# Lecture 1

2025-09-03

Course introduction; Overview of computational neuroscience; Basics of neurobiology; Python and Jupyter Notebook setup; Quickstart on Python programming

- Online slides: lec01-intro.html
- Code: code01.ipynb
- Homework: hw01.html, hw01-dat.txt

Username: cns, Password: nycu2025

Laboratory of Computing Biological Networks

# Course information

## Computational Neuroscience, NYCU

Autumn 2025

**Time:** 09–12 on Wednesdays

**Room** 839, Library, Information and Research Building

**Instructor:** Chun-Chung Chen

### Textbook (optional)

- "*Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*" by Dayan & Abbott (2005, ebook)

### Additional references

- "*Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition*" by Gerstner et al. (2014, online)
- "*Computational Neuroscience*" by Students of NS/PY 357 Bates College (2022, online)

# Background of instructor

**Statistical and Computational Physics**

Directed Percolation, Interface growth model, Sandpile avalanche, Non-equilibrium critical phenomena

**Theoretical Modeling of Reversibly-Associated Polymers**

Ring-chain equilibrium, Metallosupramolecular Polymers, Diblock copolymer micelle, End-functionalized polymer brush

**Physics of Biological Networks**

Complex systems and complex networks, Integrate-and-fire neurons, Spike-timing-dependent plasticity

# Personal hobbies and interests

- Programming
- Network applications
- Electronic circuits
- Game of Go
- Photography

# Course coverage

- Computational and mathematical tools
- Neural representations
- Modeling neural circuits
- Functions of neural systems

Another course: Data Analysis in Neuroscience in Springs

... Information extraction from neural data

# Lecture and tutorial

- Each class will consist of two hours of lecture and one hour of tutorial.
- The lecture will start with a review of the last lecture and any questions can be raised at anytime during the lecture.
- The tutorial hour (11am to noon) is optional. The instructor will remain to answer questions or help with your class work.
- It is recommended to go through the homework or at least make sure you know how to complete everything before leaving the class.
- You are also welcome to bring your own, preferably related, research questions or problems to the tutorial session for free discussion.

# Homework and grades

- Homeworks will be graded on efforts and correctness. You will receive credits just by turning them in.
- Homeworks that are late within a week will be accepted but will receive only 90% of original credits.
- Homeworks that are late for more than a week will still be accepted before the final week but will receive only 80% of original credits.
- The final exam will be either in class or (most likely) take home, to be determined before the final week.
- The final grades will base on the homework credits and the final exam score.

# How to turn in homeworks

- Homeworks should be turned in through the E3 Digital Teaching Platform @ https://e3p.nycu.edu.tw/.
- Put all text writing and code of your answers in a single Jupyter Notebook (`.ipynb`) file unless otherwise specified.
- Restart the kernel and run all cells in one shot to make sure all codes are correct. Make sure your code cells can be run in sequence without error.
- Save the file and upload it to the E3 server when everything looks satisfactory.
- Graded homeworks will be returned as annotated PDF files.
- Solutions to homeworks will be posted to the e3 server when they become available.

# What is computational neuroscience

*Computational neuroscience is the field of study in which mathematical tools and theories are used to investigate brain function. It can also incorporate diverse approaches from electrical engineering, computer science and physics in order to understand how the nervous system processes information.*

nature.com

I. Computational analysis of neural data

II. Study of computation in neural systems

III. Application of neural computational principles

# Three aspects of computational neuroscience

## Computational analysis of neural data

Studies beyond experimenting with and measuring of the neural systems. (Using models and computers) To be further considered in "*Data Analysis in Neuroscience*"

