



ENGR 21:

Computer Engineering Fundamentals

Lecture 8
Thursday, September 25, 2025

Introducing `numpy`



What is numpy?

<https://numpy.org/>
<https://github.com/numpy/>

- Numerical Computing in Python
 - Gives Python the capabilities of MATLAB
 - But less \$\$
- Free & Open Source
- Lives on github
- A package that you must import
 - Does not come pre-installed with Python installation
- Much faster than regular Python
- Optimized for scientific + engineering calculations

Features of the numpy package

- Data types for numerical computing
 - float16
 - float32
 - float64
 - int8, int16, int32, int64
 - uint8, uint16, uint32, uint64
- Vast library of functions useful for engineering
- Everything is 'array-native' –
 - works seamlessly on large sets of numbers

```
>>> import numpy

>>> import math

>>> numpy.linspace(1,5,9)

array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. ,
       4.5, 5. ])

>>> numpy.logspace(1,3,3)

array([ 10., 100., 1000.])

>>> math.sqrt(range(1,10))

TypeError: must be real number, not range

>>> numpy.sqrt(range(1,10))

array([1.          , 1.41421356, 1.73205081,
       2.          , 2.23606798,
       2.44948974, 2.64575131, 2.82842712, 3.
       ])
```

Introducing `pip`, Python's native package manager

Installing Packages for Python

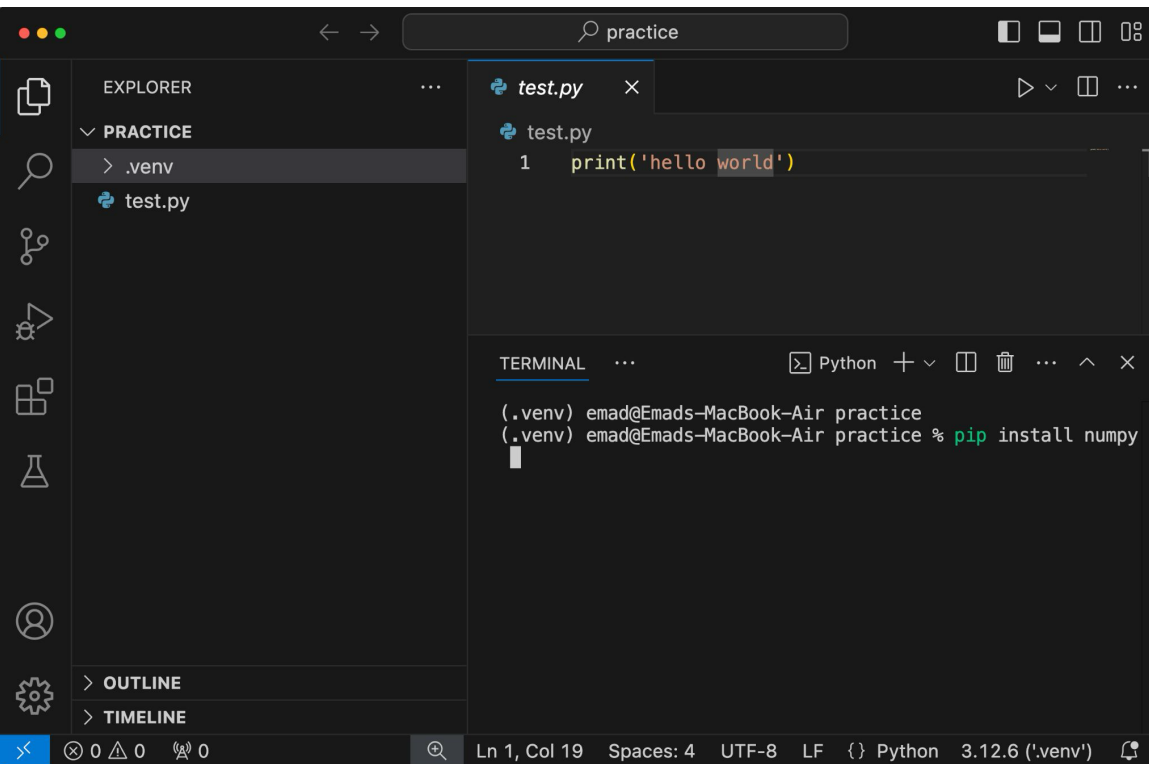
Inside a Terminal (NOT Python REPL):

```
pip install <package name>
```

