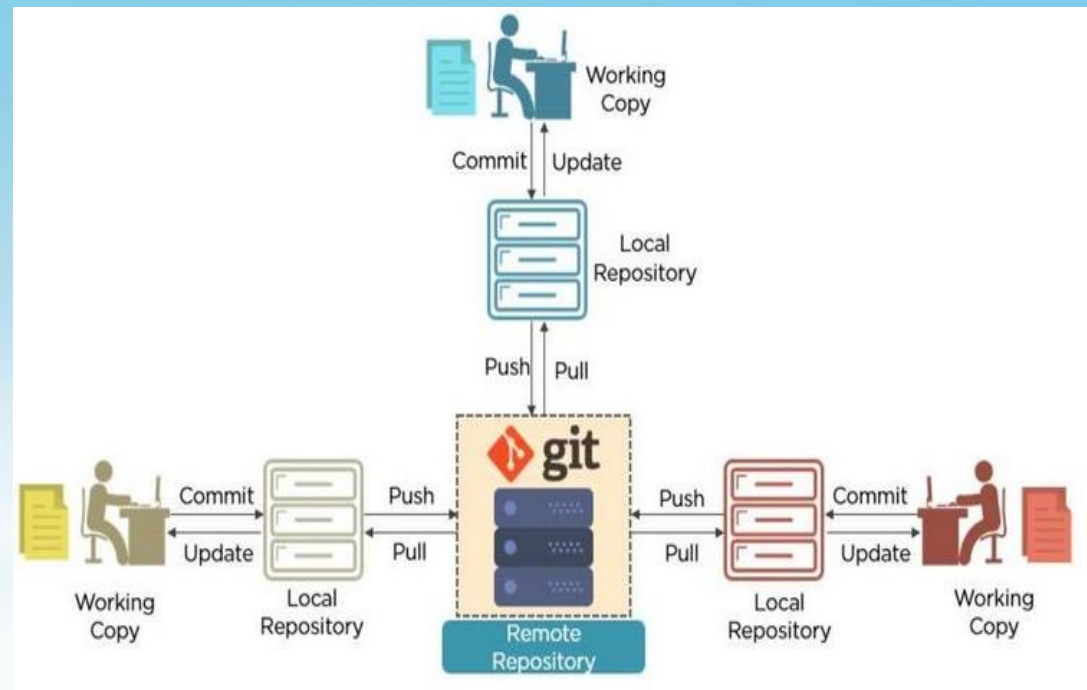
Репозитории Git – что это и зачем?

- ❑ Git - система управления версиями
- ❑ Придумал Линус Торвалдс при разработке ядра Линукс
- ❑ Современный стандарт для совместной разработки, в том числе в научной сфере и Data Science
- ❑ Задачи Git:
 - Синхронизация
 - Резервное копирование
 - Отслеживание и отмена изменений
 - Командная работа...



