

Programming for ML and Data Science

Preliminaries

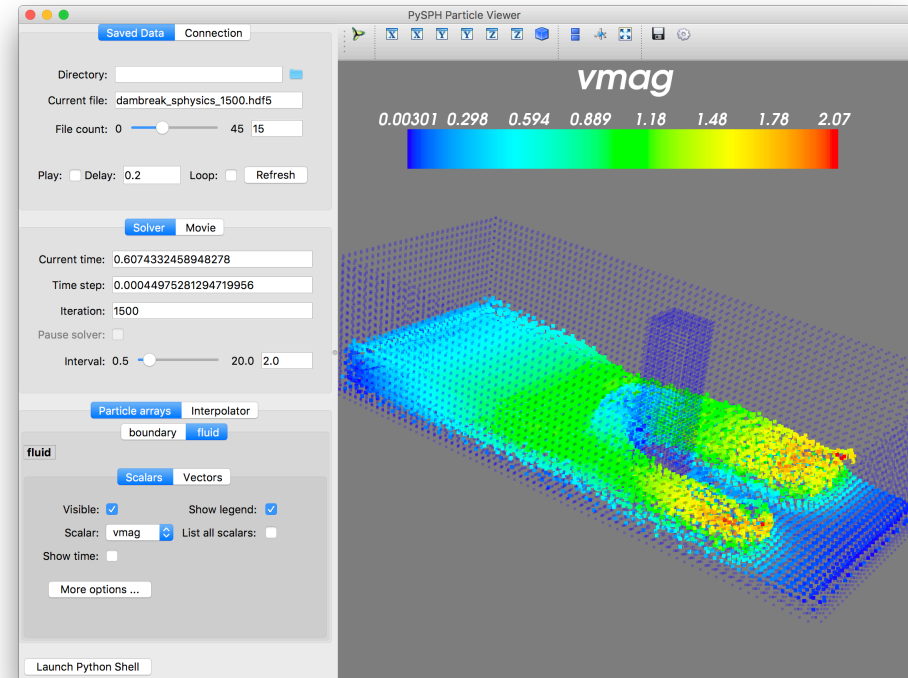
Prabhu Ramachandran

Department of Aerospace Engineering, IIT Bombay

2026-02-01

About Prabhu

- Professor of Aerospace Engineering
- Work broadly in computational science
 - Numerical simulation of fluids
 - High perf. and parallel computing
 - Scientific data visualization
- Open source contributor
 - Mayavi (2001): 3D visualization
 - PySPH (2012) and many others
- Programming Python since 1998
- Nominated [PSF fellow](#) since 2010



General information

- Schedule: Thu. 7:30pm to 9pm and Sat. 9:30am to 11:00am.
- Prof. Vinay Kulkarni and Prof. Prabhu Ramachandran

TAs

- Abhishek Choudhary
- Aabid Hassan
- Brijesh Kumar
- Ashutosh Kumar Singh
- Dibyajyoti Baidya
- Mayank Bajaj

Structure of course

- 2 x 1.5 hour lectures per week
- 1 hour tutorial session
- 24 lectures and 12 tutorials
- Plenty of hands-on material and exercises

Evaluation

- Quiz: 5 x 2 Marks
- Mid term test: 20 Marks
- Project: 20 Marks
- End semester examination: 50 marks

Getting the best out of the course

- Programming requires doing
- Not enough to read
- Type, execute the code, change it, and run it
- Experiment
- Do the homework
- Attend the tutorial classes
- Complete periodic take-home exercises

Expectations

- Expect you to give your best
- Stop us if you have questions
- Keep questions focussed on the content
- Also ask TAs questions

What we do not teach

- Too much theory
 - Focus of other the other courses
- No specific end-user applications
 - Focus on tools of the trade

Learning outcomes

- At the end of the course you will
 - Be able to use Python for ML/DS
 - Get a broad overview of applied ML/DS
 - Create ML models and evaluate them

Course syllabus

- Introduction to Python and Jupyter
- Data visualization and plotting with Python
- Overview of programming concepts
- Statistical plotting and computation
- Probabilistic programming
- Processing images and text
- Exploratory data analysis with pandas

Course syllabus

- High level overview of ML
- Regression, Classification, Clustering
- Data science life cycle
- Feature engineering for text, image, audio, and video
- Data engineering
- Big data and cloud resources