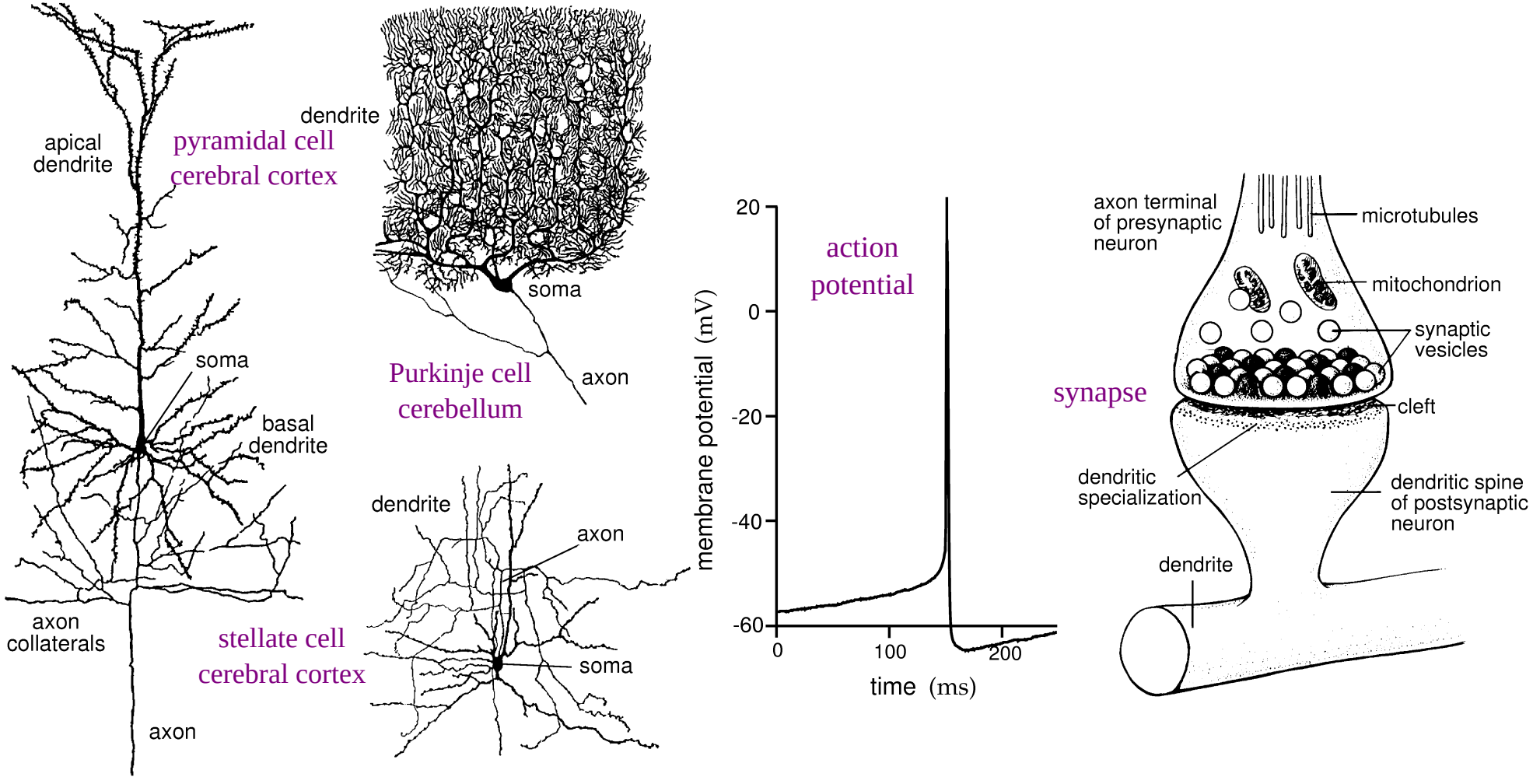
## Study of computation in neural systems

Information processing in biological neural networks: perception, memory, coding, organization, decision making, behavior generation