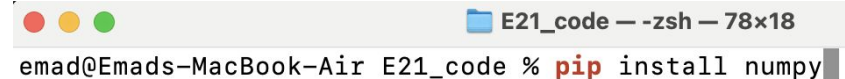
numpy

pip3

Options for Terminals:



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project named 'PRACTICE' with a subdirectory '.venv' and a file 'test.py'. The main editor area shows 'test.py' with the code: `1 print('hello world')`. Below the editor is a terminal window titled 'TERMINAL' with a Python icon. The terminal shows the command `pip install numpy` being executed in a virtual environment `(.venv)`. The status bar at the bottom indicates the file is at Line 1, Column 19, using UTF-8 encoding, and the Python interpreter is set to 3.12.6 in the `(.venv)` environment.

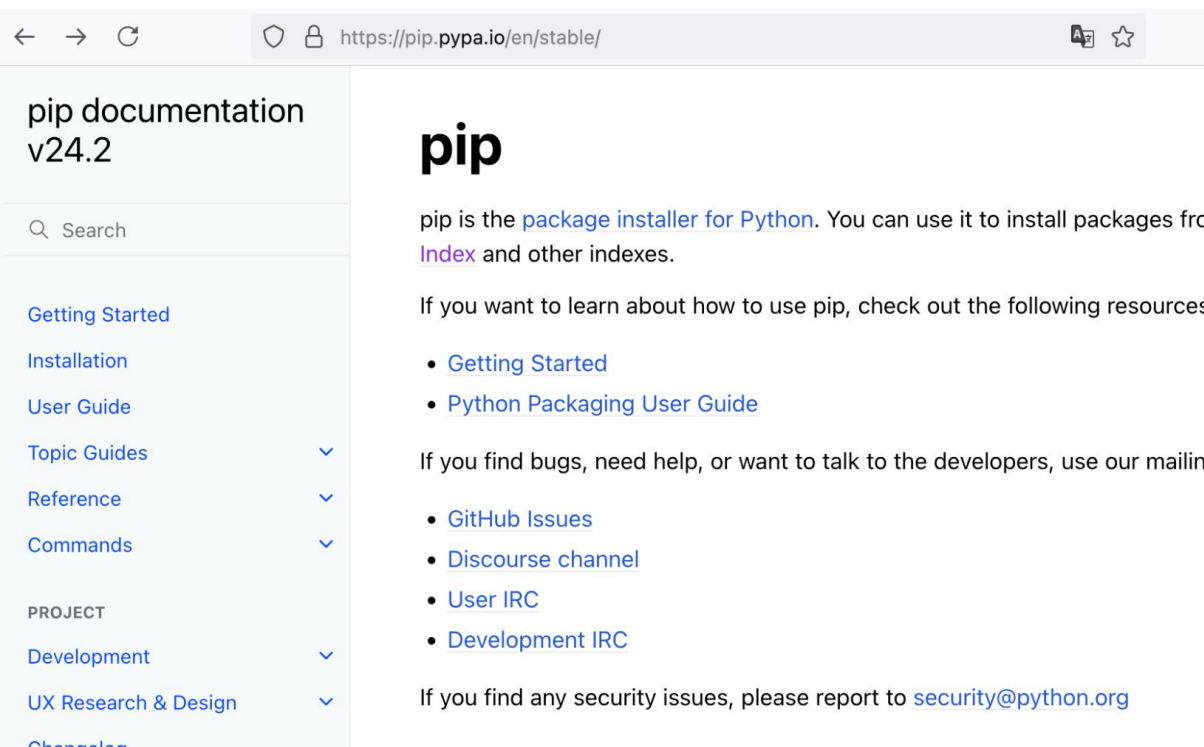


The screenshot shows a terminal window titled 'E21_code --zsh-- 78x18'. The prompt is `emad@Emads-MacBook-Air E21_code %` and the command `pip install numpy` is being entered.

What does pip do, exactly?

A 'package manager' for Python

Looks up packages on the official **Python Package Index** <https://pypi.org/>



pip documentation v24.2

Search

Getting Started

Installation

User Guide

Topic Guides

Reference

Commands

PROJECT

Development

UX Research & Design

Changelog

pip

pip is the [package installer for Python](#). You can use it to install packages from [Index](#) and other indexes.