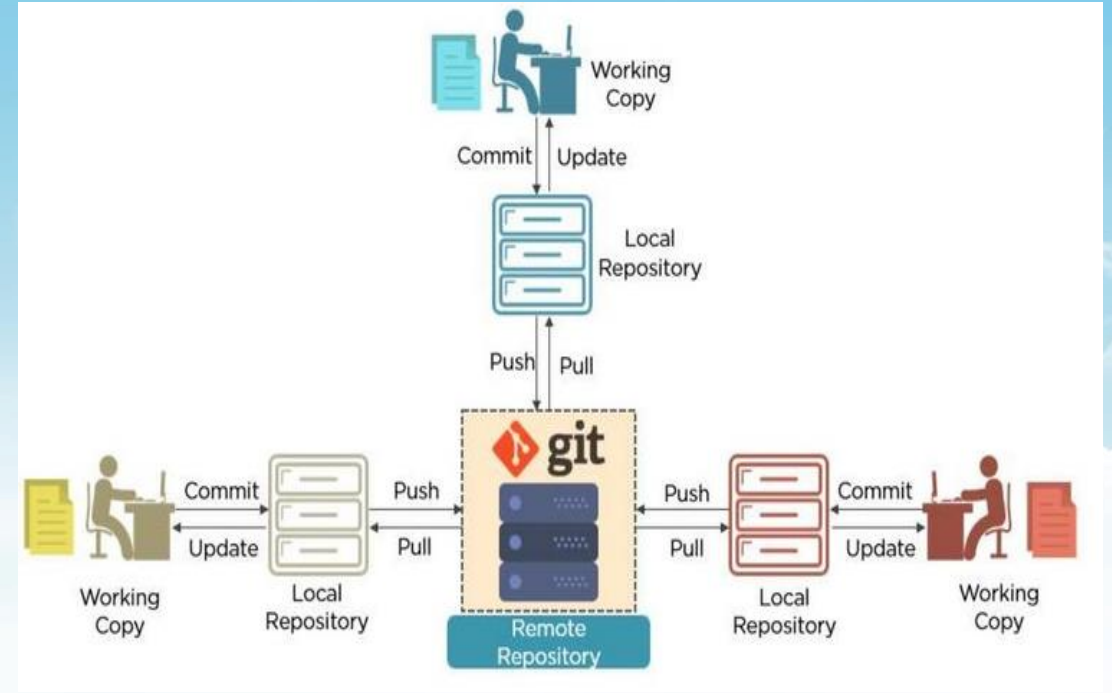
Основные понятия Git

- ❑ **Репозиторий** — место, где хранится код (или данные)
 - Локальный
 - Удаленный (remote)
- ❑ **Коммит (commit)** — зафиксированное и неизменяемое состояние репозитория. Чаще всего их создают, когда:
 - Создан новый функционал
 - Добавлен новый блок на верстке
 - Исправлены ошибки по коду
 - Завершен рабочий день
- ❑ **Ветка (branch)** - независимая последовательность коммитов в хронологическом порядке



Основные команды Git

- ❑ **git clone** – получения локальной копии существующего Git-репозитория
`git clone https://github.com/mvarentsov/Urban-climate-modelling4HSE.git`
- ❑ **git fetch** – загрузка содержимого из удаленного репозитория (без изменения локального репозитория)
- ❑ **git pull** = **Git fetch** + **Git merge** – загрузка содержимого из удаленного репозитория и обновление локального репозитория
- ❑ **git add** – добавить файл в список тех, которые отслеживаются для текущего КОММИТА
- ❑ **git commit** – зафиксировать текущие изменения
- ❑ **git push** – выгрузка содержимого локального репозитория в удаленный репозиторий.



```
MINGW64:/d/git_test/Urban-climate-modelling4HSE

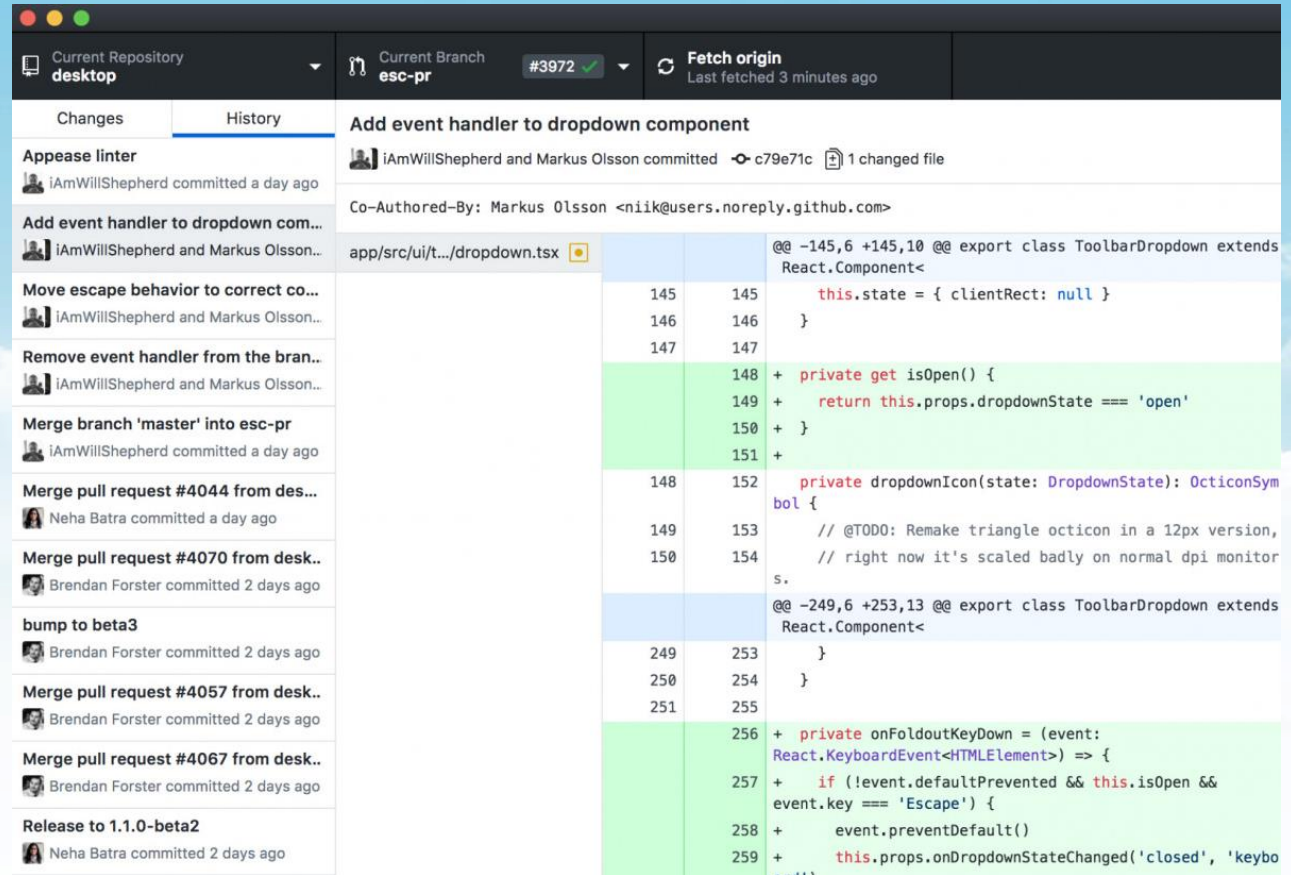
mvar9@DESKTOP-5FCDH8E MINGW64 /d/git_test/Urban-climate-modelling4HSE (main)
$ git fetch

mvar9@DESKTOP-5FCDH8E MINGW64 /d/git_test/Urban-climate-modelling4HSE (main)
$ git pull
Already up to date.

mvar9@DESKTOP-5FCDH8E MINGW64 /d/git_test/Urban-climate-modelling4HSE (main)
$ git clone https://github.com/mvarentsov/Urban-climate-modelling4HSE.git
```