## Application of neural computational principles

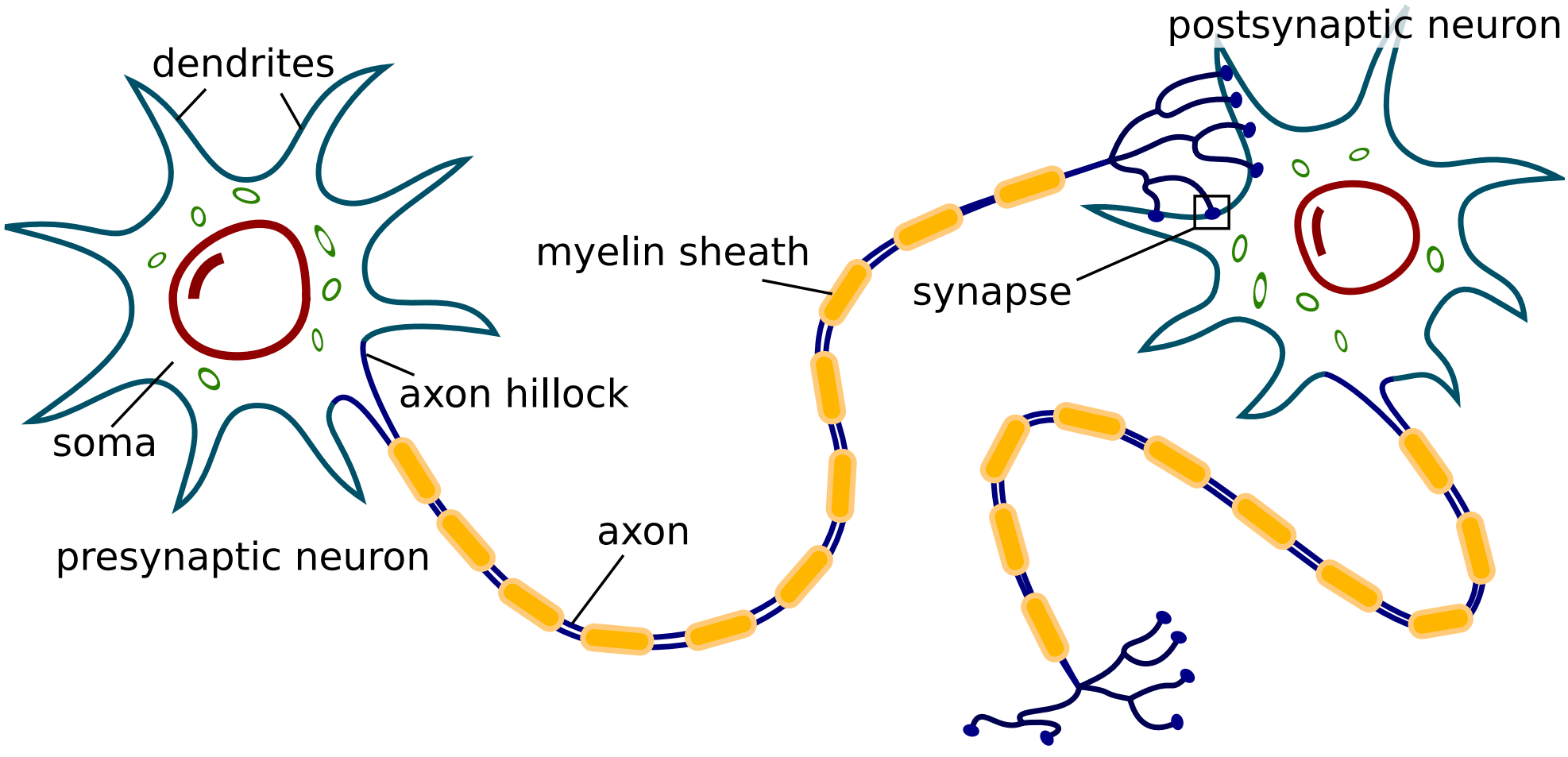
Architecture of computational networks, principles of learning, artificial intelligence, nature of cognition and consciousness

# Some basics of neurobiology

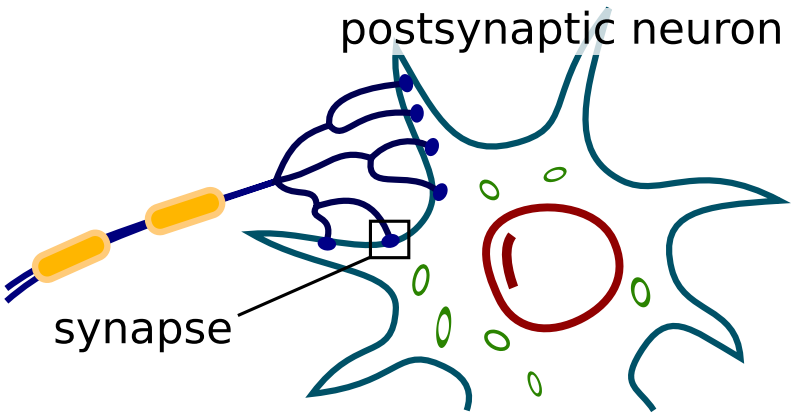


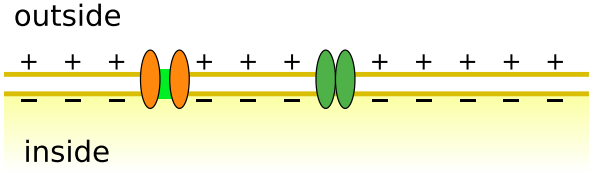

Dayan & Abbott (2001)

