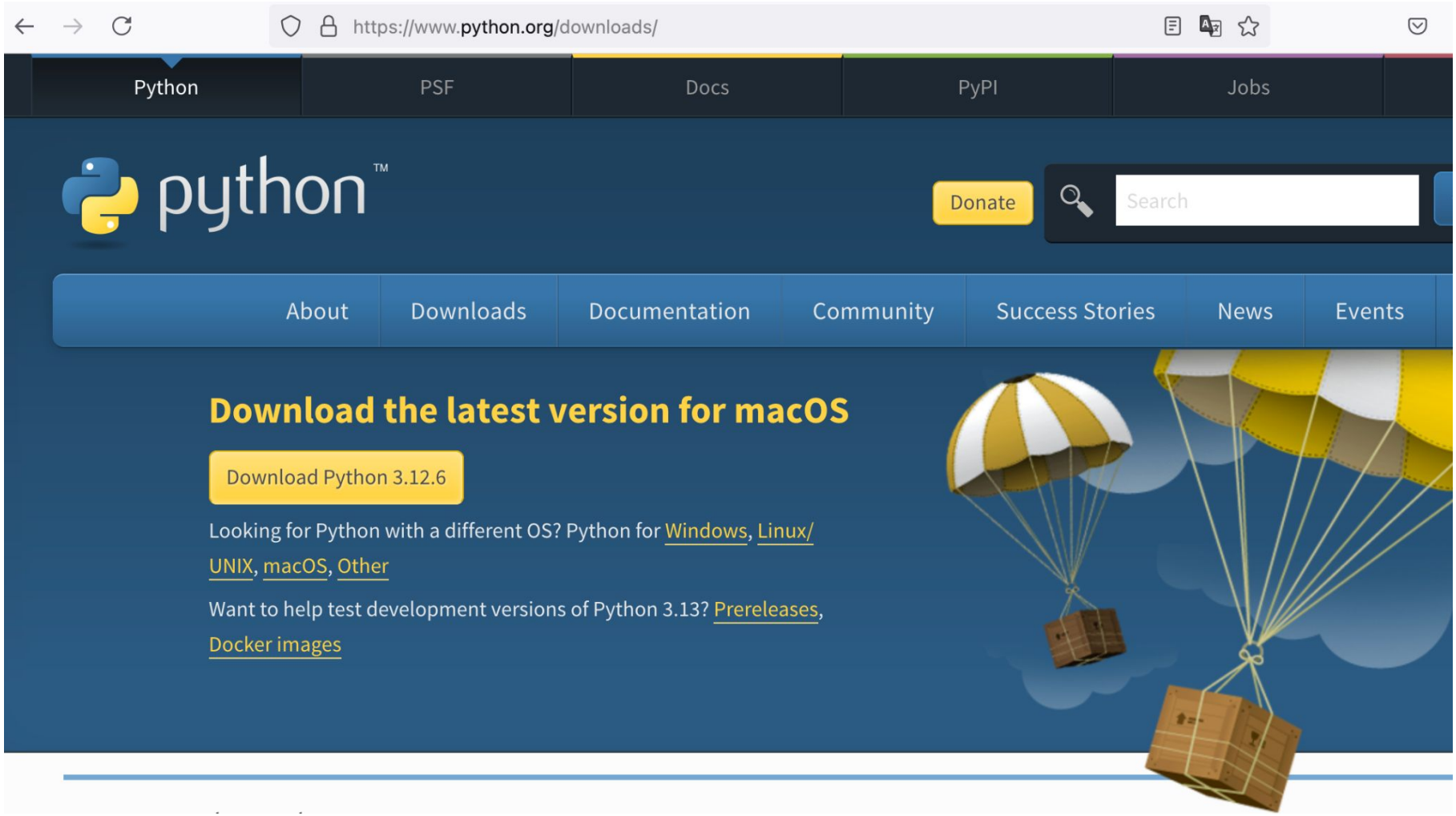
ENGR 21:
Computer Engineering Fundamentals