Графические интерфейсы для Git

- ❑ **GitHub – облачный Git-сервер**
 - Альтернатива – создать собственный сервер, например на базе GitLab
- ❑ **GitHub Desktop – графический клиент для GitHub.**
- ❑ **Есть и альтернативы**
 - Sourcetree (Windows, macOS и Linux)
 - GitKraken (Windows, macOS)
 - ...



Python для анализа данных



Python: повторение

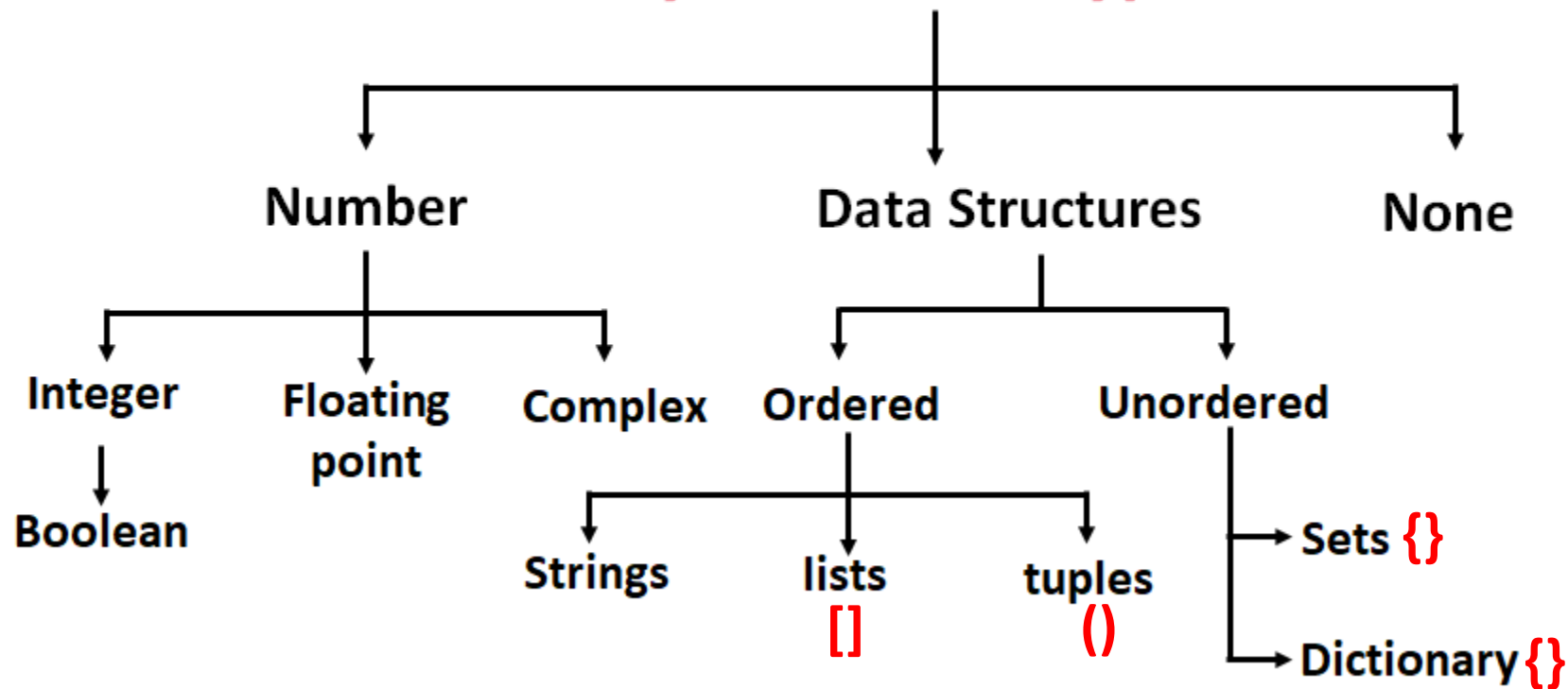
- ❑ Python – один из интерпретируемых языков программирования (наряду с R, Matlab, Julia и др.).
 - Исполняется ровно точно то, что написано
 - Исполняется построчно
 - Ошибки идентифицируются только в момент исполнение
- ❑ Среды исполнения: python (встроенная), ipython, Jupiter notebook, Google Colab и пр.
- ❑ Среды разработки: VS Code, PyCharm, Spyder, Jupiter Lab и пр.
 - Включают в себя поддержку работы с проектами подсветку кода, автозаполнение, отладку, интеграцию с git и прочие удобства
- ❑ Anaconda – один из дистрибутивов python, экосистема, ориентированная на научные вычисления