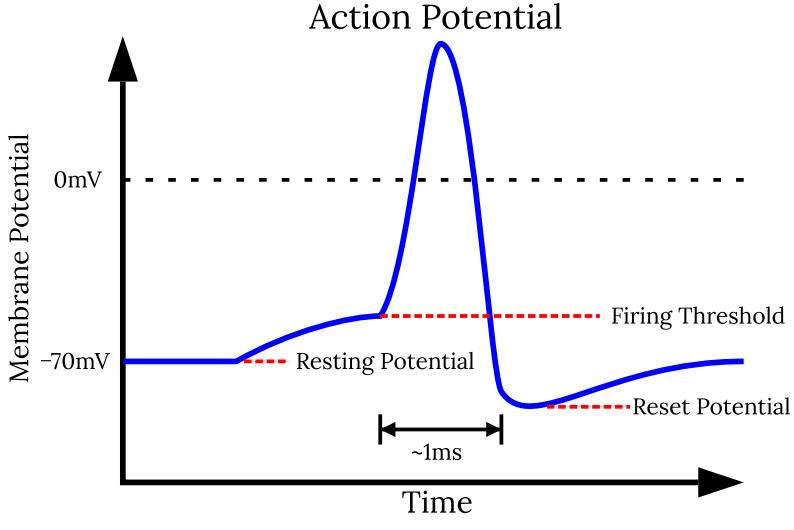
# Structure of neurons



# Neural transmission



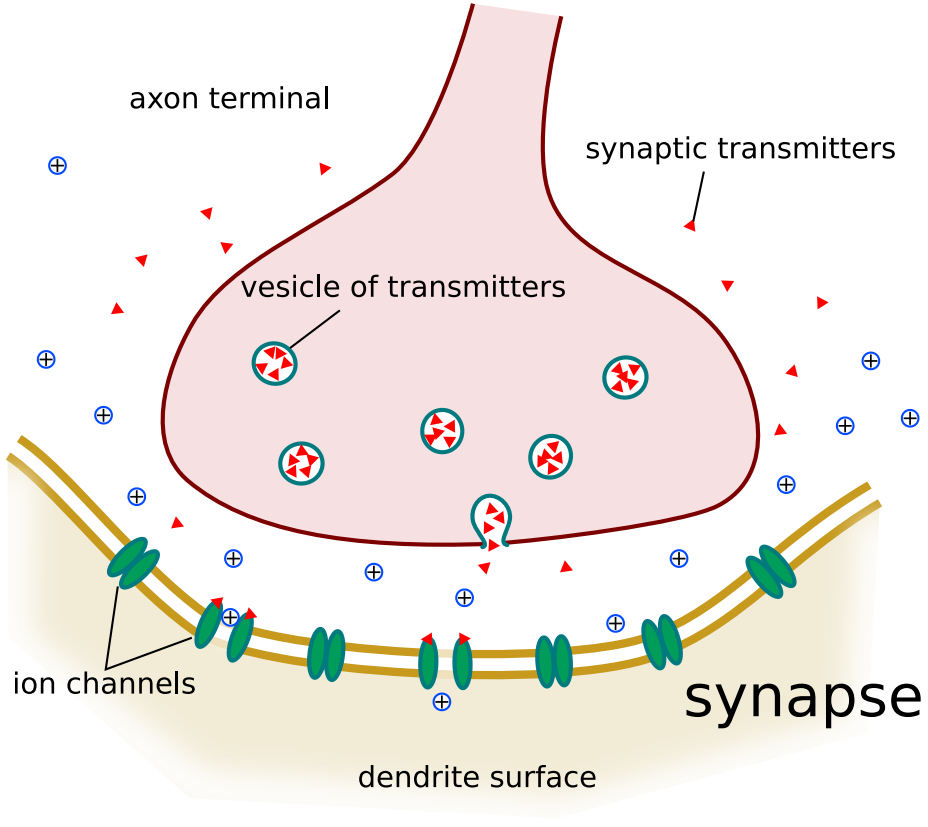




Inside Cell: Membrane Potential (Elect.)

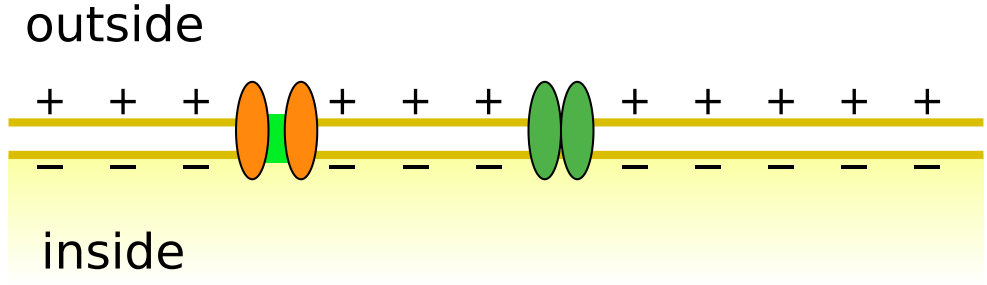
Between Cells: Synaptic Transmitter (Chem.)

Message: Firing (Emit Action Potential)



