Computational NeuroEthology

Introduction

EINSTEIN

Albert Einstein College of Medicine

Class 1

September 29th, 2025

Mikhail Kislin

Roland Ferger

Overview

Introduction to programming with Python and LLMs

Statistical approaches

Computational tools for studying behavior

Modeling Neural Imaging and Electrophysiological Recording Data

Linking behavior to neural activity through data and modeling

Final projects

Aims

- Apply computational and machine learning tools to study animal behavior.
- Use open-source packages for data analysis.
- Critically evaluate research data and design analysis pipelines
- Work collaboratively on group projects while demonstrating individual accountability.

Final projects

- Teamwork: Form groups of 3 students (up to 5 max)
- Dataset: Choose from provided examples or explore the DANDI Archive
  https://dandiarchive.org/
- Hands-on Assignment
  Starting from Class 2, submit short weekly project tasks.
  Due Friday at noon via Slack (before the next class).

What is behavior?

"The total movements made by the intact animal."
Tinbergen (1955)

Niko Tinbergen's 4 Types of questions

focuses on immediate causes

focuses on evolutionary forces that have shaped a trait over time

Mechanism (Causation)

Function (Survival Value)

Development (Ontogeny)

Evolution (Phylogeny)

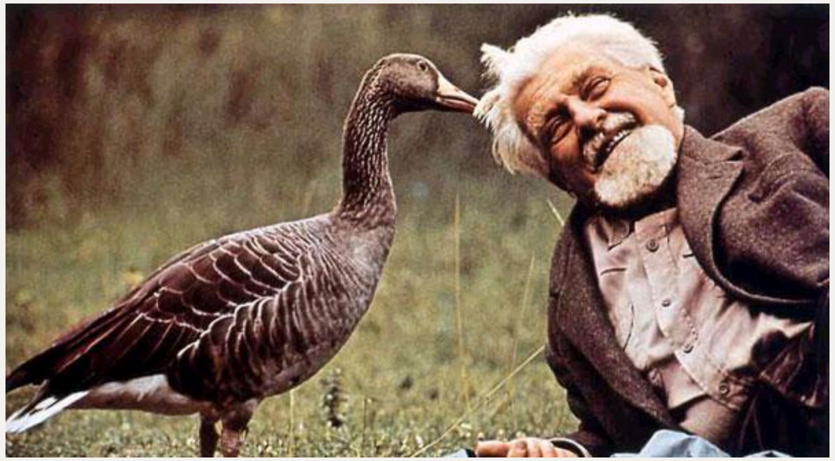
Proximate

Ultimate

1. Mechanism (Causation): What triggers the behavior? What processes control it?
2. Development (Ontogeny): How does behavior change with age or development?
3. Function (Survival Value): How does behavior improve survival or reproduction?
4. Evolution (Phylogeny): How did this behavior evolve across species?

On the Aims and Methods of Ethology (Tinbergen,1963)

Three foundations of animal behavior



1. The force of natural selection

2. The ability of animals to learn (individual learning)

3. The power of transmitting learned information to others (cultural transmission)

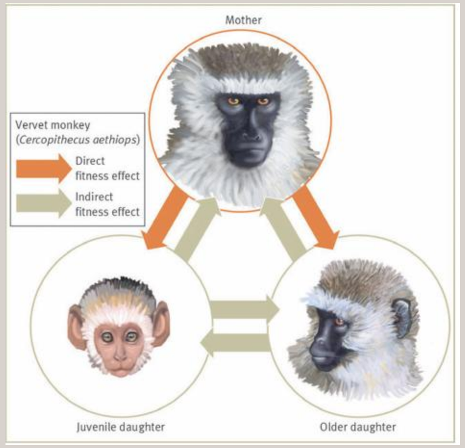
Three Approaches to Ethology

Empirical

- Observational
(Naturalistic study of behavior)

- Experimental
(Testing hypotheses under controlled conditions)