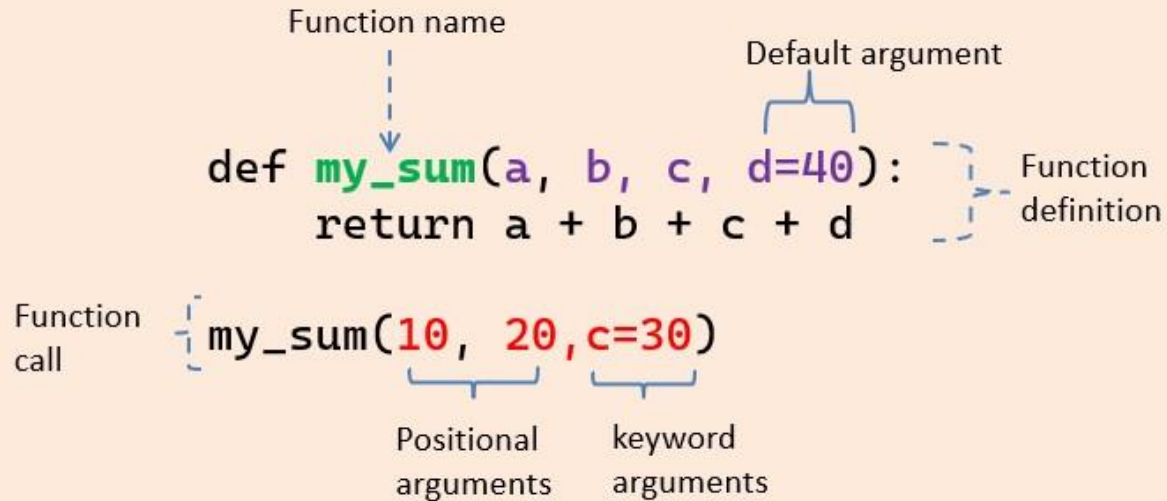
Базовые типы данных в Python

teachoo

Python Data Types



Функции и классы в Python



- **Positional argument** values get assigned as per the sequence. Now `a=10` and `b=20`
- **Keyword arguments** are those arguments where values get assigned to the arguments by their keyword
- **Default arguments:** Assign default values to the argument using the '=' operator at the time of function definition

```
class Student:  
  
    def __init__(self, name, marks):  
        self.name = name  
        self.marks = marks  
  
    def check_pass_fail(self):  
        if self.marks >= 40:  
            return True  
        else:  
            return False  
  
student1 = Student('Harry', 85)  
did_pass = student1.check_pass_fail()  
print(did_pass)  
  
student2 = Student('Janet', 30)  
did_pass = student2.check_pass_fail()  
print(did_pass)
```

Python recap



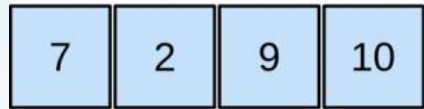
Библиотеки для работы с данными

- ☐ [NumPy](#) [intro] – работа с многомерными массивами, математические операции
- ☐ [Pandas](#) – работа с таблицами, в т.ч. дата-сетами для машинного обучения
- ☐ [Xarray](#) – работа с многомерными массивами данных, имеющих пространственно-временную привязку
- ☐ [Matplotlib](#) – графики на все случаи жизни с ручной настройкой
- ☐ [Seaborn](#) – продвинутые графики для статистического анализа