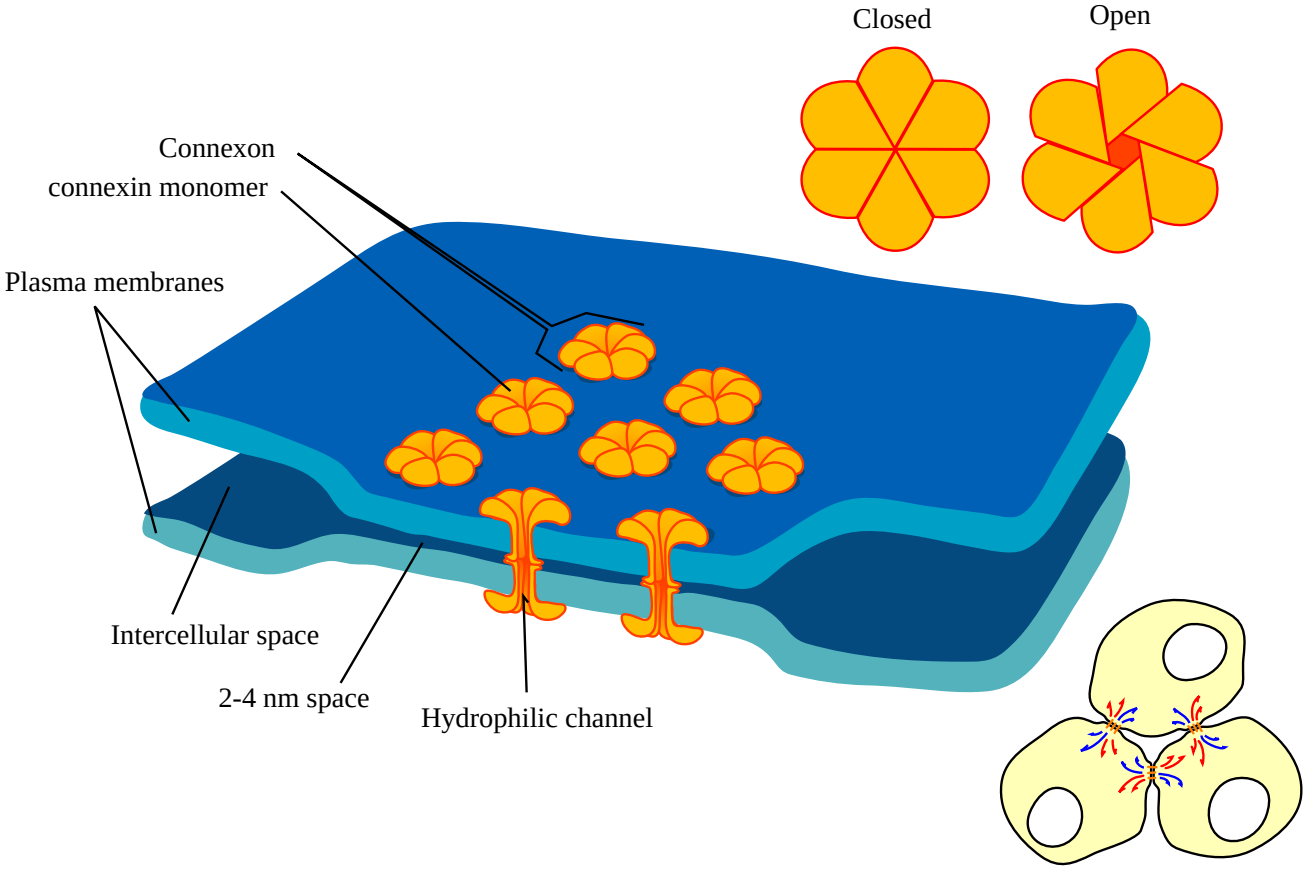
# At the membrane of a neuron

- Major ions: sodium ($Na^+$), potassium ($K^+$), calcium ($Ca^{2+}$), and chloride ($Cl^-$). With more $Na^+$ outside, more $K^+$ inside
- Resting potential $\approx -70$ mV
- Action potential $\Delta V \approx 100$ mV, $\Delta t \approx 1$ ms, followed by absolute and relative refractory periods