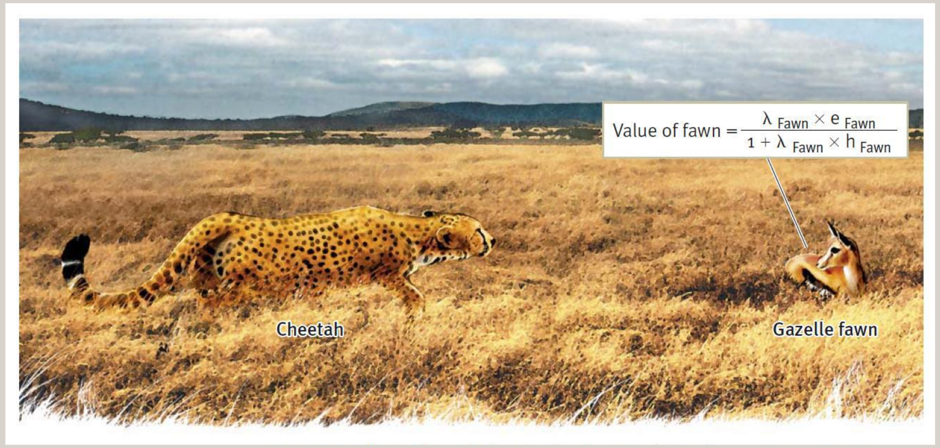
Conceptual



- Integrates ideas into frameworks

- Generates new experiments and reshapes perspectives

Theoretical

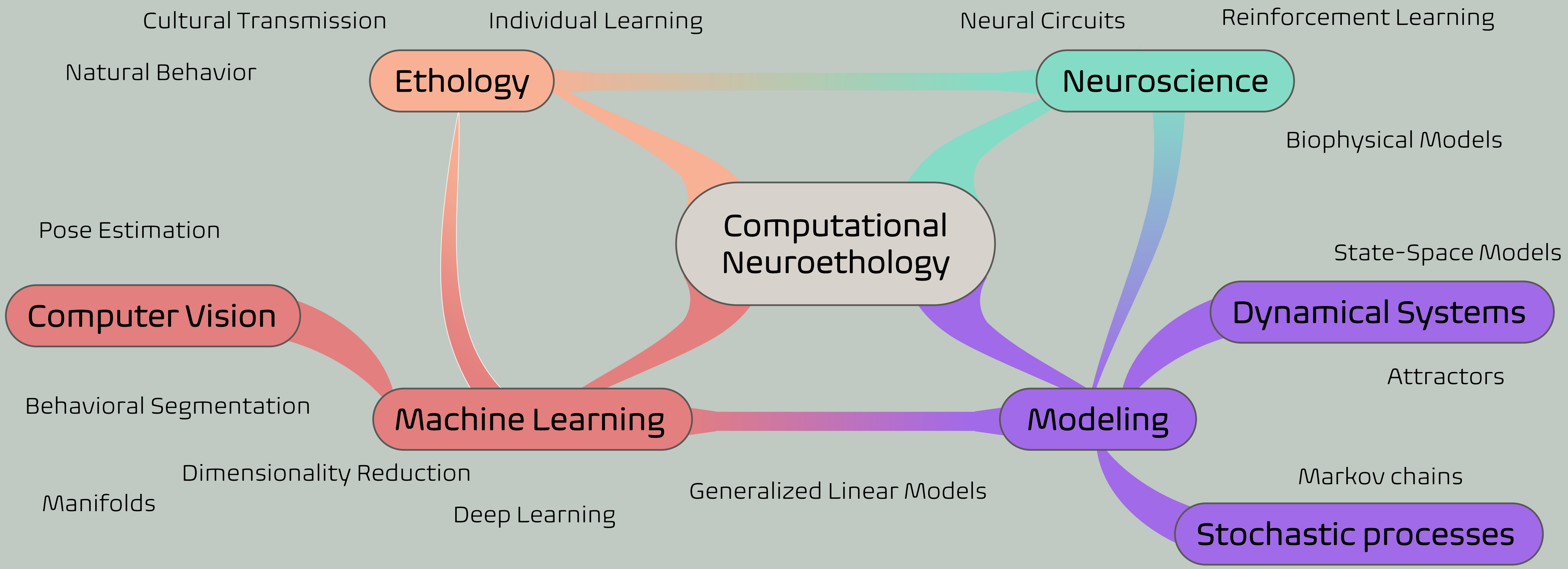


Value of fawn = $\frac{\lambda_{Fawn} \times e_{Fawn}}{1 + \lambda_{Fawn} \times h_{Fawn}}$
Cheetah
Gazelle fawn

- Explain behavior with mathematical models

- Predict outcomes of behavioral strategies

Which parameters of behavior are under neural control?



02 What is Programming and why to use it?

Programming is telling a computer what to do.

Programming teaches you to break complex problems into smaller, manageable parts.

It will primarily benefit you by helping automate tasks and save your time.

Powerful tool to address research questions

Programming Languages

A programming language is how we write code, what keywords we use, to tell the computer what to do.