NumPy

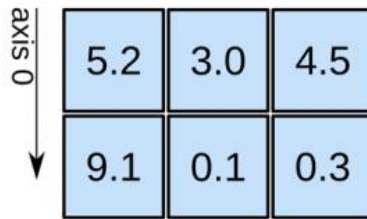
1D array



axis 0

shape: (4,)

2D array

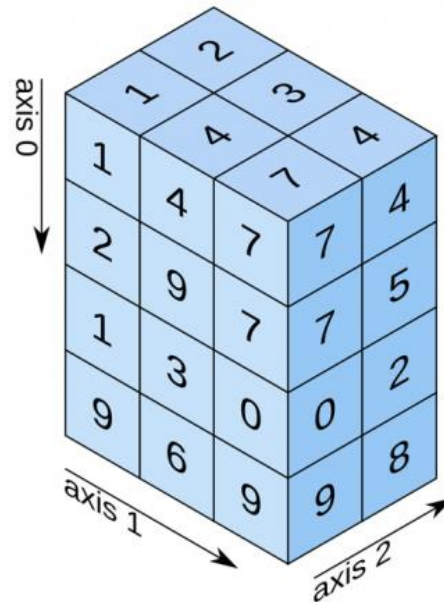


axis 0

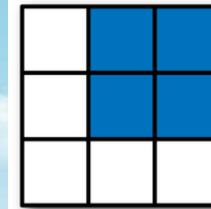
axis 1

shape: (2, 3)

3D array



shape: (4, 3, 2)

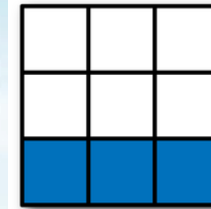


Expression

`arr[:2, 1:]`

Shape

(2, 2)



`arr[2]`

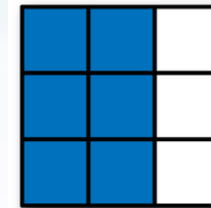
(3,)

`arr[2, :]`

(3,)

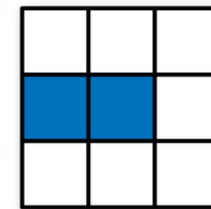
`arr[2:, :]`

(1, 3)



`arr[:, :2]`

(3, 2)