Lecture 7
Tuesday, September 23, 2025

Review of Python Installation

https://www.python.org/downloads/

Installing Python on your computer



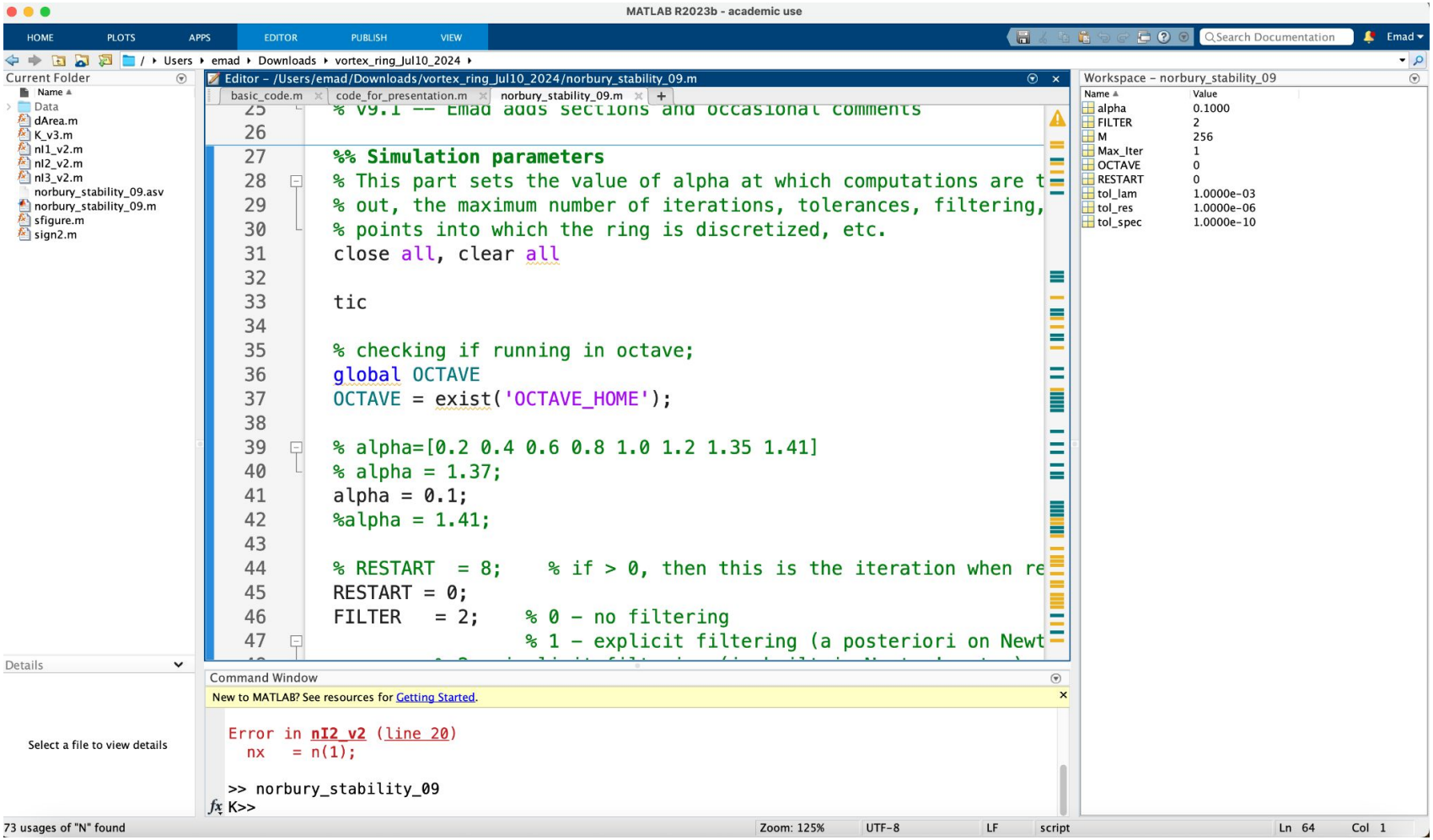
Choice of IDE (Integrated Development Environment)

What's an IDE?