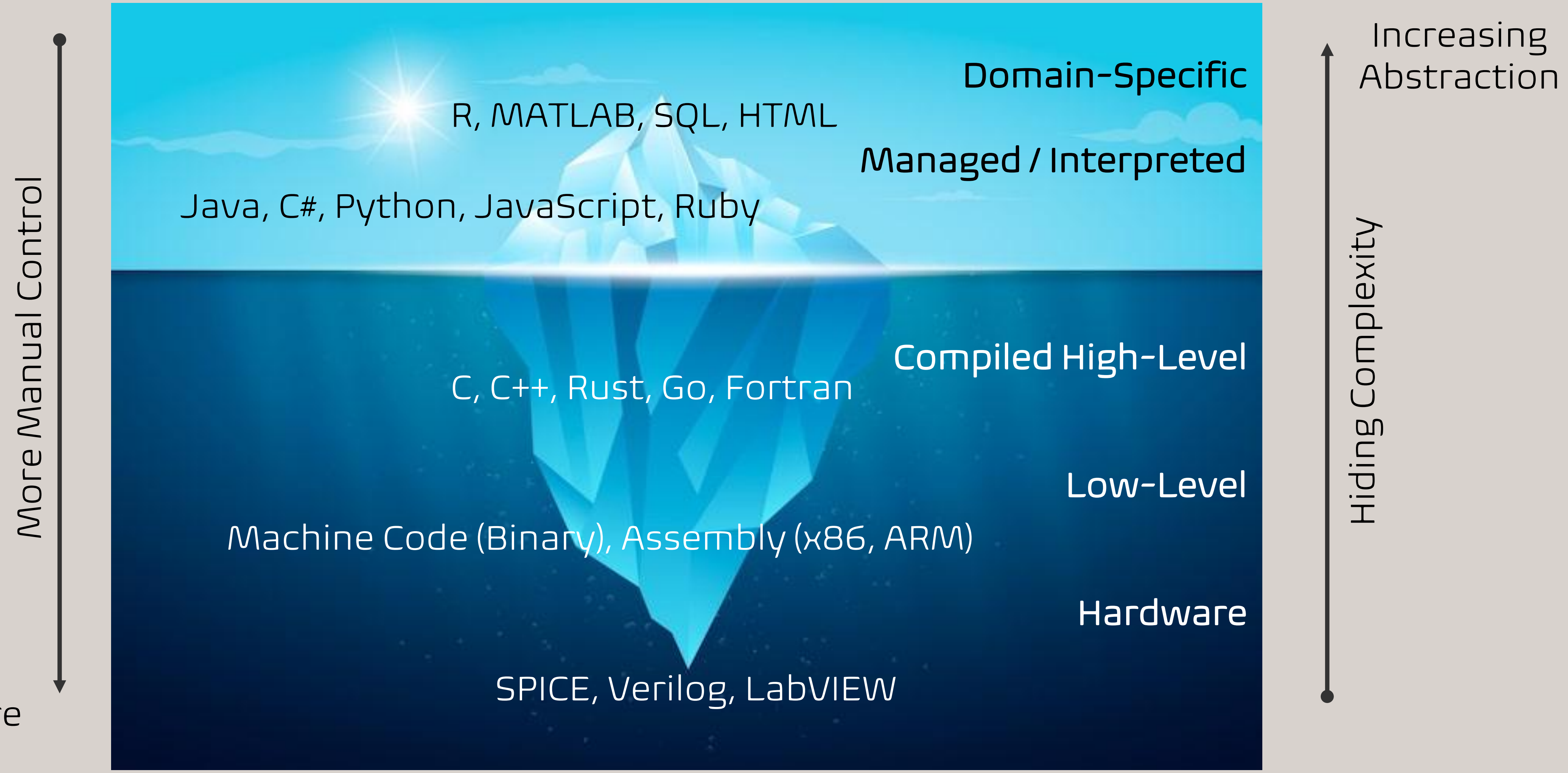
large language models

LLMs

- Custom GPTs
- Claude
- Grok 4
- Gemini
- Llama

English

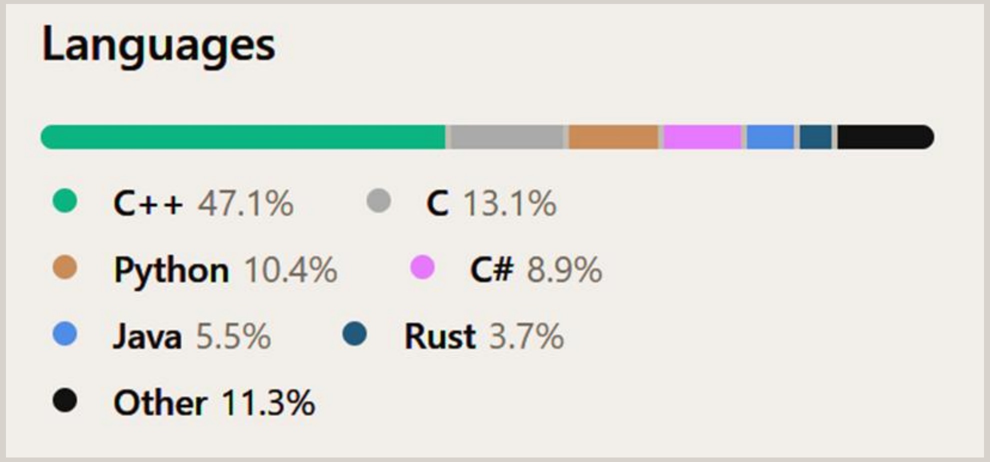
Types of Programming Languages



Complex projects use multiple languages

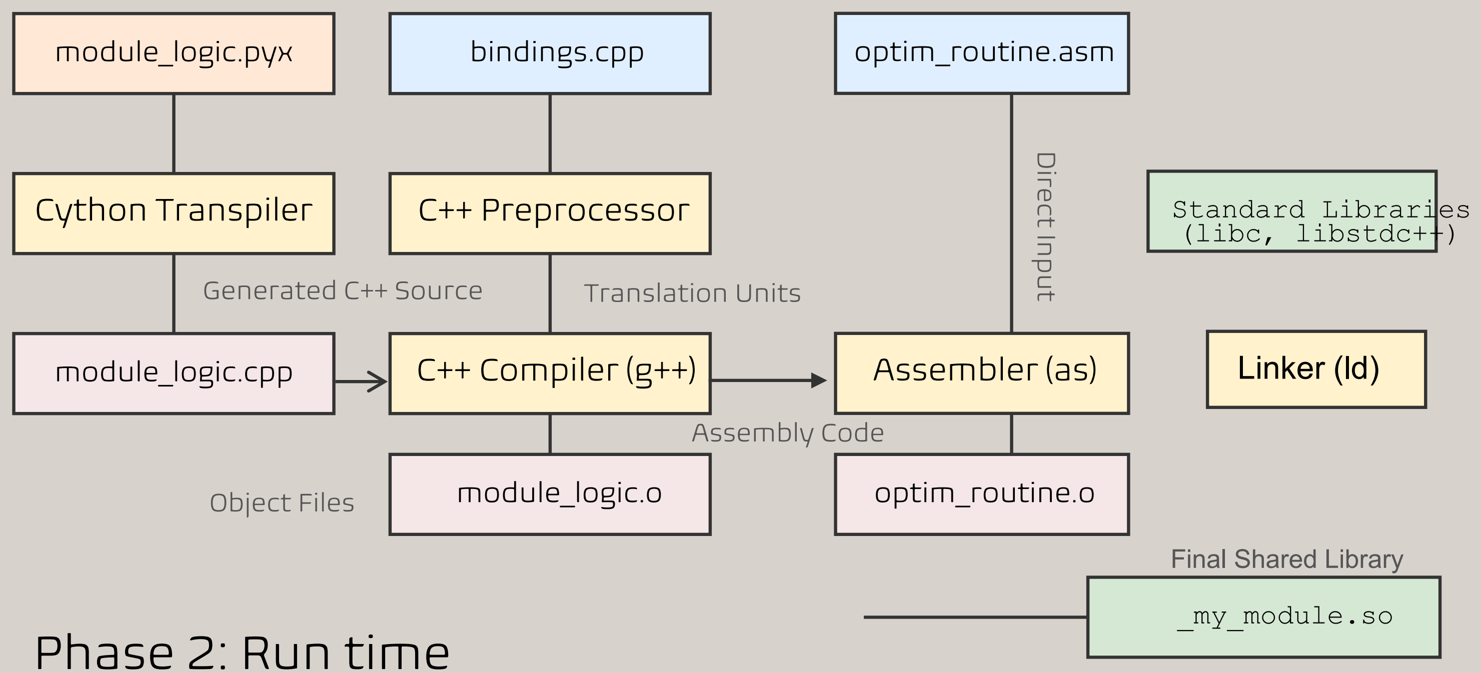
BrainFlow
How Biosensors Work

library intended to obtain, parse and analyze EEG, EMG, ECG, and etc.

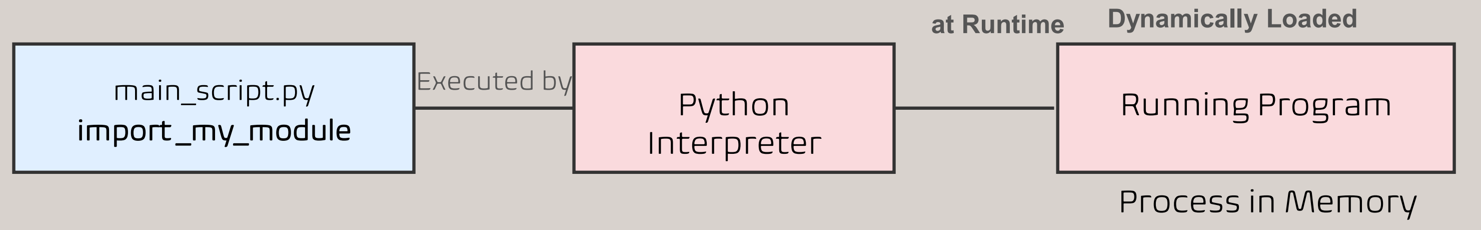

Languages

- ● C++ 47.1%
- ● C 13.1%
- ● Python 10.4%
- ● C# 8.9%
- ● Java 5.5%
- ● Rust 3.7%
- ● Other 11.3%

Phase 1: Build time



Phase 2: Run time



Key Programming Concepts

Syntax
Specific rules (grammar) of a programming language that must be follow to run code

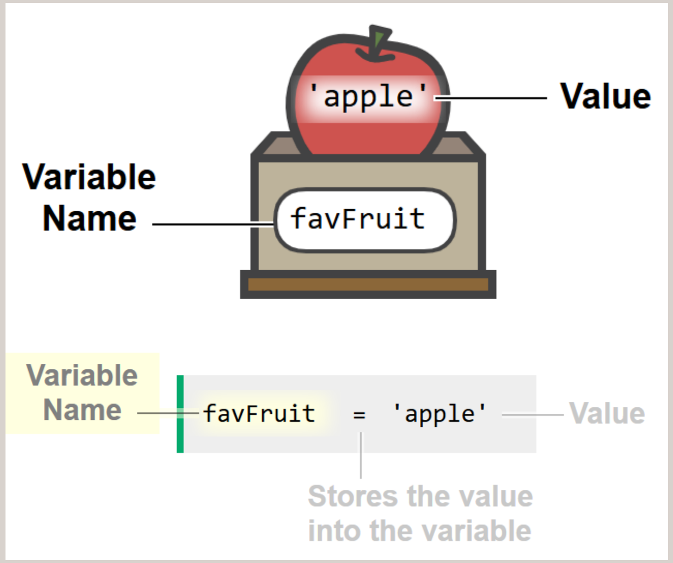
Algorithms
Step-by-step procedures to solve a problem or complete a task

Control Flow
Mechanisms, like loops that determine the order in which instructions are executed

Sequencing
The process of arranging instructions in a specific order

Variables in Programming

variable serves as a named storage location in a computer's memory that holds a value



Memory Abstraction

The operating system and programming language handle the underlying physical memory management

Mutability

values can be changed or updated during execution

Data Types

specifies the kind of data

allocate appropriate memory and perform valid operations

reserved words, or keywords of the language, and cannot be used as ordinary variable.

Data Types

Data types are the kind of values that can be stored in a variable.