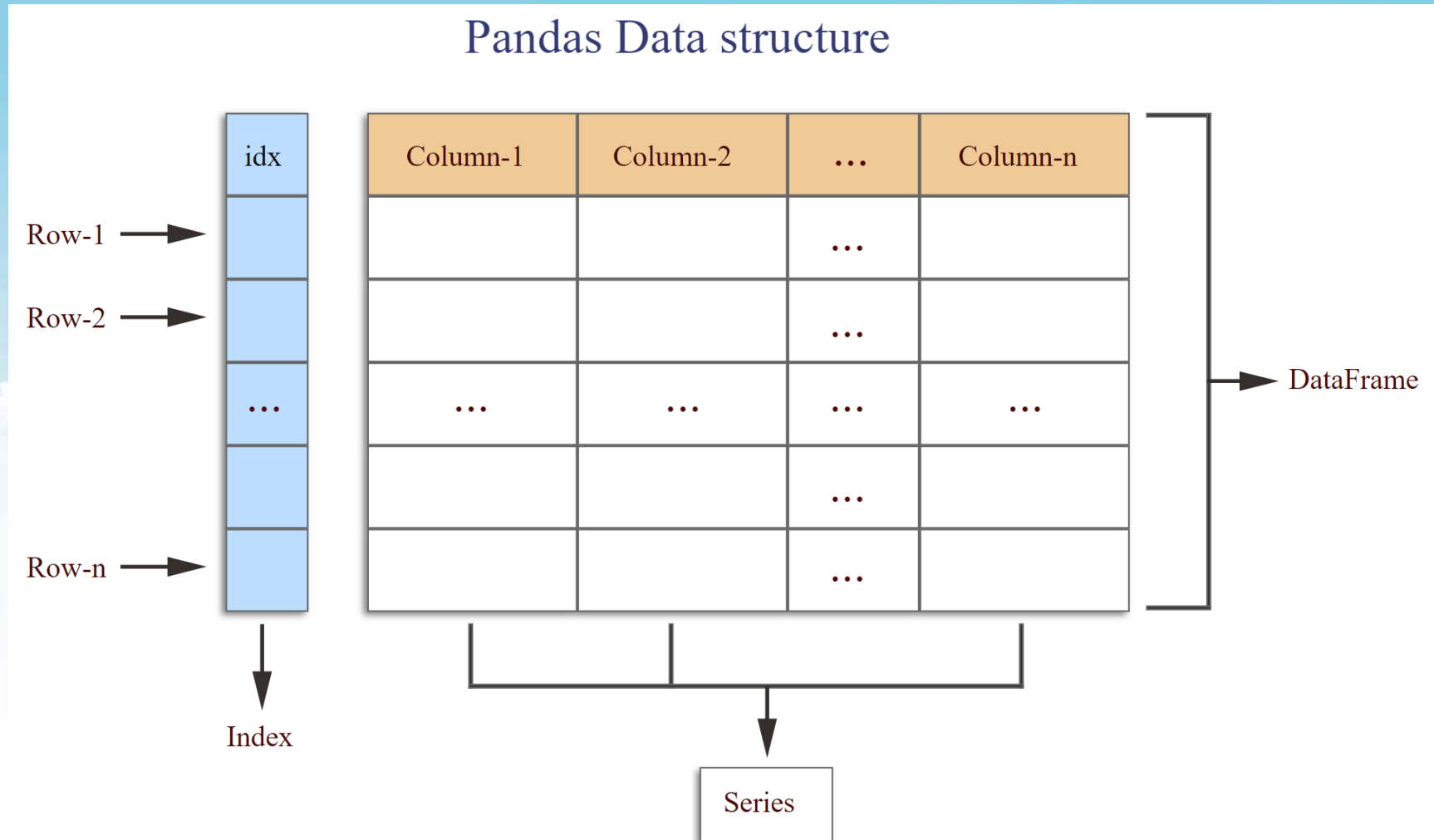
`arr[1, :2]`

(2,)

`arr[1:2, :2]`

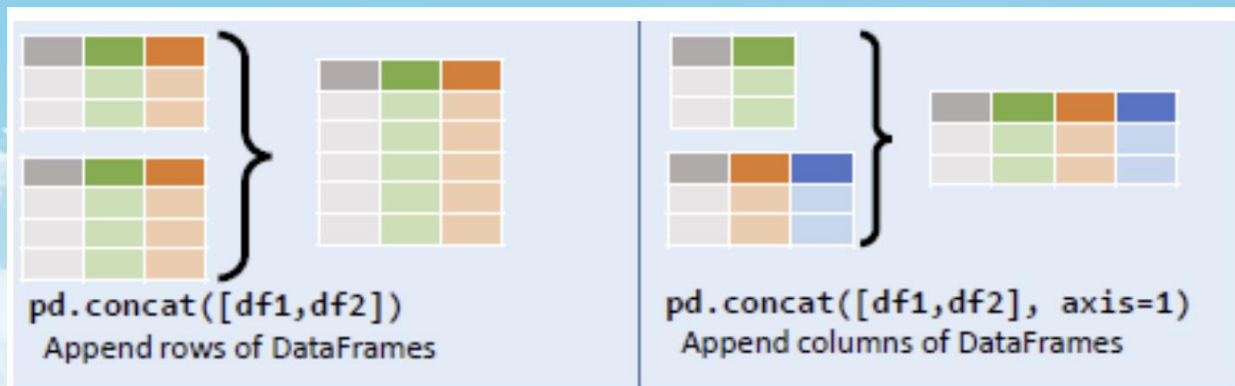
(1, 2)