- A program that lets you interface with a programming language
- Usually a "visual" interface
- Multiple IDEs can be installed; they will use the same underlying programming language

IDEs you (may) have seen before

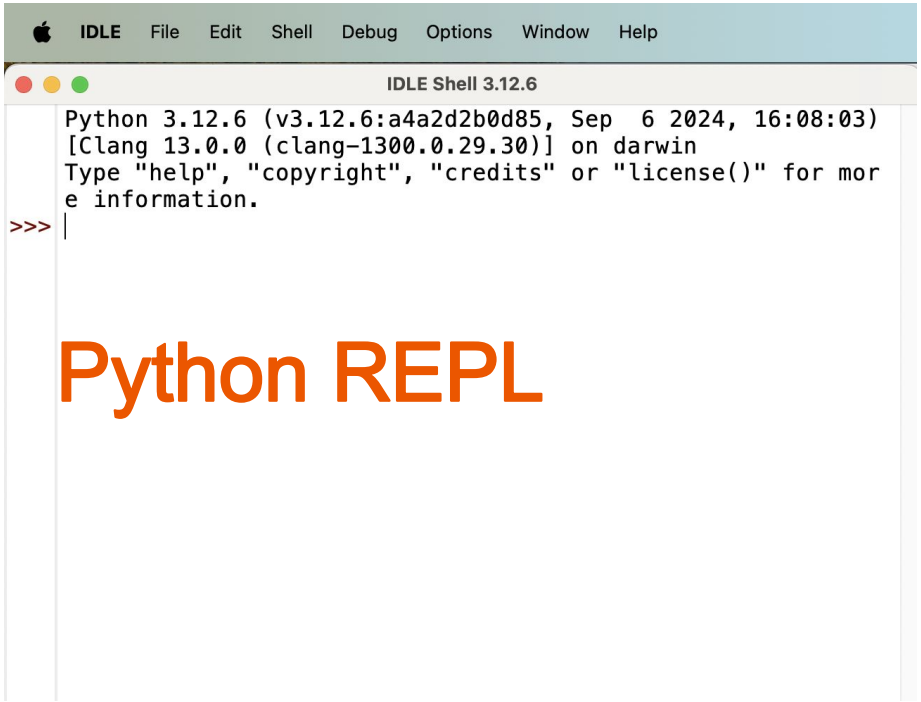
MATLAB has a built-in IDE

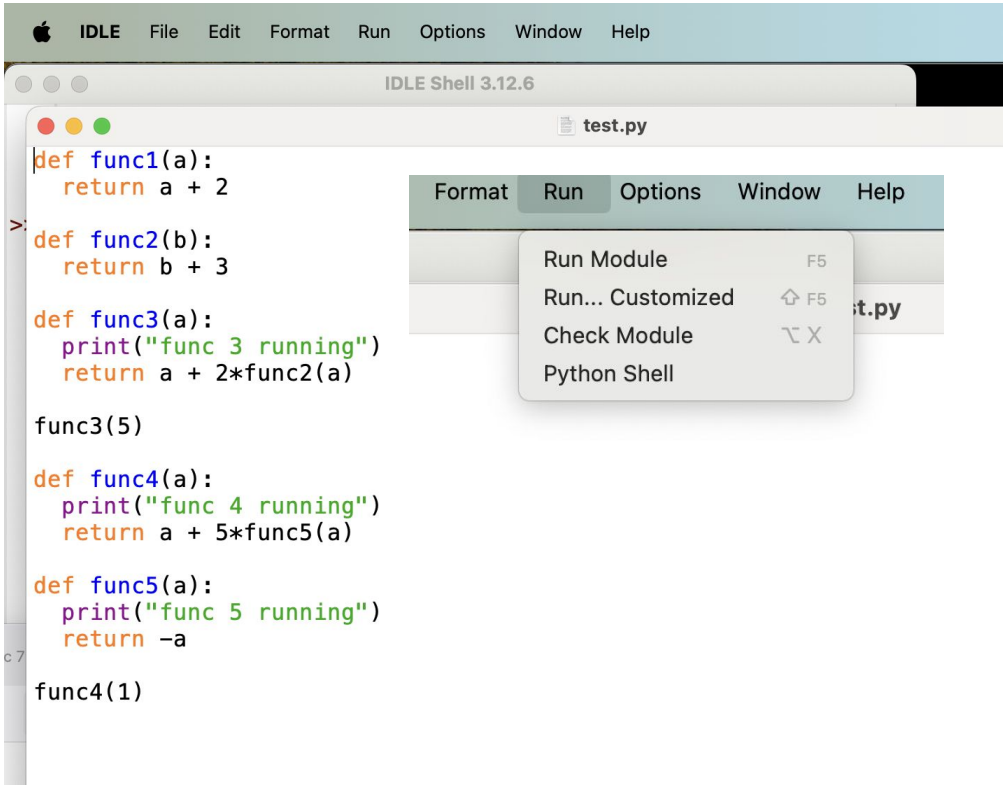


The simplest IDE for Python: IDLE (Integrated Development and Learning Environment)

