

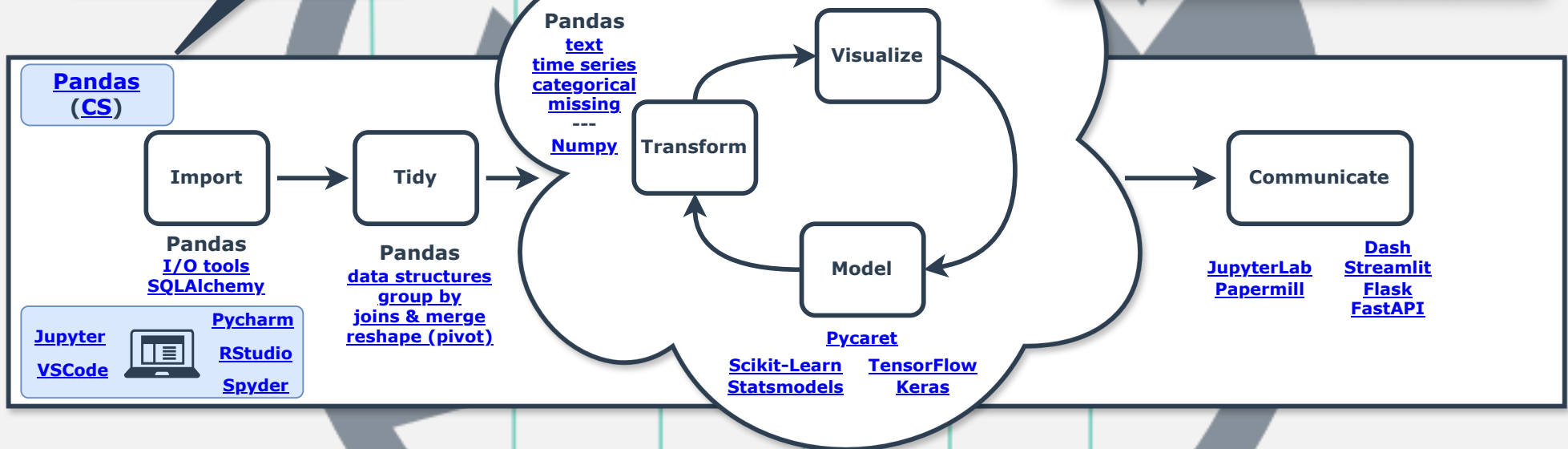
# Data Science with Python Workflow

If you want to learn Python, then join our course: [Python for Data Science Automation \(DS4B 101-P\)](#).



CS = Cheat Sheet

Click the links for Documentation



## Important Resources

- **Anaconda Distribution:** <https://www.anaconda.com/download/>
- **Python Documentation:** <https://docs.python.org/>
- **Python Standard Library:** <https://docs.python.org/3/library>

Join the [Python for Data Science Automation Course](#)



Business Science University  
[university.business-science.io](https://university.business-science.io)

# Data Science



## Special Topics

### Time Series Forecasting

- [Nixtla](#) - TimeGPT, StatsForecast, MLForecast, NeuralForecast, Hierarchical Forecast
- [sktime](#) - Scikit-Learn Extension for Time Series
- [statsmodels](#) - Time Series Analysis
- [GluonTS](#) - MXNet/Gluon Deep Learning for Time Series

### Time Series Analysis

- [PyTimetk](#) - Time series analysis in python
- [TSFresh](#) - Time Series Feature Engineering
- [tslearn](#) - Time Series Features
- [Pandas](#) Time Series
- [Arrow](#) - Human-Friendly Time

### EDA

- [pandas-profiling](#), [SweetViz](#), [lux](#)

### Web

- [beautifulsoup](#) - Extract data from HTML
- [requests-html](#) - HTML Parsing
- [scrapy](#) - Web crawling

### MS Office & PDF

- [XlsxWriter](#) - Create Excel Workbooks
- [pyexcel](#) - Read/Write Excel
- [xlwings](#) - Call python from Excel
- [python-docx](#) - Word Documents
- [python-pptx](#) - PowerPoint Documents
- [pdfminer](#) - Text extraction from PDF
- [textract](#) - Extract text from any document
- [PyPDF2](#) - Create PDF documents
- [gsread](#) - Google Sheets

### Text Analysis & NLP

- [NLTK](#) - Text Tokenization & Modeling
- [spaCy](#) - NLP using Cython for Speed
- [fuzzywuzzy](#) - Fuzzy String Matching

### Recommendation Systems

- [Annoy](#) - Approximate Nearest Neighbors
- [LightFM](#) - Popular recommendation algo's.