Course outline: part 1











- **Week 1: Introduction to Python Programming**
 - Jupyter notebooks, and plotting with matplotlib, using numpy arrays
 - More Python and plotting: data structures, loops, conditionals, writing functions
 - Local installation, writing and running Python scripts
- **Week 2: Python for statistical analysis**
 - Review of Python core concepts
 - Python for statistics using numpy and matplotlib
 - Loading and visualizing data
 - Introduction to probabilistic programming with numpy and scipy
- **Week 3: Python for statistics data processing**
 - Probabilistic programming with numpy and scipy
 - Elementary image processing with numpy
 - Interactive widgets

Course outline

- **Week 4: Advanced numpy and text processing**
 - Advanced numpy: special indexing, broadcasting, masked indexing
 - Text processing with Python, input and output
 - Applications with other Python data structures
- **Week 5: Exploratory data analysis**
 - Introduction to pandas for data analysis
 - Reading, processing, and plotting data using pandas
 - Basic regression using numpy, statsmodels, and scikit-learn
- **Week 6: Regression and classification**
 - Linear and polynomial regression
 - Logistic regression, decision trees and random forests
 - Classification metrics: confusion matrix, ROC, AUC etc.

So why Python?

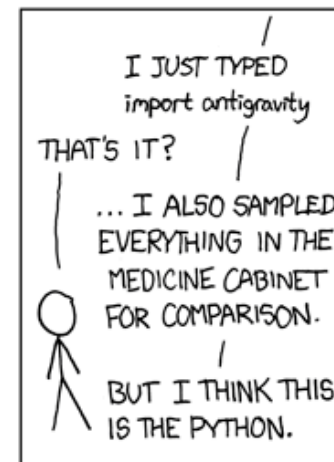
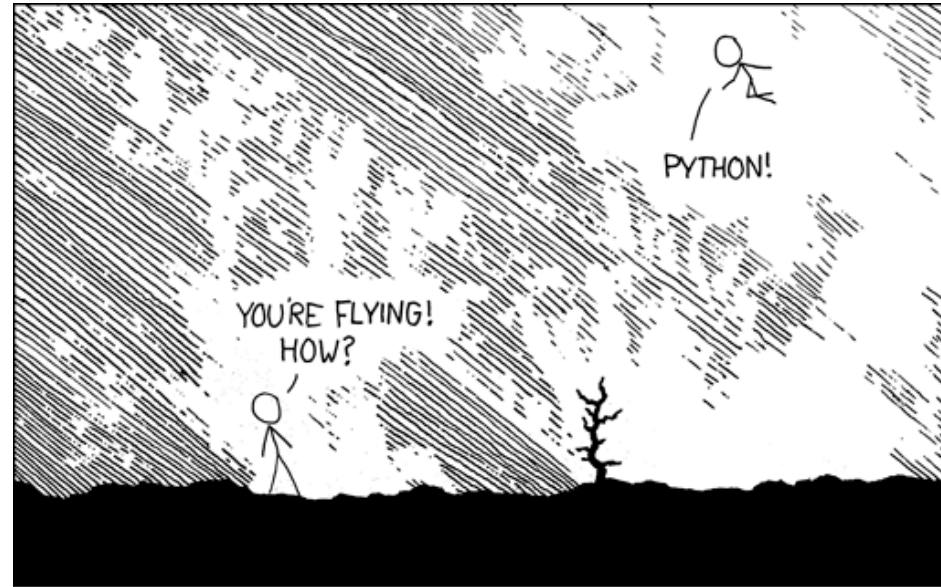
- Most popular language for ML
- #1 on [TIOBE rankings](#)

| Jul 2025 | Jul 2024 | Change | Programming Language | | Ratings | Change |
|----------|----------|--------|---|----------------------|---------|---------|
| 1 | 1 | |  | Python | 26.98% | +10.85% |
| 2 | 2 | |  | C++ | 9.80% | -0.53% |
| 3 | 3 | |  | C | 9.65% | +0.16% |
| 4 | 4 | |  | Java | 8.76% | +0.17% |
| 5 | 5 | |  | C# | 4.87% | -1.85% |
| 6 | 6 | |  | JavaScript | 3.36% | -0.43% |
| 7 | 7 | |  | Go | 2.04% | -0.14% |
| 8 | 8 | |  | Visual Basic | 1.94% | -0.13% |
| 9 | 24 | ⬆ |  | Ada | 1.77% | +0.99% |
| 10 | 11 | ⬆ |  | Delphi/Object Pascal | 1.77% | -0.12% |

So why Python?

- Easy to learn, read, and write
- Powerful libraries
- Strong and welcoming community

Python gives you superpowers



1

Poll: How well do you know Python?

- How much Python programming do you know?

a. None

b. A little bit (novice)

c. Moderate

d. Proficient

Poll: Did you take the online course?

- The one on <https://yaksh.fossee.in> ?

a. Yes

b. No

Why learn programming today?