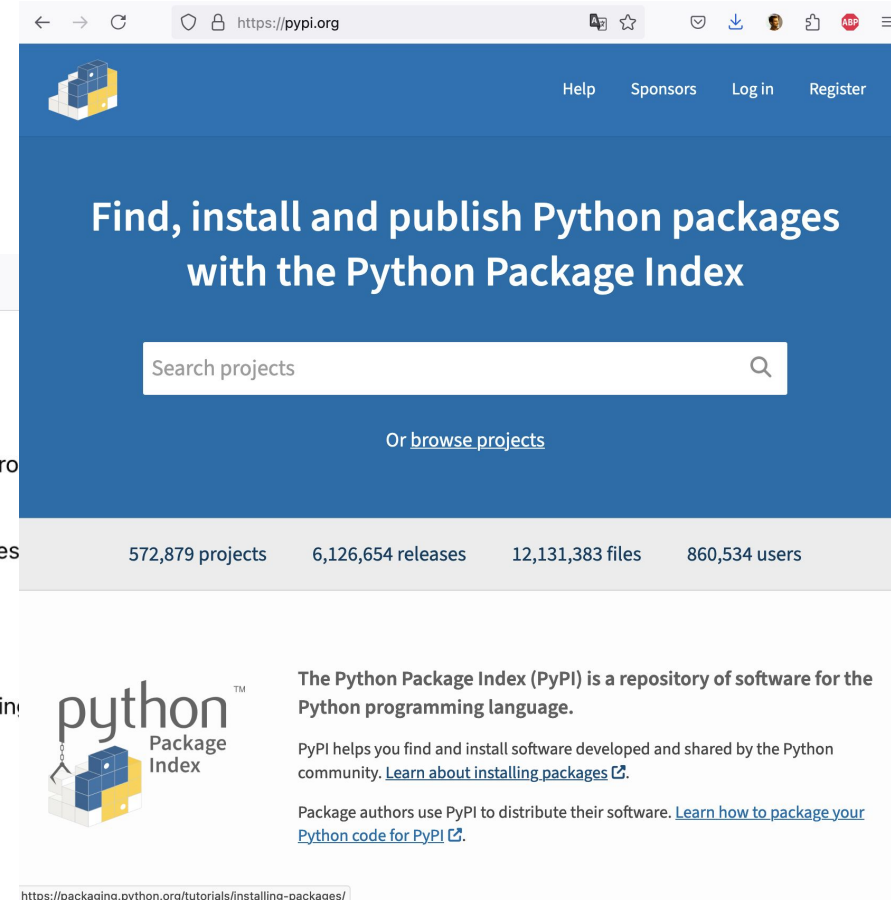
If you want to learn about how to use pip, check out the following resources

- [Getting Started](#)
- [Python Packaging User Guide](#)

If you find bugs, need help, or want to talk to the developers, use our mailing

- [GitHub Issues](#)
- [Discourse channel](#)
- [User IRC](#)
- [Development IRC](#)

If you find any security issues, please report to security@python.org



Help Sponsors Log in Register

Find, install and publish Python packages with the Python Package Index

Search projects

Or [browse projects](#)

572,879 projects 6,126,654 releases 12,131,383 files 860,534 users

python™
Package Index

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#).

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#).

<https://packaging.python.org/tutorials/installing-packages/>



Some useful pip commands

```
pip3 install <package name>  
pip3 list  
pip3 uninstall <package name>
```




Install numpy on your computer

1. Run `pip3 install numpy` from a terminal
 - a. Either inside VS Code or in your operating system's terminal.
2. Check that numpy works
 - a. Enter Python REPL (enter `python3` into a terminal window)
 - b. Enter `import numpy` at the REPL

Back to `numpy`

Conventions for numpy

Three approaches to importing the package

Write at the top of any Python file that will use numpy one of the following
each time code runs

```
import numpy
```

Must use `numpy.<function>`

```
from numpy import *
```

imports all functions

Can simply use `<function>`

```
import numpy as np
```

Must use `np.<function>`

Only a convention!
Can choose any name

Why does it matter how packages are imported?