Pandas

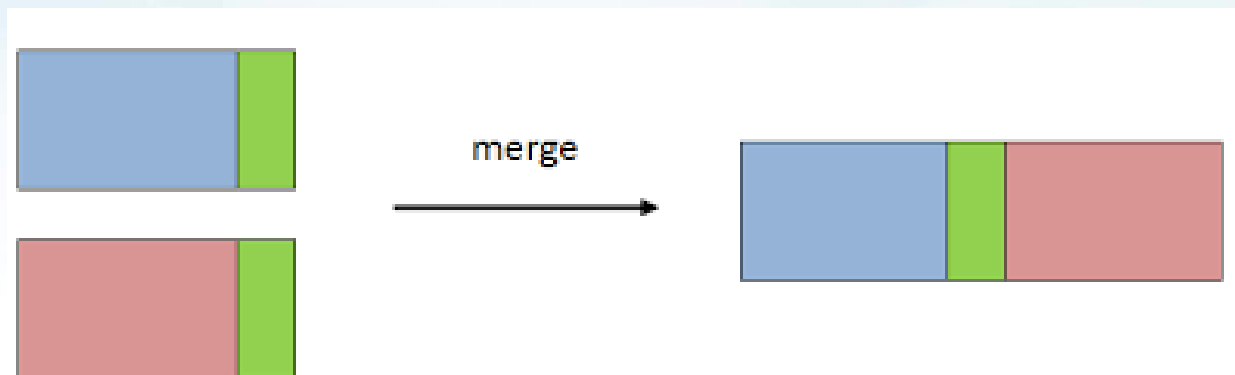


Конкатенация данных

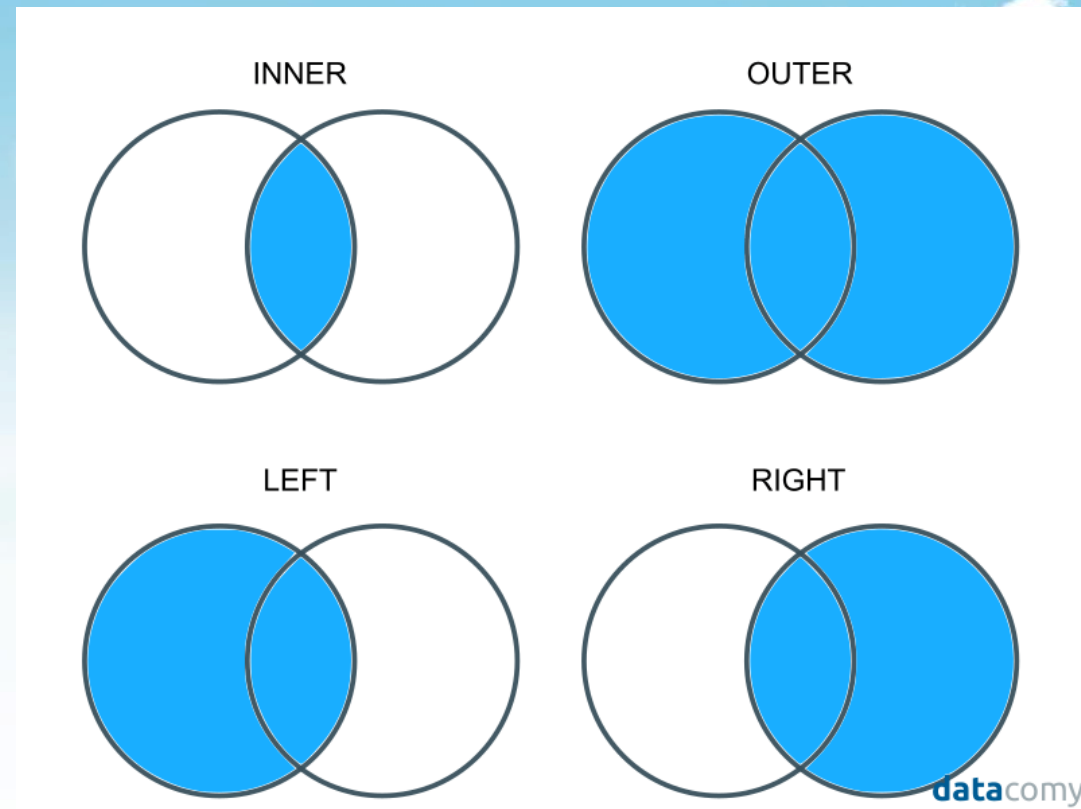
pd.concat



pd.merge



DataFrame.join



- Функция **join()** объединяет два кадра данных по индексу.
- Функция **merge()** объединяет два кадра данных по любому указанному вами столбцу.



The end

Вопросы?

Домашнее задание (эссе)

- ❑ С учетом опыта сегодняшнего мозгового штурма, необходимо написать эссе на эту же тему («Как знания о городском климате можно использовать для решения проблем населения и экономики?»)
- ❑ Отличия от сегодняшнего задания:
 - Необходимо использовать материалы из двух следующих лекций («Мониторинг городского климата», «Моделирование городского климата»).
 - Необходимо использовать ссылки на российскую и зарубежную литературу
- ❑ Ориентировочный объем эссе ≈5 стр.
- ❑ Использование генеративных нейросетей допускается, но должно быть явно указано
- ❑ Срок сдачи – через три недели (12 февраля)

Практическая работа №1

Часть 1

- ☐ Исследовать доступность данных стационарных наблюдений для выбранного города и его окрестностей (метеостанции, аэропорты, специализированные сети*)
- ☐ Выбрать репрезентативные городские и загородные станции
 - Если таких станций нет → выбрать другой город
- ☐ Определить точное местоположение этих станций
- ☐ Скачать данные за несколько лет
- ☐ Построить графики временной динамики основных метеовеличин (температуры, влажности, скорости ветра) ветра на городских и загородных станциях за месяцы с контрастными метеоусловиями, например, для лета/зимы
- ☐ Проанализировать городские аномалии температуры и влажности(разности «город-фон»):
 - Построить графики временной динамики за отдельные периоды (≈15-30 дней)
 - Построить график осредненного суточного хода
 - Построить график осредненного сезонного хода

