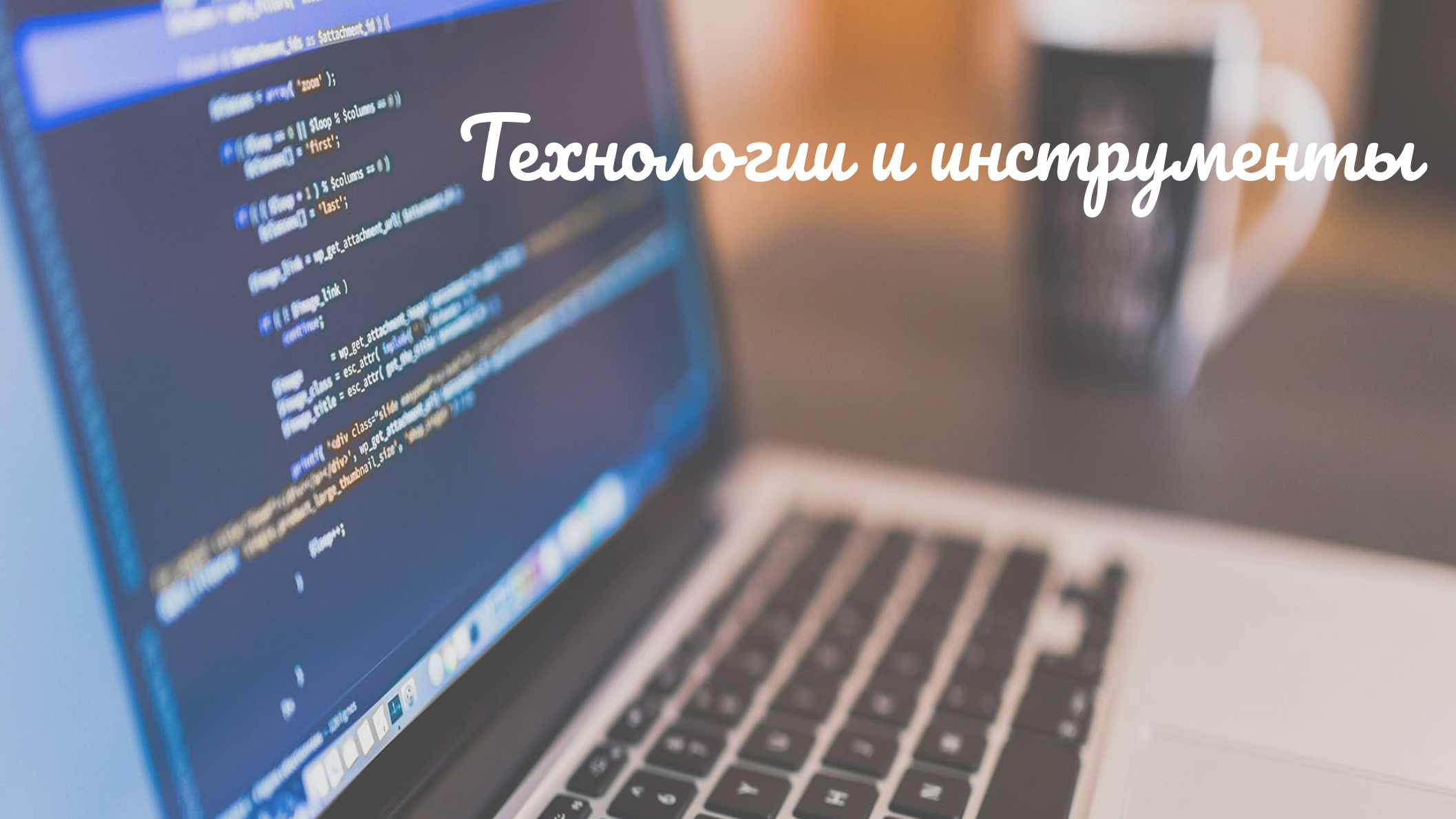


# Технологии и инструменты



# Немного спойлеров...

- **Anaconda**
- **VirtualEnv and CondaEnv**
- **Jupyter notebook**
- **Google Colab**
- **PyCharm**



# Python — интерпретируемый язык программирования

Active Python - <https://www.activestate.com/products/python/downloads/>

Anaconda - <https://www.anaconda.com/products/individual>



By data scientists, for  
data scientists



# Anaconda

Это самая популярная Data Science платформа. А чем она полезна?

Не нужно устанавливать библиотеки и интерпретатор и разбираться с возможными проблемами — просто устанавливаем Анаконду и вот у нас уже есть почти всё необходимое.



# Virtual environments

*Что это?*

Изолированные среды для разных проектов/задач.

*Для чего это?*

Избежать конфликта разных версий библиотек, облегчить перенос проектов между разными машинами.