### Apps & APIs

- [FastAPI](#) - Web framework for building APIs in Python
- [Flask](#) - Web Development
- [Dash](#) & [Streamlit](#) - DS Web Frameworks

### MLOps

- [Pycaret MLFlow Integration](#)
- [MLFlow](#) - Machine Learning Lifecycle, Tracking, Deployment
- [MetaFlow](#) - Scalable AWS Jobs for Data Scientists

### Cloud

- [boto3](#) (AWS) - AWS Python SDK
- [Google Cloud](#) - GCP Python SDK
- [Azure](#) - Azure Python SDK

### ETL & Automations

- [Airflow](#) - Workflow Scheduling & Monitoring
- [Luigi](#) - Batch Job Tool, Scheduling, Monitoring
- [Ansible](#) - Deployment Automation
- [JobLib](#) - Run python jobs

### Machine Learning

- [Scikit-Learn](#) - ML in Python
- [H2O](#) - Scalable & AutoML
- [TPOT](#) - TPOT Automated ML Tool
- [PyCaret](#) - PyCaret Low Code ML
- [Dask ML](#) - Scalable ML with Dask
- ML Packages: [XGBoost](#), [LightGBM](#), [CatBoost](#)

### Feature Engineering

- [Sklearn Data Transformations](#)
- [sklearn-pandas](#) - Sklearn Extension for Pandas
- [Featuretools](#) - Automated Feature Engineering
- [category\\_encoders](#) - Categorical Encoding
- [imbalanced-learn](#) - Resampling for Imbalanced
- [fancyimpute](#) - Extended imputation strategies

### Deep Learning

- [TensorFlow](#) & [Keras](#)
- [PyTorch](#)
- [MXNet](#), [Gluon](#), & [GluonTS](#)
- [OpenAI Gym](#) - Reinforcement Learning

### Image & Comp Vision

- [OpenCV](#) - Open Source Computer Vision
- [Scikit Image](#) - Image Processing
- [Pillow](#) - Python Imaging Library

### Speed & Scale

- [datatable](#) - C++ Speed Up
- [Dask \(CS\)](#) - Parallel Pandas & Scikit Learn
- [RAPIDS \(CS\)](#) - GPU Accelerated Pandas
- [PySpark](#) - Spark Clusters
- [Optimus](#) - PySpark Extension for Humans

### Coming from R?

- [R-to-Pandas Comparison](#)
- [siuba](#) & [plydata](#) - dplyr/tidyr ports
- [datatable](#) - data.table port
- [plotnine](#) - ggplot2 port





### AI LLM Frameworks

Frameworks for Large Language Models (LLMs)

- [LangChain](#) - Application development framework for apps powered by LLMs with many integrations, tools, and community extensions, support for a broad spectrum of Agents and more.
- [LangGraph](#) - Build DAG Graphs to combine multiple LLMs and Agents.
- [LLamaIndex](#) - An alternative to LangChain that focuses on RAG (Retrieval Augmented Generation) and Vector Indexing and Retrieval.

### Vector Databases

Used to store text embedding and similarity search

- [ChromaDB](#) - Open source vector DB
- [FAISS](#) - Facebook AI Similarity Search
- [Pinecone](#) - Scalable, cloud-based vector DB
- [Milvus](#) - Scalable cloud-based vector DB
- [Zilliz](#) - Fully managed cloud built on Milvus

[More vector databases](#)

### Embedding Models

Text Embeddings Models:

- [OpenAI Embedding](#)
- [Hugging Face Transformers Library](#)

[More Embedding Models](#)

### LLM Models & APIs

LLM Inference Models, APIs, and SDKs:

- [Hugging Face Models](#) - Massive library of open-source models for data science, machine learning, and AI.
- [OpenAI Python SDK and API](#) - A software development kit for interfacing with OpenAI API.
- [Anthropic Claude SDK and API](#) - A software development kit for interfacing with Anthropic API and Claude models
- [Meta Llama Models](#) - Open source LLMs by Facebook / Meta
- [Ollama](#) - Run open-source LLMs such as Llama 2 and 3 locally
- [Groq](#) - Blazing fast inference
- [Mistral AI](#) - Open source and commercial LLM models

[More LLMs](#)

### Document Loaders

LangChain Native Document Loaders:

- [PDF](#)
- [CSV](#)
- [HTML](#)
- [JSON](#)
- [MS Office \(Excel, PowerPoint, Word\)](#)

[More Native Document Loaders](#)

[LangChain 3rd Party Document Loaders](#)

### Text Splitters

LangChain Native Document Transformers:

- [Character Splitter](#)
- [Recursive Character Splitter](#)
- [HTML Header](#)
- [HTML Section](#)

[More Splitters](#)

### Output Parsers

LangChain Built-In Output Parsers:

- [CSV](#)
- [Datetime](#)
- [Pandas DataFrame](#)

[More Output Parsers](#)

### Tools & Toolkits

Integrations and Community Tools:

- [Pandas DataFrame](#)
- [SQL Databases](#)
- [Python REPL Tool](#)
- [Spark Dataframe](#)
- [Spark SQL](#)

[More Tools](#)

[More Toolkits](#)

### Agents

Building agents with Tools:

- [Agent Concepts](#)
- [Agent Types](#)
- [Custom Agents](#)

[More on Agents](#)