Three choices:

```
import numpy
```

```
from numpy import *
```

```
import numpy as np
```

```
>>> import math  
>>> import numpy
```

from 'standard library'

```
>>> sqrt(4)  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
NameError: name 'sqrt' is not defined
```

```
>>> math.sqrt  
<built-in function sqrt>
```

```
>>> numpy.sqrt  
<ufunc 'sqrt'>
```

```
>>> math.sqrt(4)  
2.0 → type : 'float'
```

```
>>> numpy.sqrt(4)  
np.float64(2.0) → type 'np.float64'
```

Calculations in numpy are:

- ❖ More efficient
- ❖ More scalable
- ❖ Faster

Compared to regular Python

Arrays in numpy

- Like a python list, but better
- Can be multidimensional



■	□	□	■	□	□
□	■	□	□	■	■
■	■	□	■	□	□
□	□	■	□	■	■
■	□	■	□	□	■
□	■	□	■	■	□

Native Python approach:

```
[True, False, False, True, False, False,
False, True, False, False, True, True,
True, True, False, True, False, False,
False, False, True, False, True, True,
True, False, True, False, False, True,
False, True, False, True, True, False]
```

Numpy approach

```
array([[ True, False, False,  True, False, False],
       [False,  True, False, False,  True,  True],
       [ True,  True, False,  True, False, False],
       [False, False,  True, False,  True,  True],
       [ True, False,  True, False, False,  True],
       [False,  True, False,  True,  True, False]])
```

Or, we could have made a list of lists ...

Floating-point numbers

Recall: How do we write non-integers?

Decimal Numbers	21.5679 $\dots 10^1 10^0 10^{-1} 10^{-2} 10^{-3} \dots$
Binary Numbers	101.10101 $\dots 2^1 2^0 2^{-1} 2^{-2} \dots$

Task: Interpret the following binary numbers:

Binary	Decimal / Fraction
1.1101101001 $2^{-1} + 2^{-2} + 2^{-4} + \dots$	$\frac{1897}{1024} \approx 1.85$
1.0010011000	

$$1 + \frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{16} + \frac{1}{32} + 0 + \frac{1}{128} + 0 + 0 + \frac{1}{1024}$$

By how much do binary fractions increment?

Preliminary exercise:

1.796
1.797 } bigger by $\frac{1}{1000}$

Binary	Decimal
1.0010011000	$1\frac{19}{128}$
1.0010011001	bigger by $\frac{1}{1024}$

$$\frac{1}{8} + \frac{1}{64} + \frac{1}{128} = \frac{16}{128} + \frac{2}{128} + \frac{1}{128} = \frac{19}{128}$$

$$\frac{1}{8} + \frac{1}{64} + \frac{1}{128} + \frac{1}{1024}$$

The IEEE Standard for Floating-Point Binary Numbers

1.23×10^{79}

+/-

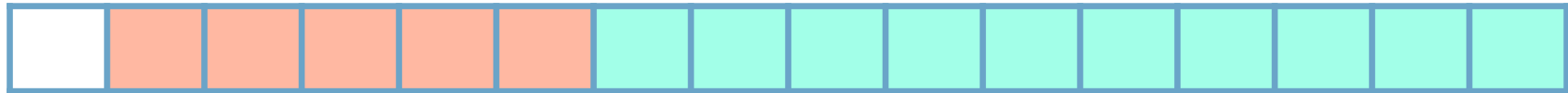
1.

Significand

$\times 2$

Exponent

Bias.
Subtract bias from
exponent to get
true value.



Size	Sign	Significand	Exponent	Bias	Colloquial Name
16-bit	1	10	5	15	Half precision
32-bit	1	23	8	127	Single precision
64-bit	1	52	11	1023	Double precision

16-bit (half)

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 = 0x4248

$1 \times 2^1 \times 1.571 = 3.141$

The Exponent Bias

The exponent bias is introduced to allow floating-point numbers to have both positive and negative exponents on the “2”.

16-bit (half)

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
0 1 0 0 1 0 0 1 0 1 0 0 1 0 1 1 = 0x494B

$$1 \times 2^3 \times 1.324 = 10.59$$

10010 :- 18

16-bit (half)

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
0 0 1 0 1 0 0 1 0 1 0 0 1 0 1 1 = 0x294B

$$1 \times 2^{-5} \times 1.324 = 0.04136$$

$$2^0 \times 0 + 2^1 \times 1 + 2^2 \times 0 + 2^3 \times 1 = +2 + 8$$

01010 :- 10

Practice reading the IEEE Standard for Floating-Point Binary Numbers

$$\boxed{} 1. \boxed{} \times 2^{\boxed{}}$$

0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

- 1) Interpret exponent bits as an integer.
- 2) Subtract bias from it to get exponent
- 3) Interpret significand as the binary digits in the number $1.\underline{\hspace{1cm}}$
- 4) Multiply significand with 2^{exponent}
- 5) Interpret sign bit as: $\begin{cases} \text{negative if } 1 \\ \text{positive if } 0 \end{cases}$