- Why not just use GPT/Gemini/whatever?
- How can you tell if the code is correct?
- Must understand what is going on
- Solid understanding of programming model

Our approach

- Learn by doing
- High level introduction to Python
- Immersion in the tools

Important

Expect you to type along and do

Python, **python**, Python

- Language
- Interpreter
 - CPython: **python**
 - Jython: Python on the JVM
- Distribution
 - Official CPython, ActiveState Python
 - Anaconda, miniconda
 - Enthought Deployment Manager, Python(x,y) etc.

Python interpreter

- Interactive usage

```
$ python
```

```
Python 3.11.11 | packaged by conda-forge | ... [GCC 13.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```

- Executing Python code (Python scripts)

```
$ python hello.py  
Hello world!  
$
```

Python standard library

- Builtin additional functionality
 - Through Python modules
- “Batteries included”

Other community packages

- Many other open source packages
- We will use many of them
- `jupyter`, `numpy`, `scipy`
- `matplotlib`, `sklearn`, `pytorch`, `tensorflow`, etc.
- Community maintained
- Often provided in a “distribution”: conda-forge etc.

Summary: Python ecosystem

- Python Language
- Python interpreters
 - CPython: `python`
 - Jython: Python on the JVM
- Python standard library
- Community packages
- Python distributions

Programming

A famous quote¹

Programs must be written for people to read, and only incidentally for machines to execute.

A mental model

- `python` interprets each statement
- Top to bottom
- Sequentially

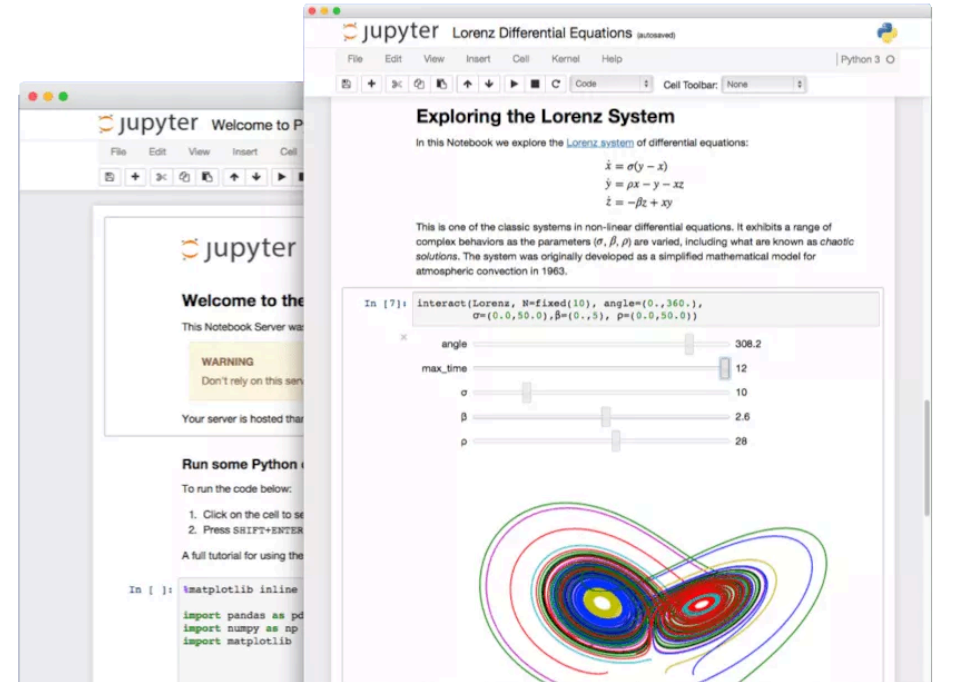
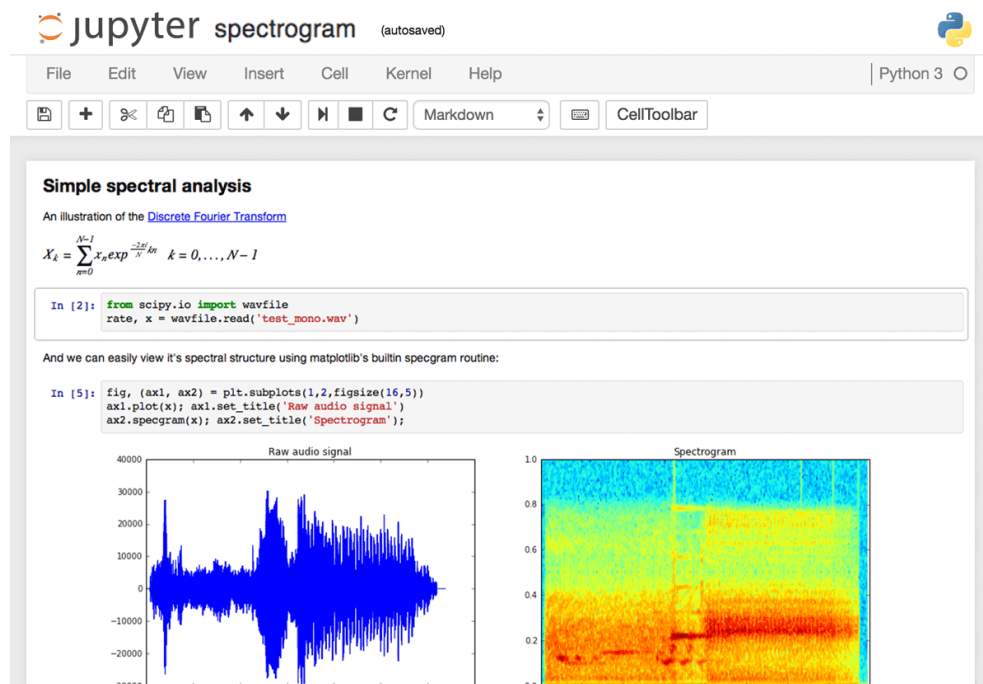


Tip

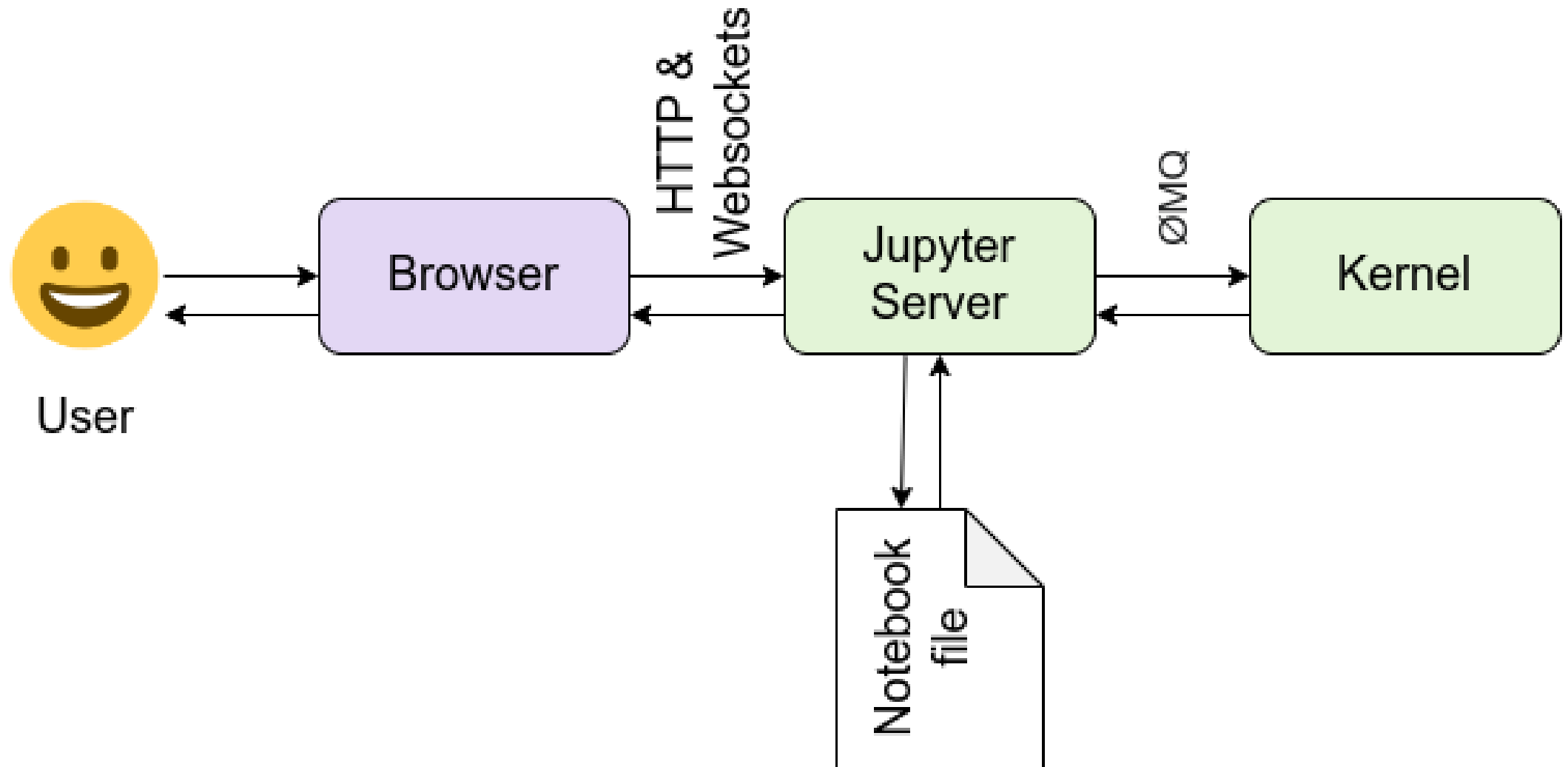
If you find programming hard, perhaps you need to provide simpler instructions?

What is Jupyter?

- Powerful interface
- Combine code, text, math
- Interactive widgets



How does it work?



How does it work?

- Kernels manage execution and communication
 - Supports 40 different languages