Comes pre-installed with Python if you get it from www.python.org/downloads

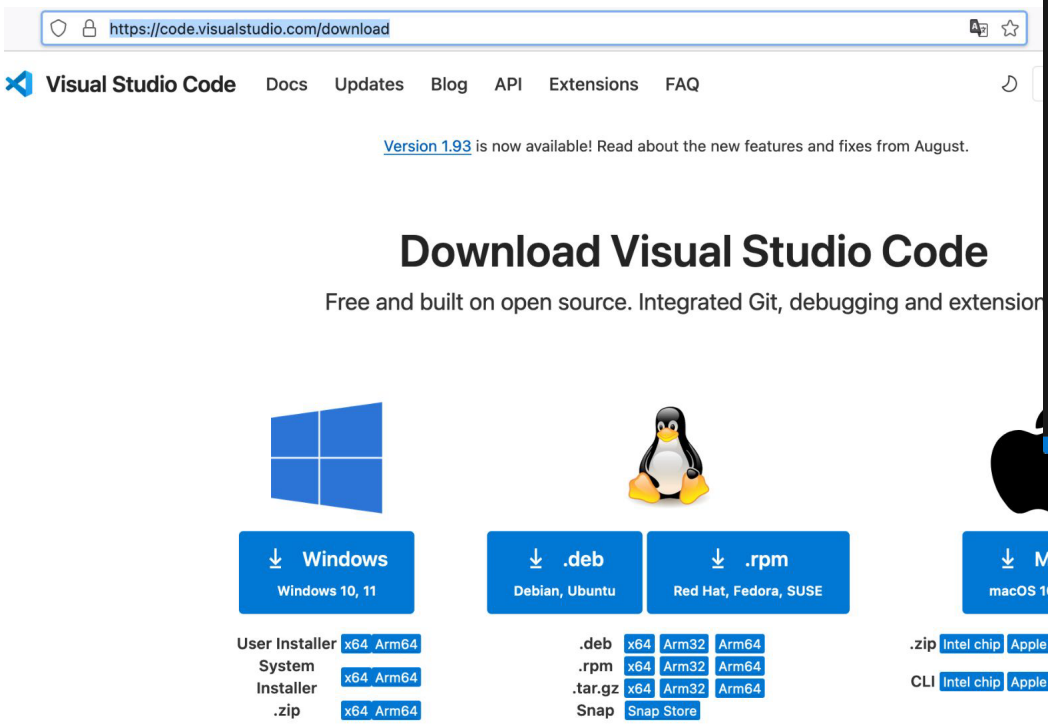
Search for 'IDLE' in start menu or Launchpad