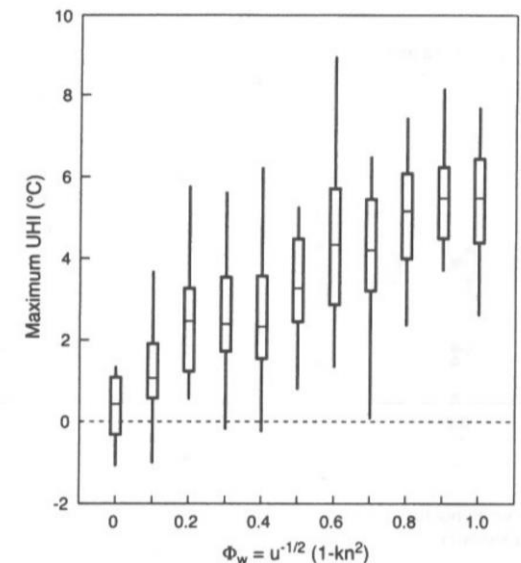
Практическая работа №1

Часть 2

- ❑ Добавить в список анализируемых метеостанций хотя бы одну фоновую или городскую станцию (чтобы хотя бы одна выборка включала не менее двух станций)
- ❑ В рамках анализа данных из предыдущей части задания использовать средние значения по выборке городских/фоновых станций
- ❑ Рассчитать по данным наблюдений на загородных станциях эмпирический «фактор погоды»
- ❑ Исследовать статистическую связь между «фактором погоды» и разностью температуры «город-фон»

In addition, we use a so-called weather factor (WF) as a predictor for ΔT , an empirical function, of wind speed, cloud fraction and cloud type, as suggested in [58] and explained in more detail in [59]. We use a slightly simplified formulation based on 10-m wind speed (vel10m), total (tcc), and low (lcc) cloud fractions, which allows WF to be calculated without information about cloud type and is applicable both for observations and reanalysis.

$$F_w = \left[1 - 0.8 \cdot tcc^2 - 0.4 \cdot (tcc - lcc)^2 \right] \cdot \min\left(vel10m^{-\frac{1}{2}}; 1\right) \quad (3)$$



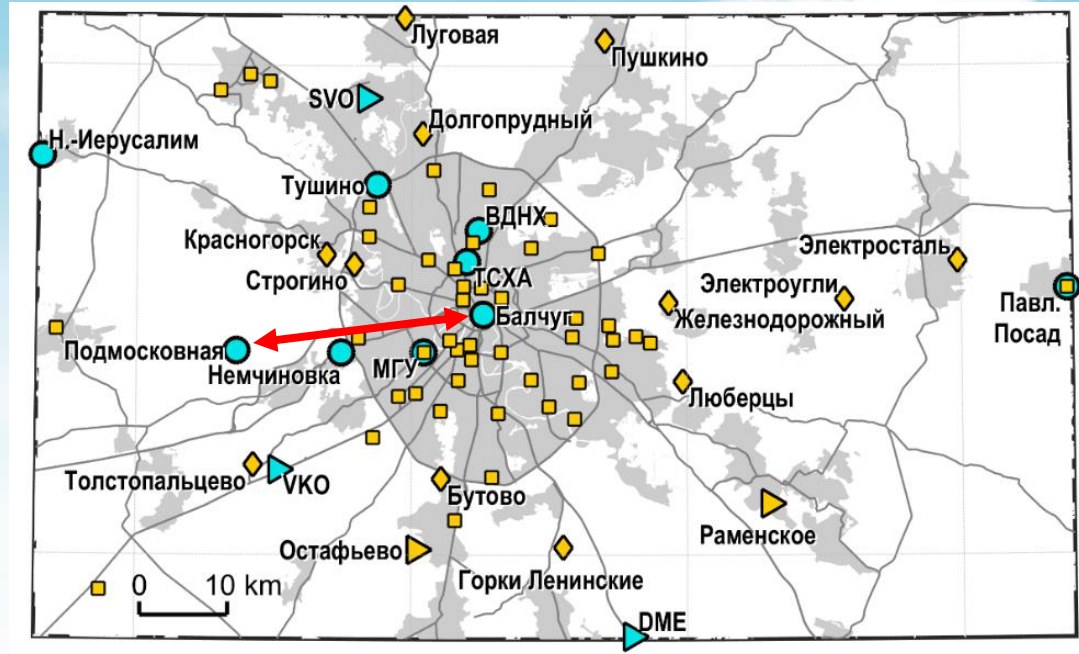
Практическая работа №1

Тренируемся на данных для Москвы

Urban-climate-modelling4HSE / Practice /

mvarentsov PW1 first version

Name	Last commit message
..	
.ipynb_checkpoints	PW1 first version
PW1_data	PW1 first version
PW1_example1.ipynb	PW1 first version



<https://github.com/mvarentsov/Urban-climate-modelling4HSE/tree/main/Practice>