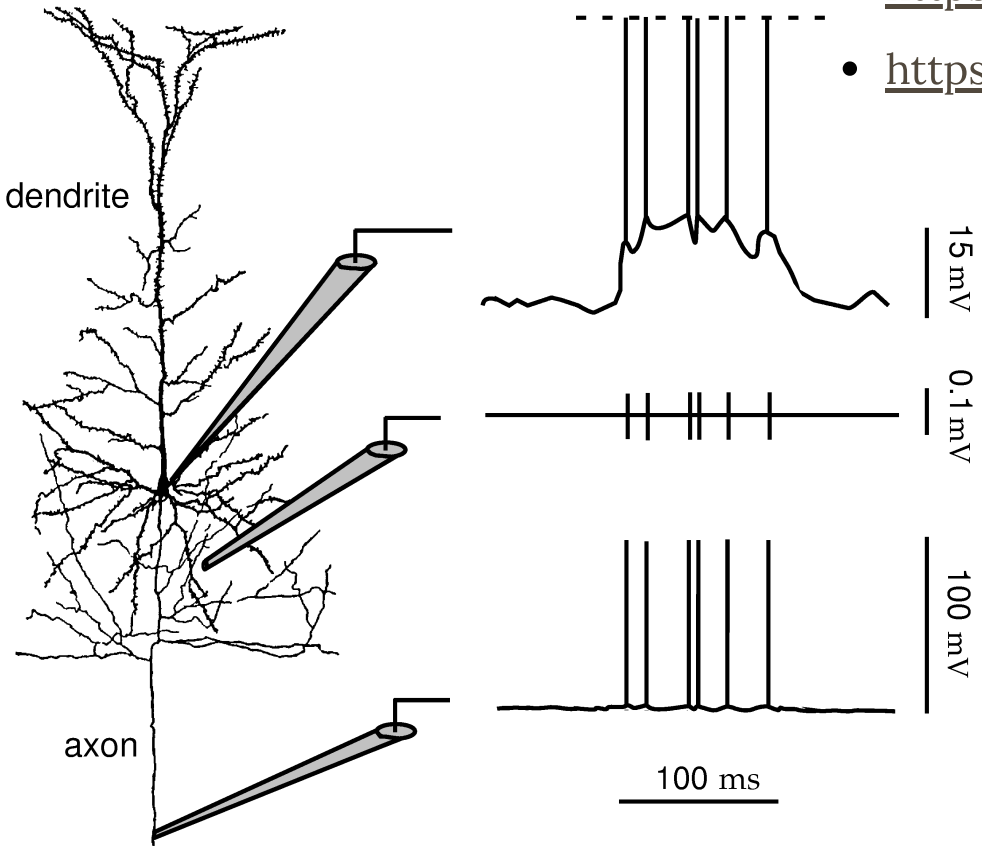
# Gap junctions

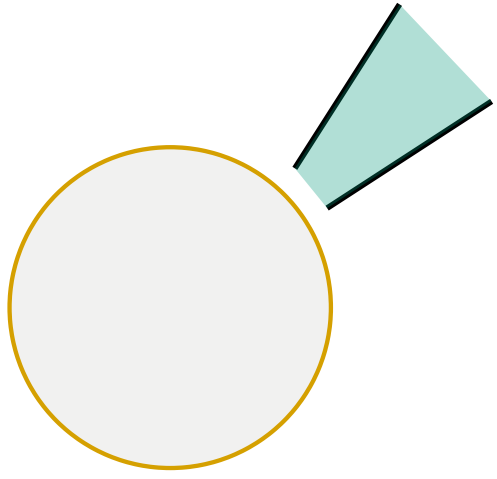
https://en.wikipedia.org/wiki/Gap_junction



# Measuring neural activities

- https://en.wikipedia.org/wiki/Electrophysiology
- https://en.wikipedia.org/wiki/Patch_clamp



Dayan & Abbott (2001)

# Installing Python on your computer

- Required version: ≥ 3.13
- Recommended distribution: Miniconda
- Package management: `conda` (command-line)
- Development environment: JupyterLab

## Other distributions

- Anaconda
- Python.org

# Installing Miniconda with a shell

## Download and install

1. wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
2. bash Miniconda3-latest-Linux-x86_64.sh -b -p $HOME/miniconda
3. source miniconda/bin/activate
4. conda init