Primitive Types:

- Integer (e.g., 5)
- Float (e.g., 3.14)
- Boolean (True/False)
- Char (e.g., 'a')
- Data/Time/Datetime (e.g., '9/29/2025')

Composite Types:

- Array/List (e.g., int[] nums = {1,2,3})
- Tuple (e.g., (int, string) pair = (1, "one"))
- String (e.g., string name = "Alice")

Abstract Types:

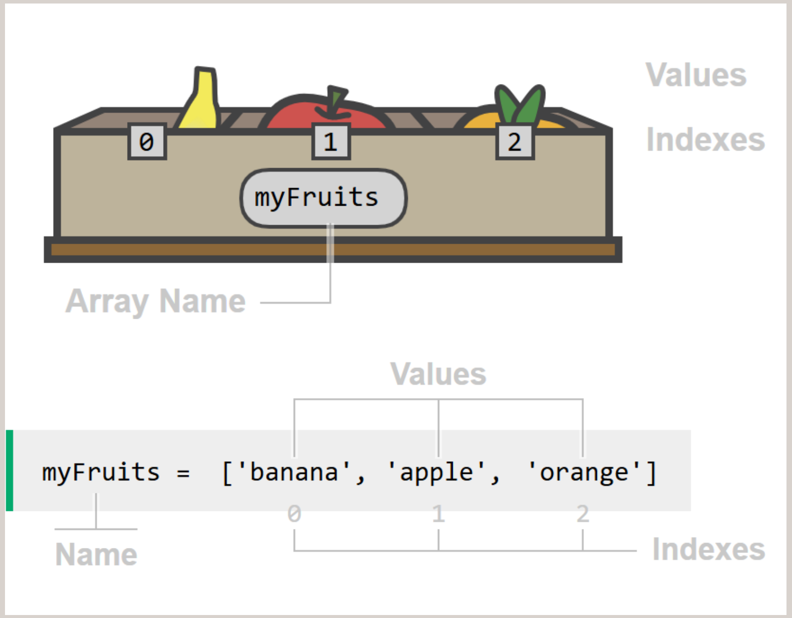
- Stack, Queue, Linked List

Special Types:

- Null or None (absence of a value)
- Enum
- Void (represents the lack of a data type)

Arrays in Programming

Arrays are made for storing many values together.



```
myFruits =  ['banana', 'apple', 'orange']
```

- Contiguous Memory Allocation
- Index-Based Access
- Fixed or Dynamic Size
- Declaration and Initialization

Homogeneous Data Type

Operators in Programming

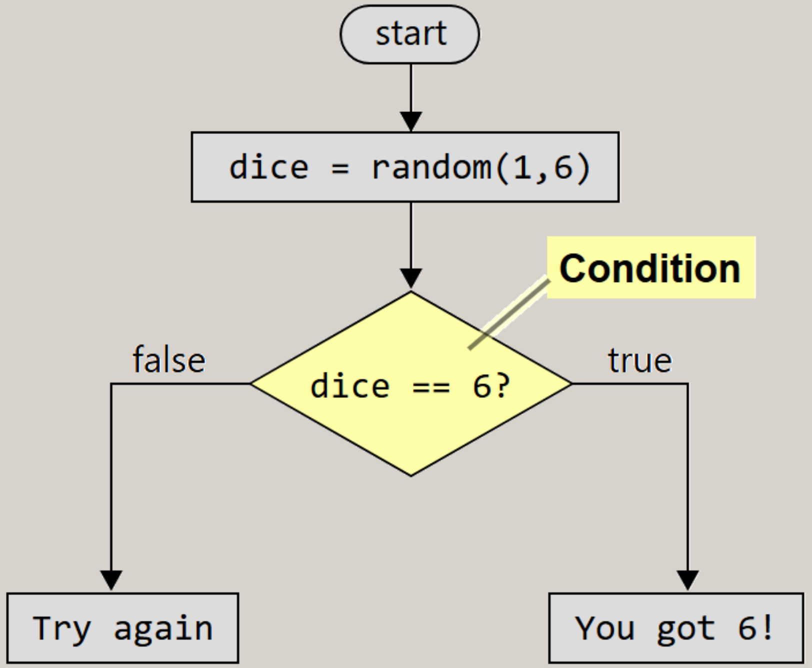
Operators are symbols or keywords that tell the computer what operations to do on values or variables

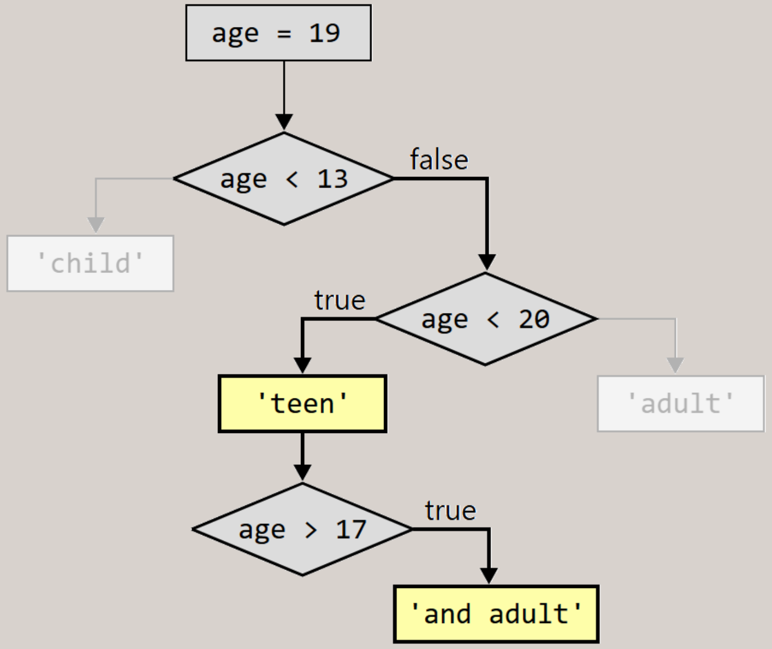
There are many types of operators, but the most common ones are:

- Arithmetic operators (+, -, *, /, %, ** etc.)
- Assignment operators (=, +=, -=, etc.)
- Comparison operators (==, >, <, etc.)
- Logical operators: && (AND); || (OR); ! (NOT)
- Bitwise operators (&, |, etc.)