*Как использовать?*

virtualenv, anaconda (рекомендую второй вариант)

*Ссылки:*

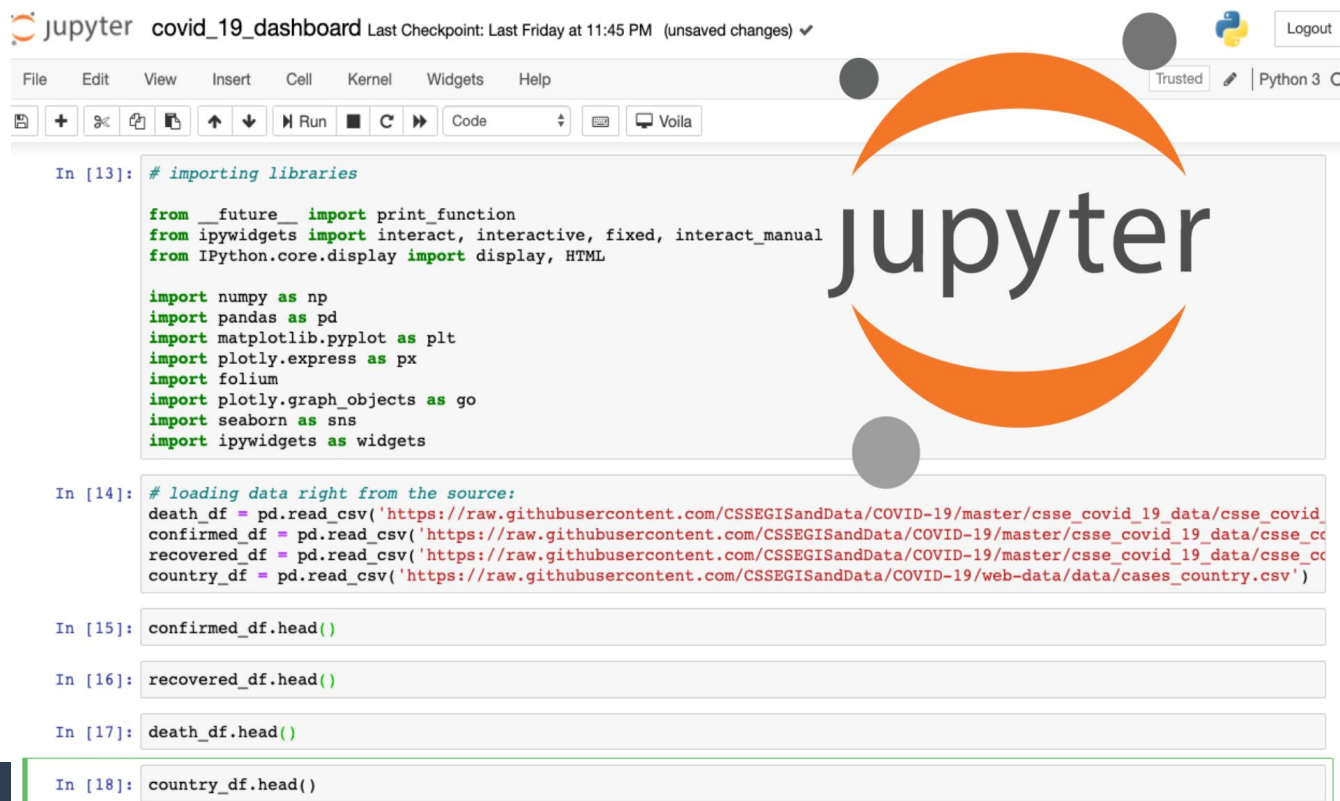
- в anaconda <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
- virtualenv <https://python-scripts.com/virtualenv>

*Совет: Для каждого проекта – своё виртуальное окружение.*



# Jupyter Notebook

- Этот инструмент, в частности, позволяет выполнять код частями в определенной последовательности.



The screenshot displays a Jupyter Notebook titled 'covid\_19\_dashboard'. The interface includes a top bar with the Jupyter logo, the notebook title, and a 'Last Checkpoint' timestamp. Below this is a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running code, and other functions. The notebook contains several code cells, each starting with 'In [n]:'. The first cell imports various libraries like pandas, matplotlib, and folium. The second cell loads data from GitHub URLs into DataFrames. The subsequent cells use the .head() method to inspect the first few rows of each DataFrame. A large, semi-transparent Jupyter logo is overlaid on the right side of the notebook interface.

```
In [13]: # importing libraries

from __future__ import print_function
from ipywidgets import interact, interactive, fixed, interact_manual
from IPython.core.display import display, HTML

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import folium
import plotly.graph_objects as go
import seaborn as sns
import ipywidgets as widgets

In [14]: # loading data right from the source:
death_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/death.csv')
confirmed_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/confirmed.csv')
recovered_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_data/recovered.csv')
country_df = pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/web-data/data/cases_country.csv')

In [15]: confirmed_df.head()

In [16]: recovered_df.head()

In [17]: death_df.head()

In [18]: country_df.head()
```

# Google Colaboratory

## Плюсы:

- ✓ Не нужно иметь мощный компьютер — серверы гугла вам в помощь;
- ✓ Можно запустить код без проблем с установкой библиотек;
- ✓ Это бесплатно, нужен только гугл-аккаунт.

## Минусы:

- Ограниченное количество памяти и времени;
- Нужен устойчивый доступ к Интернету;
- Неудобства в работе с файлами.



# IDE PyCharm

## Плюсы:

- ✓ Терминал, файловый менеджер, редактор кода/таблиц и пр. в одном месте;
- ✓ Встроенная система контроля версий;
- ✓ Удобная интеграция с Git и Github;
- ✓ Предупредит об ошибке, подскажет как сделать код лучше, согласно общепринятым стандартам;
- ✓ Возможность удобной отладки кода.

## Минусы:

- ▣ Не удобен для исполнения кода по частям.

*Совет: после скачивание Анаконды, выберите и скачайте PyCharm with Anaconda plugin.*

<https://www.jetbrains.com/pycharm/download/#section=windows>