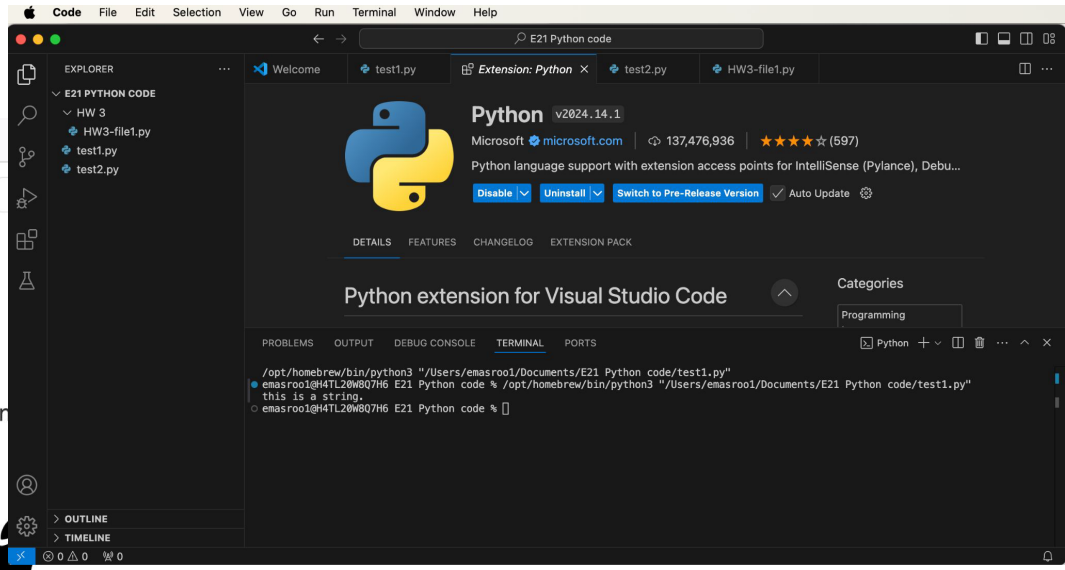




Visual Studio Code (VS Code)