If Statements in Programming

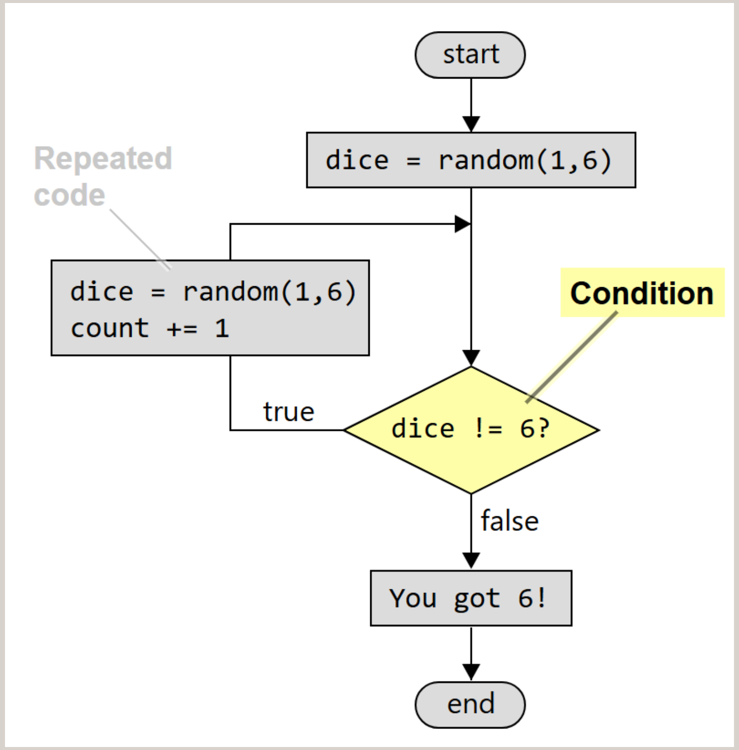
If statements allow your program to make decisions, so it can do different things depending on the situation.





Loops in Programming

Loops are used when we need to run the same code lines many times.
Fundamental for tasks requiring iteration and repetition, offering efficiency and code reusability.



Entry Controlled

for loop
initialize var -> check condition -> execute -> repeat or exit

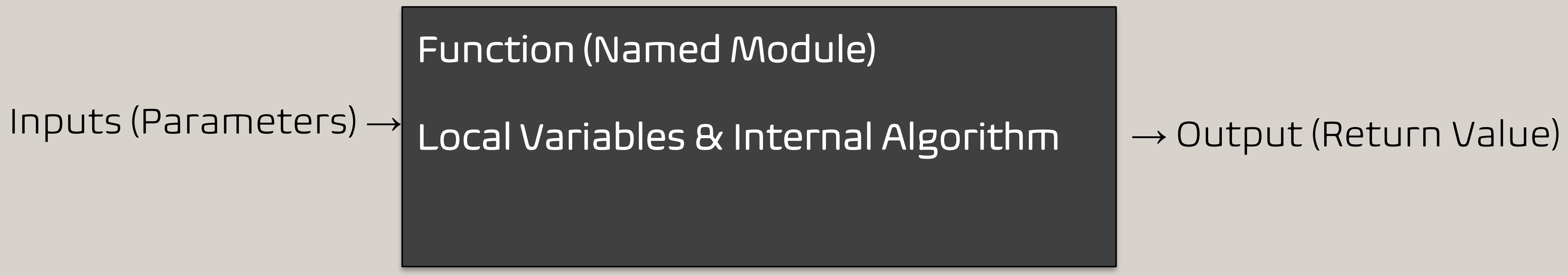
while
check condition - > if true execute -> repeat or exit

Exit Controlled

Do-while
execute a code -> check condition -> if true repeat or exit

Functions

Think of a function as a black box — only inputs and outputs matter.



- Inputs = Formal Parameters (placeholders in definition)
- Outputs = explicit Return Value(s)
- Local scope: workspace isolated
- Promotes modularity, reuse, and clean design

Key Programming Paradigms

Imperative (essentially telling the computer how to perform a task)

Declarative (focusing on what you want the program to achieve)

Procedural Programming

```python
# Calculate the sum of numbers in a list
numbers = [1, 2, 3, 4, 5]
total = 0
for num in numbers:
    total += num
print(total)
```

focuses on how to achieve a result, with explicit instructions on modifying program state

Functional Programming

```python
# Calculate the sum of numbers in a list
numbers = [1, 2, 3, 4, 5]
total = sum(numbers)
print(total)
```

focuses on what to achieve, describing the desired outcome without explicitly detailing the steps

Object-Oriented Programming

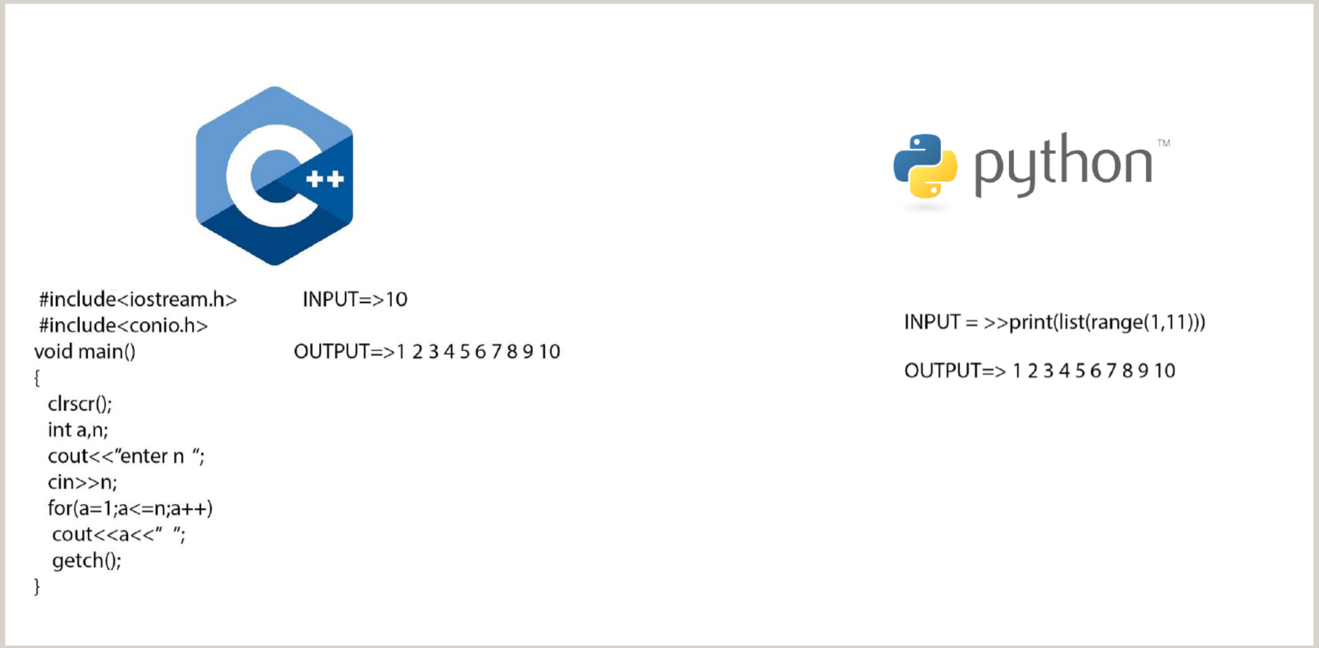
```python
class Calculator:
    def __init__(self, numbers):
        self.numbers = numbers

    def calculate_sum(self):
        return sum(self.numbers)

# Calculate the sum of numbers in a list
numbers = [1, 2, 3, 4, 5]
calc = Calculator(numbers)
print(calc.calculate_sum())
```

organizes code around "objects" that combine data and behavior

03 Why Python?


```
#include<iostream.h>
#include<conio.h>
void main()
{
  clrscr();
  int a,n;
  cout<<"enter n ";
  cin>>n;
  for(a=1;a<=n;a++)
   cout<<a<<" ";
   getch();
}
```

```
INPUT=>10

OUTPUT=>1 2 3 4 5 6 7 8 9 10
```

```
INPUT = >>print(list(range(1,11)))

OUTPUT=> 1 2 3 4 5 6 7 8 9 10
```

functional, compiled, interpreted, object-oriented (class-based), imperative, metaprogramming, extension, impure, interactive mode, iterative, reflective, scripting

How much to focus on Python syntax?