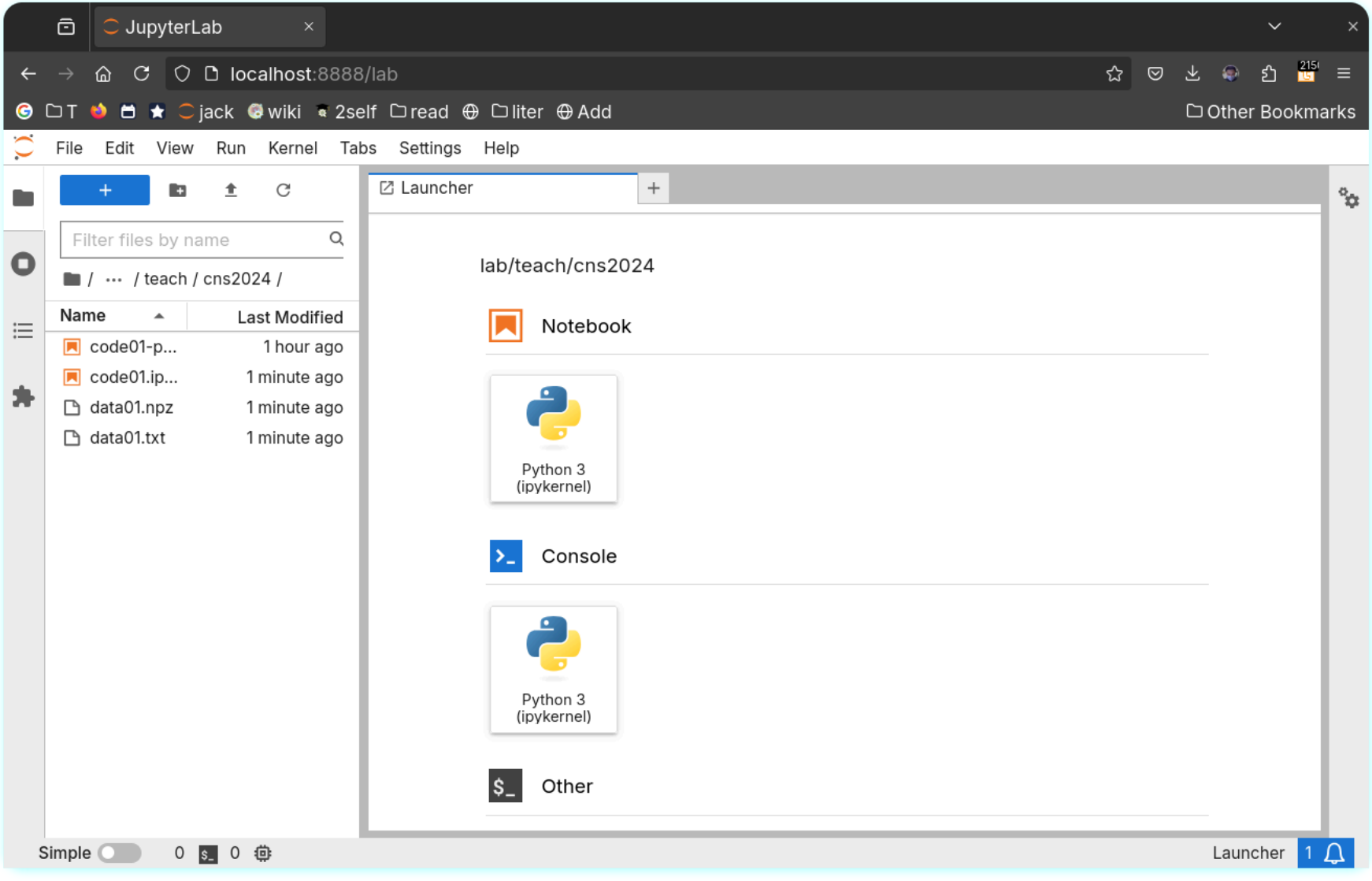
## Bring up-to-date and install essential modules

1. conda update --all
2. conda install jupyterlab numpy matplotlib ipympl scipy

## Start JupyterLab

1. jupyter-lab“ ”

# JupyterLab


JupyterLab
localhost:8888/lab
jack
wiki
2self
read
liter
Add
Other Bookmarks
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Launcher
Filter files by name
/ ··· / teach / cns2024 /
Name
Last Modified
code01-p...
1 hour ago
code01.ip...
1 minute ago
data01.npz
1 minute ago
data01.txt
1 minute ago
lab/teach/cns2024
Notebook
Python 3 (ipykernel)
Console
Python 3 (ipykernel)
Other
Simple
0
0
Launcher
1

# Programming in Python

## Getting Started with Python

- `print("Hello World!")`
- Data Types, Variables, Literals, and Operators
- User Input, Comments, Type Conversion
- Data Structure: List, Tuple, String, Set, Dictionary
- Iterable: range, ...
- Flow Control: `if`, `while`, `for`, `break`, `continue`, and `pass`

See `code01.ipynb`

# NumPy and Matplotlib

NumPy Quickstart

Pyplot Tutorial

- `numpy.array`: uniformly typed multi-dimensional array
- Plotting data and functions
- Splicing and indexing
- Getting and saving data from and to files