 - Python, Julia, R, Scala, Java, Octave, Scheme, Cling ...
-
- Notebook is an interface that talks to the kernel

Alternatives

- Jupyterlite: on your browser
- Google colab: on their servers
- Alternatives: [Marimo](#)

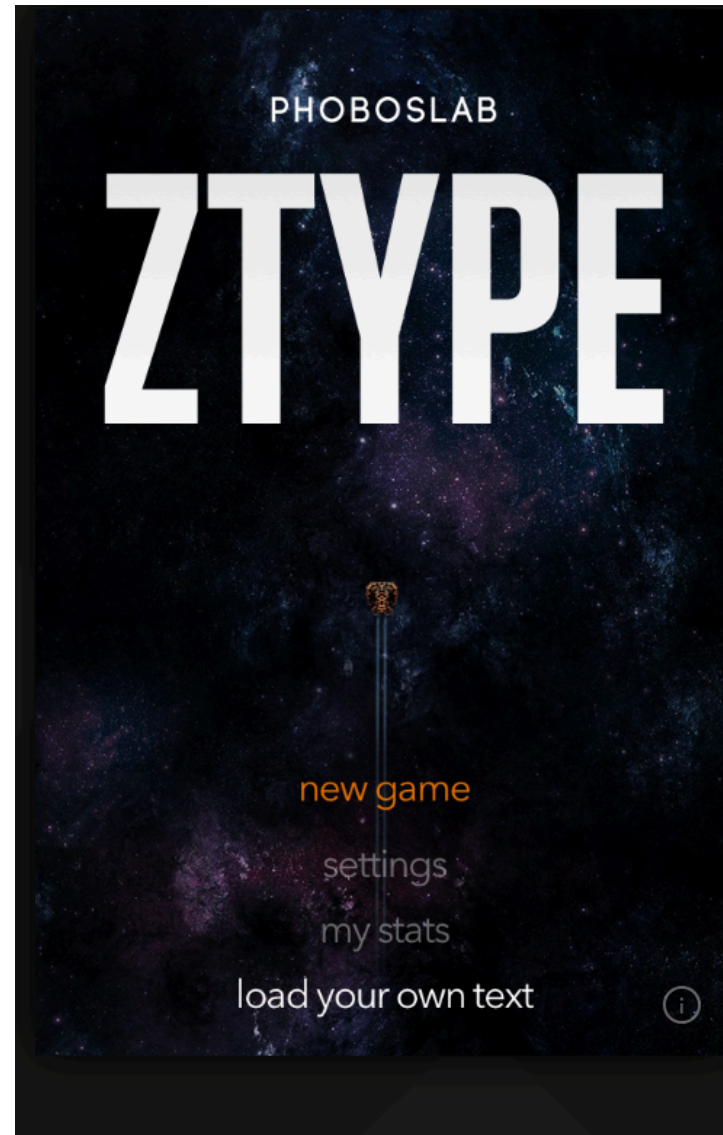


Some reference texts

- [Python data science handbook](#)
- [Learning Data Science](#)
- [Veridical Data Science](#)
- [Computational and Inferential thinking](#)

Before we start

- Get comfortable with
 1. Keyboard!
 2. Code editor
- Practice
- <https://zty.pe>
- Try to clear wave 10



Questions?

Poll: What OS do you use?

What OS do you use on your programming environment for this course? No phone/tablet.

- a. Windows
- b. MacOS
- c. Linux
- d. Other

Let us dive in!

- Visit here: prabhuramachandran.github.io/pml
- Provides a live environment
 - Tested on recent Firefox or Chrome
 - Don't use a private window
- Will have the class slides/notebooks etc.
- Avoid editing files in [slides](#), [tutorials](#)