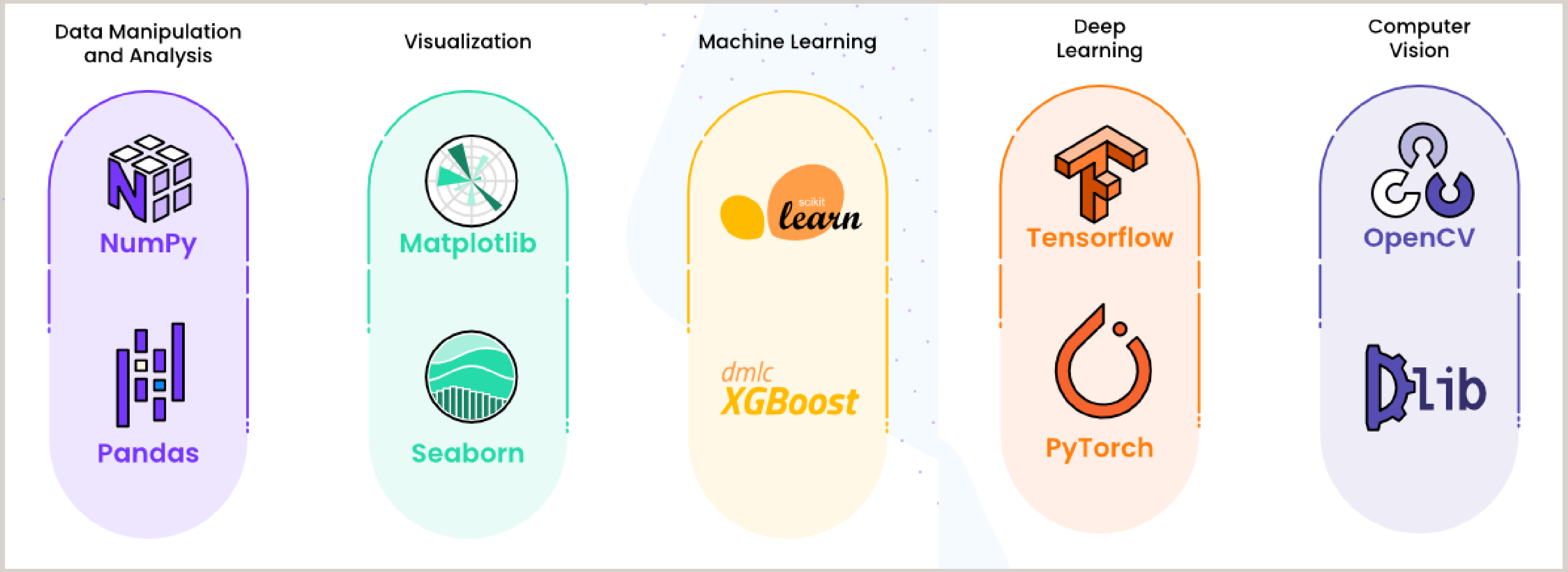
A Python program is divided into number of logical lines, and every logical line is terminated by the token NEWLINE.
A comment begins with a hash character (#) which is not a part of the string literal and ends at the end of the physical line. All characters after the character up to the end of the line are part of the comment and the Python interpreter ignores them.

Python Coding Style:

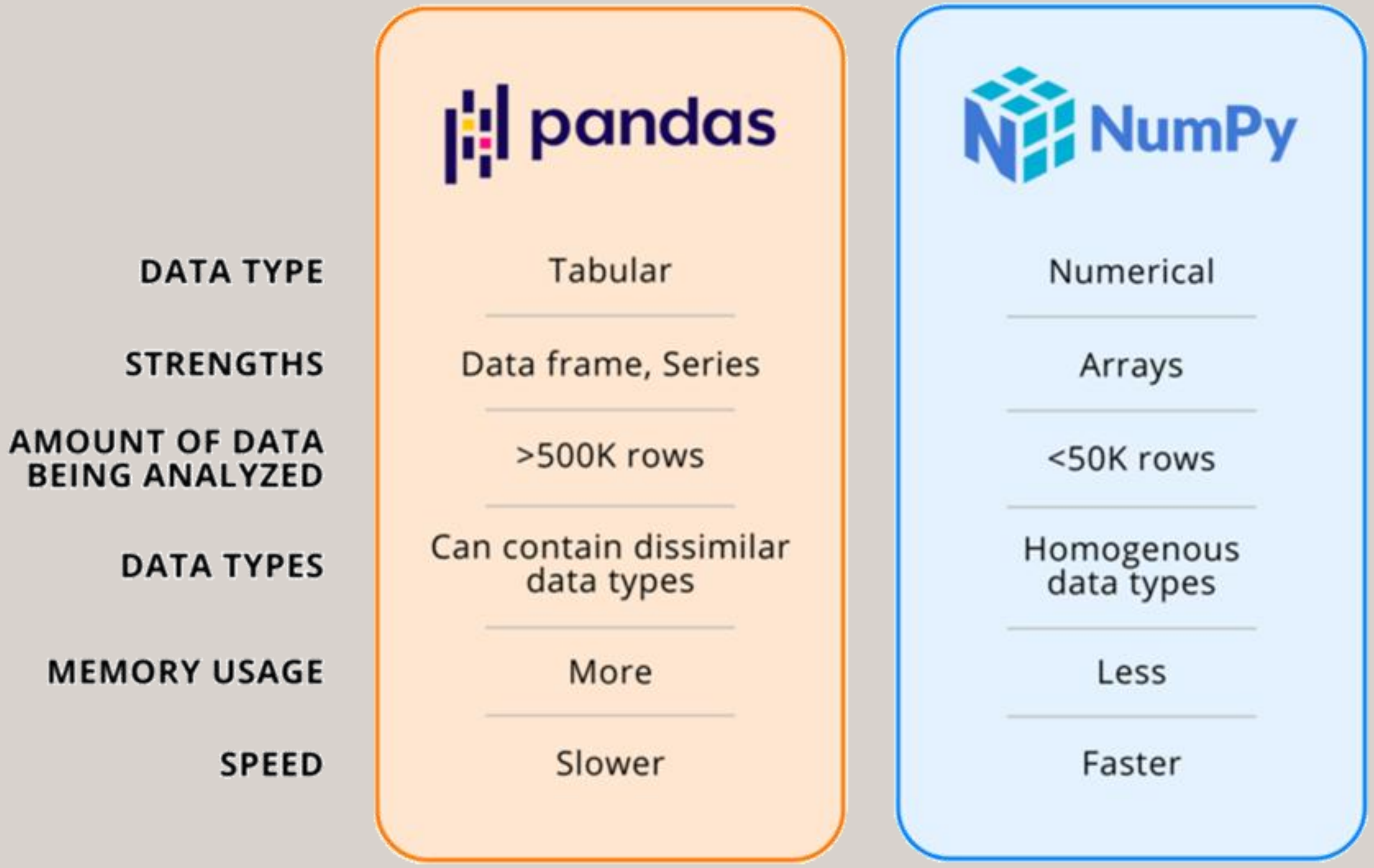
• Use 4 spaces per indentation and no tabs.
• Do not mix tabs and spaces. Tabs create confusion and it is recommended to use only spaces.
• Maximum line length : 79 characters which help users with a small display.
• Use blank lines to separate top level function and class definitions and single blank line to separate methods definitions inside a class and larger blocks of code inside functions.
• When possible, put inline comments.
• Use spaces around expressions and statements.