See `code01.ipynb`

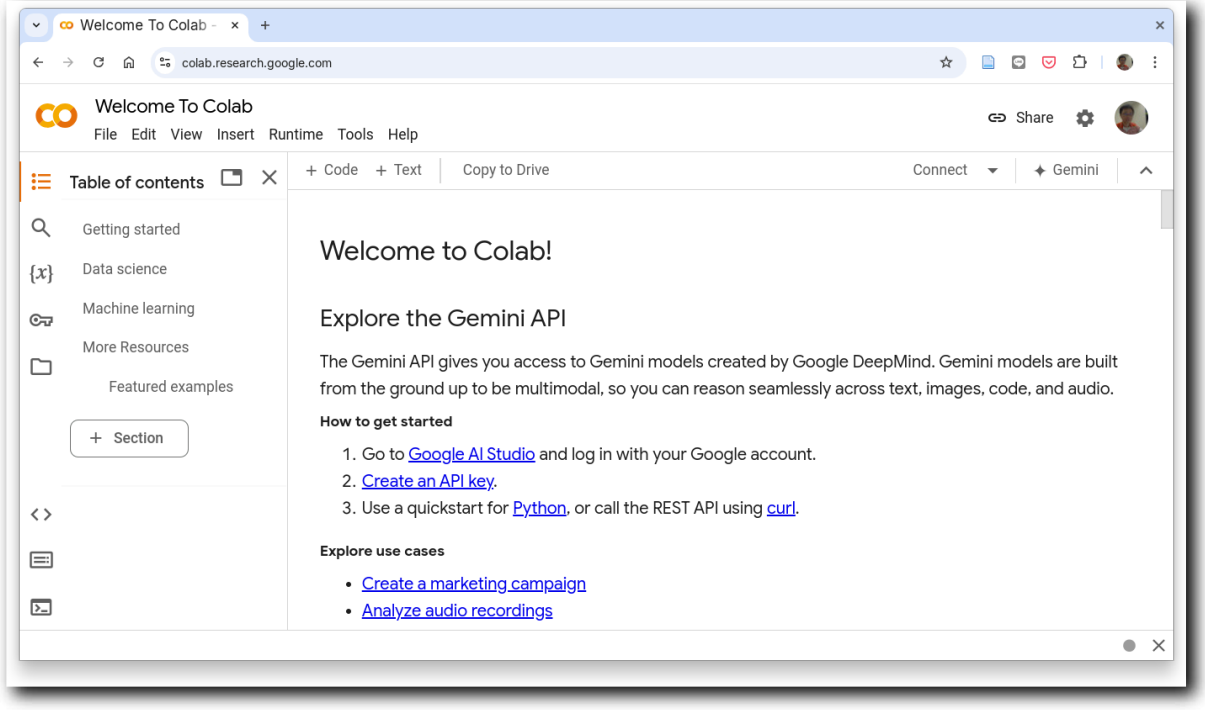
# Alternative IDEs for Jupyter Notebook

# Google Colaboratory

https://colab.research.google.com/

A Colab notebook is essentially a Jupyter notebook. You get to do everything required in the class without needing to install anything. All files are stored in your Google Drive which provides 15G space if you use your NYCU account.

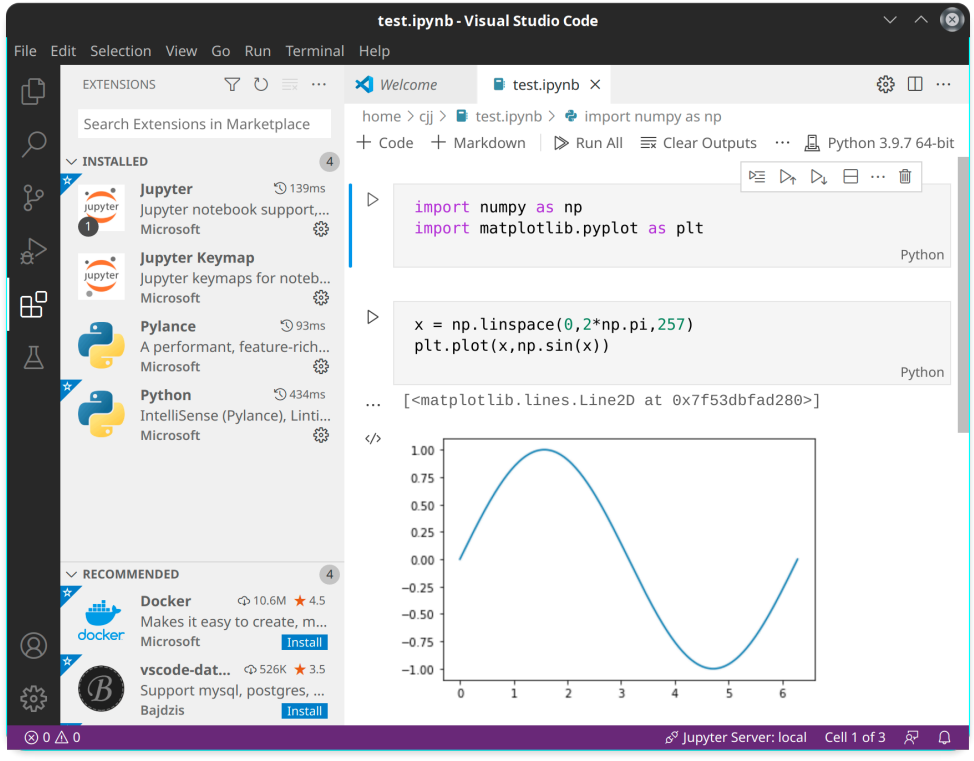
You can see code01.ipynb on Colab.



# Visual Studio Code

https://code.visualstudio.com/

VS Code is actually a code editor with a comprehensive plugin system. After installing Jupyter, support for running and editing `ipynb` files can be added with the plugins: `Python`, `Jupyter`



# PyCharm

https://www.jetbrains.com/pycharm/

PyCharm created by Jet Brains is also a very popular IDE for Python.