Exponent Bias:
Subtract 15 from exponent

16-bit (half)

16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
 0 1 0 0 0 0 1 0 0 1 0 0 1 0 0 0 = 0x4248

$$1 \times 2^1 \times 1.571 = 3.141$$

How much do 16-bit binary floats increment? v.2

Calculate the value of this number

$$1 \frac{331}{1024}$$

0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$$2^4 + 2^1$$

$$= 16 + 2 = 18$$

subtract bias

Exponent 3

$$2^{-2} + 2^{-4} + 2^{-7} + 2^{-9} + 2^{-10}$$

$$\frac{1}{4} + \frac{1}{16} + \frac{1}{128} + \frac{1}{512} + \frac{1}{1024}$$

$$= \frac{256}{1024} + \frac{64}{1024} + \frac{8}{1024} + \frac{2}{1024} + \frac{1}{1024}$$

$$= \frac{331}{1024}$$

Combine :

$$\left[1 \frac{331}{1024} \times 2^3 \right] = \frac{1024 + 331}{1024} \times 2^3 = \frac{1355}{128} \approx 10.5859$$

How much do 16-bit binary floats increment? v.2

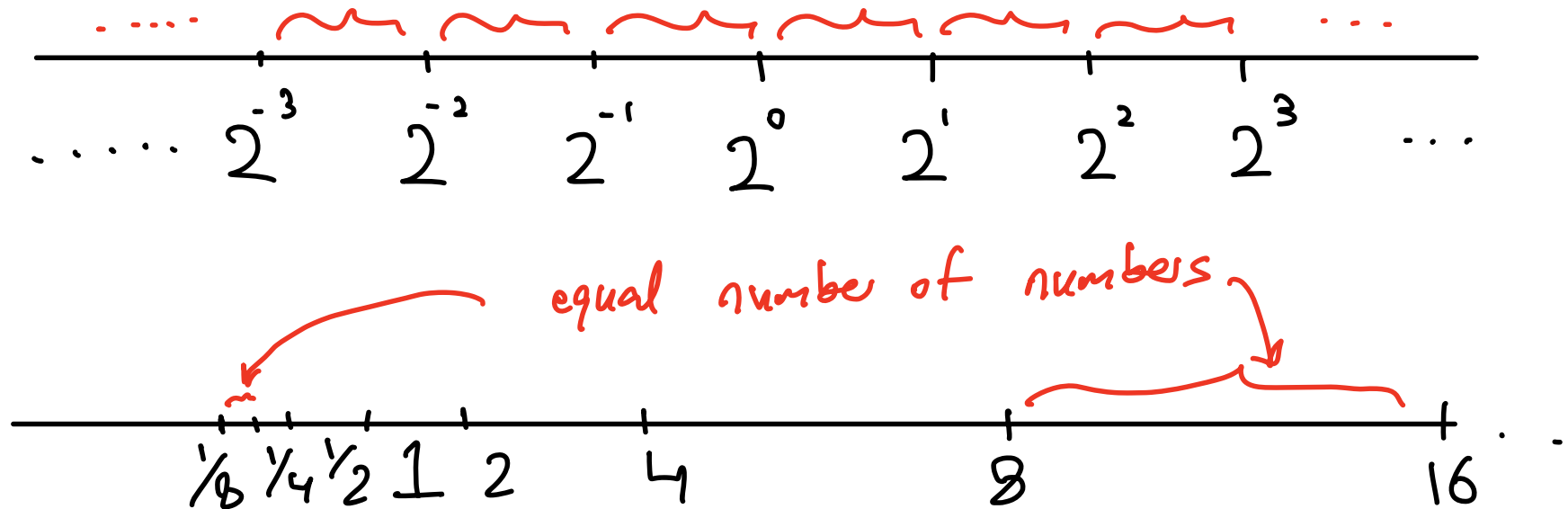
0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	1	10.5859
0	1	0	0	1	0	0	1	0	1	0	0	1	1	0	0	10.5938

Calculate the value of the next number

Difference ≈ 0.01 ?

Gaps on the number line

Floating-point system assigns an equal number of numbers to each interval shown below





Machine Epsilon for a floating-point number system

The difference between 1
and the next higher
floating-point number