Key Python libraries

Python is a community of developers

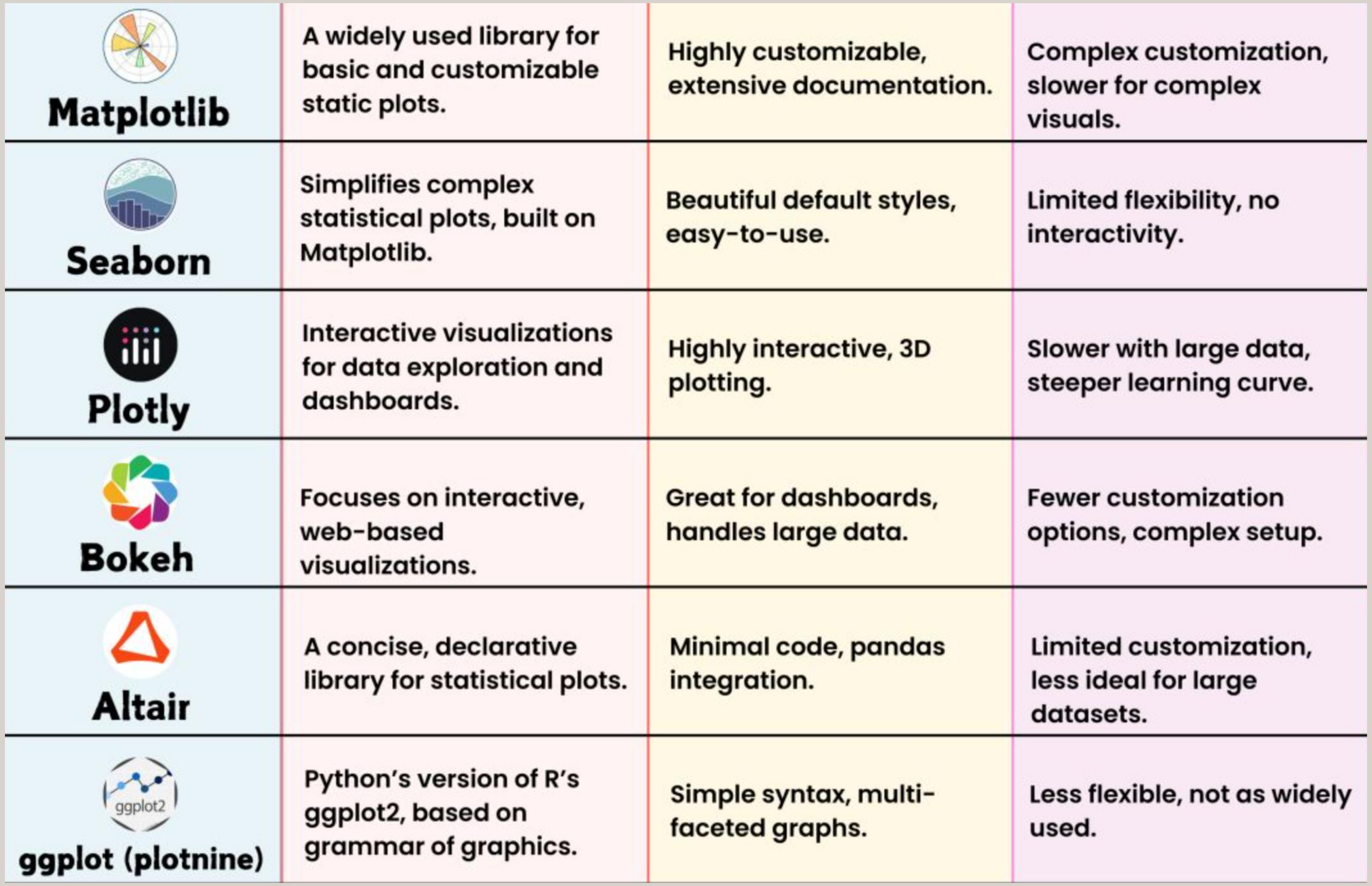


Libraries for scientific computing



	pandas	NumPy
DATA TYPE	Tabular	Numerical
STRENGTHS	Data frame, Series	Arrays
AMOUNT OF DATA BEING ANALYZED	>500K rows	<50K rows
DATA TYPES	Can contain dissimilar data types	Homogenous data types
MEMORY USAGE	More	Less
SPEED	Slower	Faster