- Download & Install VS Code
- Install Python Extension for VS Code

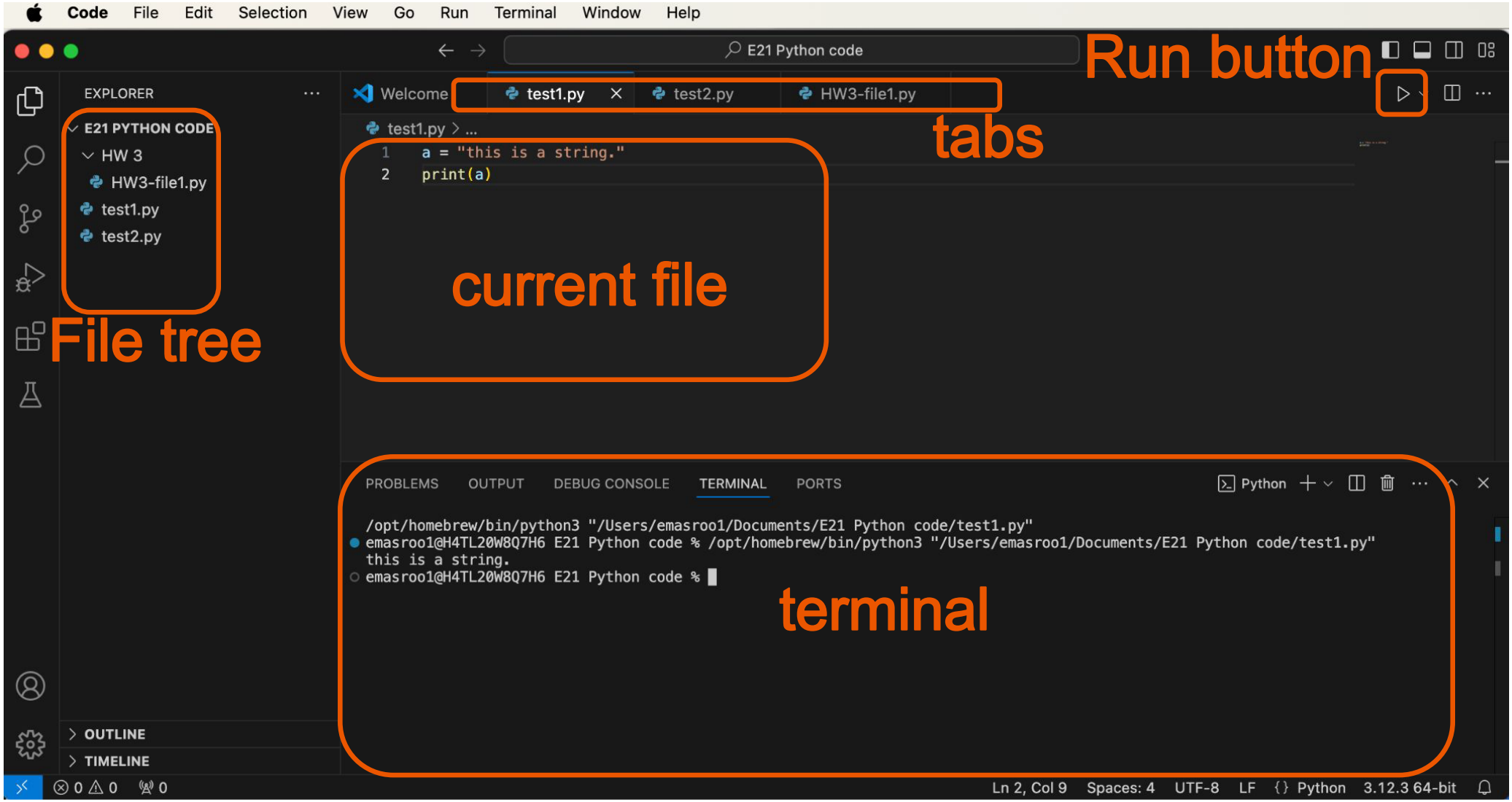




https://code.visualstudio.com/download

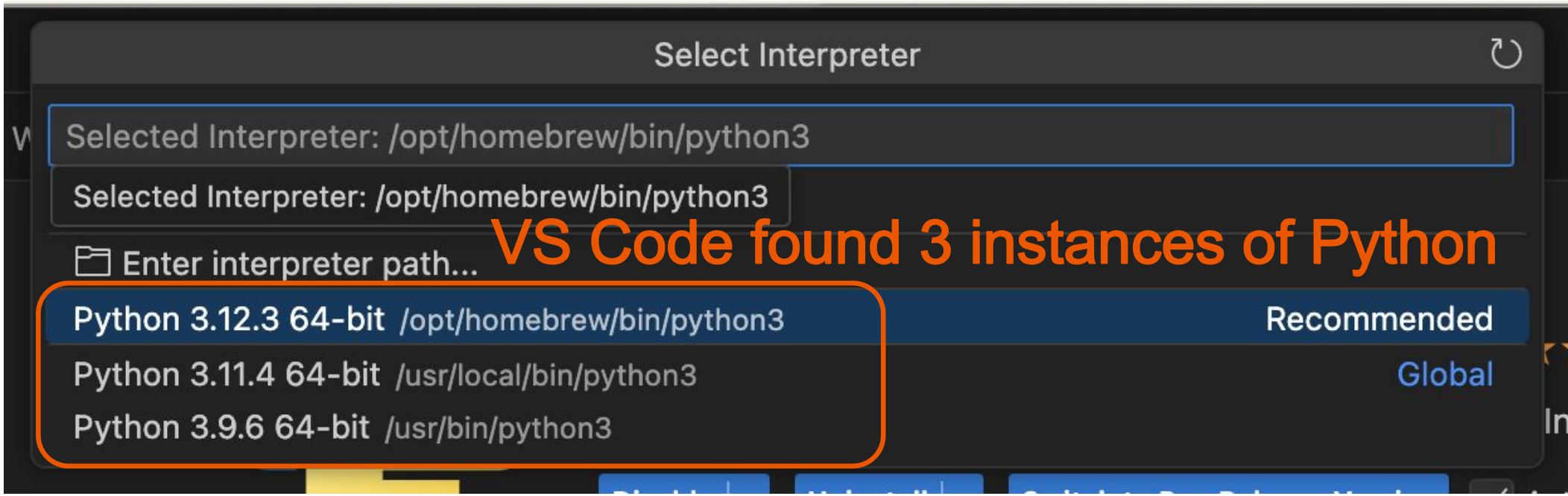
https://marketplace.visualstudio.com/

Anatomy of VS Code



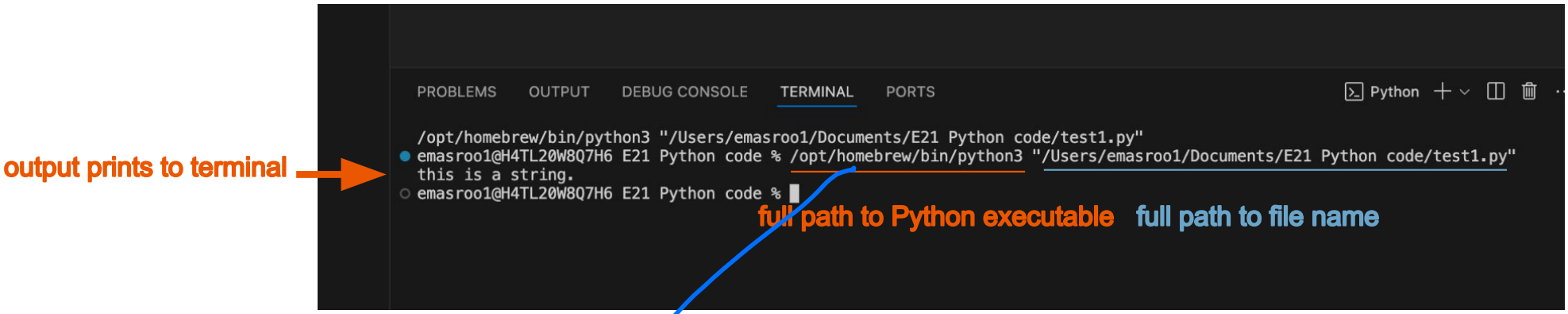
Python Versions inside VS code

- You may have more than one installation of Python on your computer
- How to "tell VS Code which one to use"
  - View → Command Palette → "Python: select Interpreter"