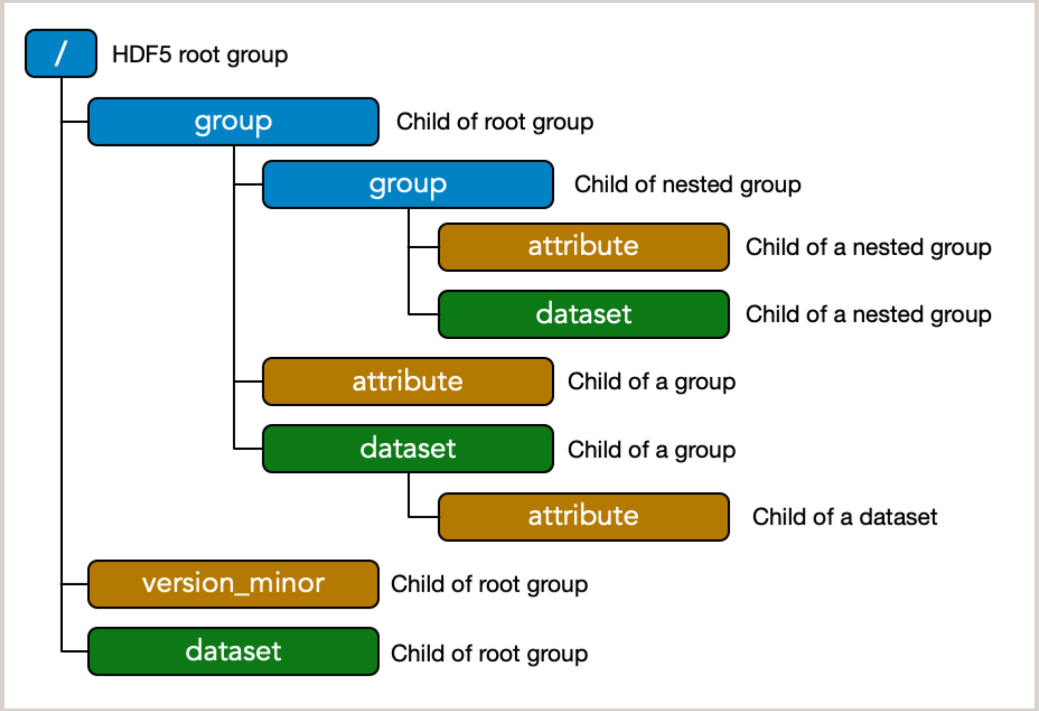
Python library for data visualization

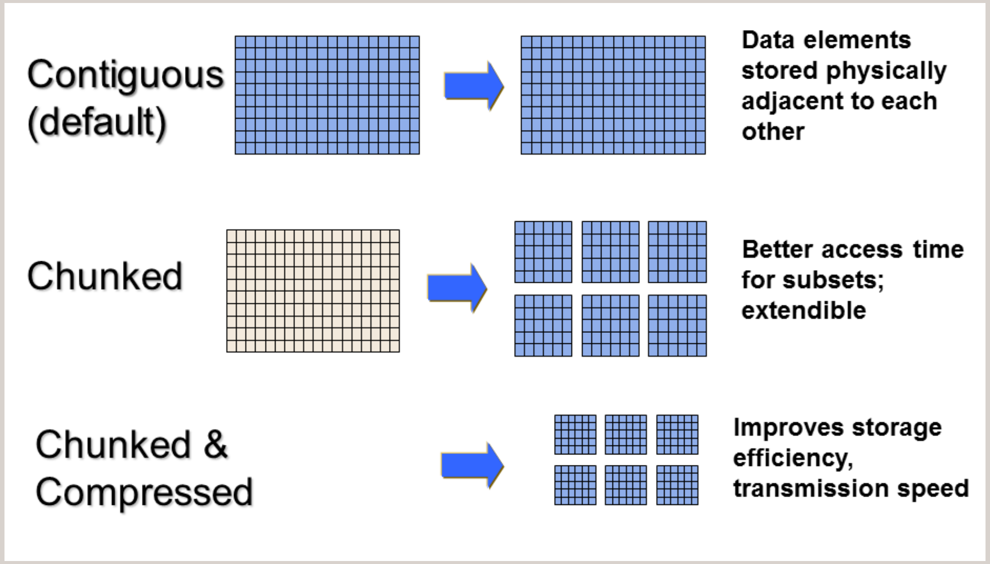


Matplotlib	A widely used library for basic and customizable static plots.	Highly customizable, extensive documentation.	Complex customization, slower for complex visuals.
Seaborn	Simplifies complex statistical plots, built on Matplotlib.	Beautiful default styles, easy-to-use.	Limited flexibility, no interactivity.
Plotly	Interactive visualizations for data exploration and dashboards.	Highly interactive, 3D plotting.	Slower with large data, steeper learning curve.
Bokeh	Focuses on interactive, web-based visualizations.	Great for dashboards, handles large data.	Fewer customization options, complex setup.
Altair	A concise, declarative library for statistical plots.	Minimal code, pandas integration.	Limited customization, less ideal for large datasets.
ggplot (plotnine)	Python's version of R's ggplot2, based on grammar of graphics.	Simple syntax, multi-faceted graphs.	Less flexible, not as widely used.

Hierarchical Data Format v5 (HDF5) for Python

HDF5 is the primary backend for NWB (Neurodata Without Borders)

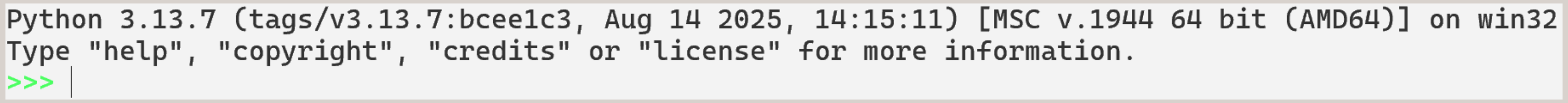



Contiguous (default)
Data elements stored physically adjacent to each other
Chunked
Better access time for subsets; extendible
Chunked & Compressed
Improves storage efficiency, transmission speed

The **h5py package** provides both a high- and low-level interface to the HDF5 library from Python. The low-level interface is intended to be a complete wrapping of the HDF5 API, while the high-level component supports access to HDF5 files, datasets and groups using established Python and NumPy concepts.

04 How to get started with Python?

On many systems Python comes pre-installed, you can try running the python command to start the Python interpreter to check and see if it is already installed.



```
Python 3.13.7 (tags/v3.13.7:bcee1c3, Aug 14 2025, 14:15:11) [MSC v.1944 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

If not go to https://www.python.org/downloads/ download and install

free online tools or user-friendly integrated development environments (IDEs),
https://mybinder.org/ , https://colab.google/, https://py3.codeskulptor.org/,
https://www.browxy.com/

Desktop-Based IDEs
Visual Studio Code (VS Code), Spyder, Sublime Text

Resources Local vs Cloud vs HPC

Local on your personal computer

Provide direct control of environment and version
Good for running scripts, developing and testing applications

Cloud computing

Perfect for prototyping, no need to manage and maintain the infrastructure.
Easily scale computing power, storage, and other resources (Pay-as-you-go model)

HPC - cluster.einsteinmed.edu

Large-scale simulations, complex data analysis, AI/ML model training, and any task that can be massively parallelized across many nodes.

05 How LLMs assist in Python programming

Code Explanation & Documentation
Explain code, add docstrings, and improve readability.

Code Completion & Suggestions
Predict next lines, functions, or variables in IDEs.

Debugging & Error Correction
Identify errors, explain messages, and suggest fixes.

Test Case Generation
Assist in creating unit and integration tests.

Refactoring & Optimization
Suggest better structure, readability, and performance.

Code Generation
Generate snippets, functions, or scripts from natural language.

Time to practice
Open the Google Colaboratory (Colab)

Final projects Datasets

1. Kondo, M., Sehara, K., Harukuni, R. et al. Multimodal dataset linking wide-field calcium imaging to behavior changes in operant lever-pull task in mice. Sci Data 12, 1264 (2025). https://doi.org/10.1038/s41597-025-05482-y ID: 001425

2. International Brain Laboratory., Angelaki, D., Benson, B. et al. A brain-wide map of neural activity during complex behaviour. Nature 645, 177–191 (2025). https://doi.org/10.1038/s41586-025-09235-0 ID: 001533

3. Jennifer J. Sun, Tomomi Karigo, David J. Anderson, Pietro Perona, Yisong Yue, & Ann Kennedy. (2021). Caltech Mouse Social Interactions (CalMS21) Dataset (1.0) [Data set]. CaltechDATA. https://doi.org/10.22002/D1.1991

4. Rodgers, C.C. A detailed behavioral, videographic, and neural dataset on object recognition in mice. Sci Data 9, 620 (2022). https://doi.org/10.1038/s41597-022-01728-1 ID: 000231

5. Tingley D, Buzsáki G. Transformation of a Spatial Map across the Hippocampal-Lateral Septal Circuit. Neuron. 2018 Jun 27;98(6):1229-1242.e5. doi: 10.1016/j.neuron.2018.04.028. Epub 2018 May 17. PMID: 29779942; PMCID: PMC6605060. ID: 000213