Coding ‘on the command line’

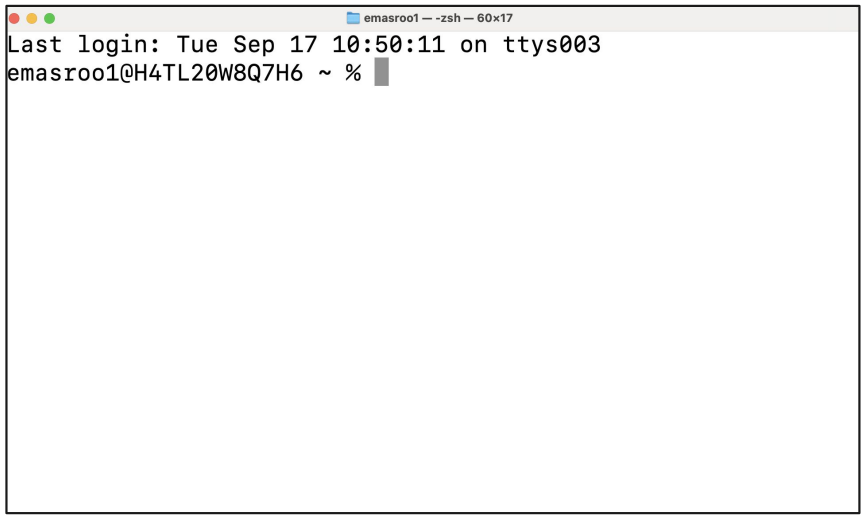
A closer look at “the terminal”



Terminals & Command-Line Interfaces today

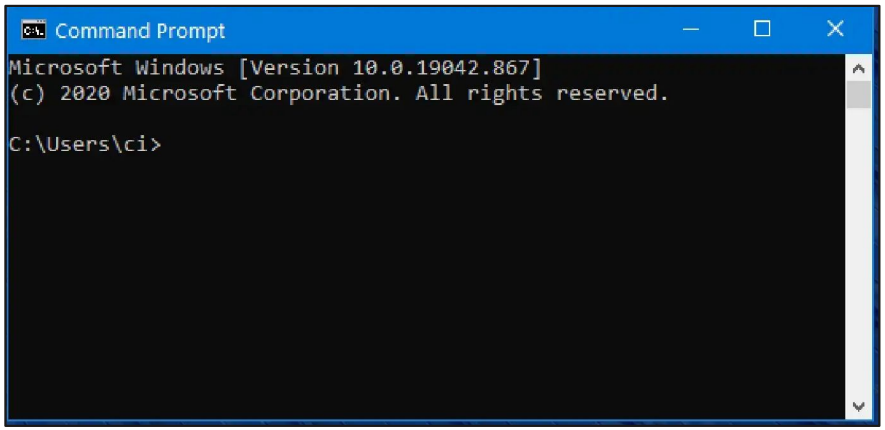
Mac

Terminal



Windows

Command Prompt, aka **`cmd.exe`**



Upgraded to Powershell

How to run a Python program via command-line interface

- Open Terminal or Powershell.
- Navigate to location of *.py file (try `cd '/full/path/to/folder'`)
-

Resources: Copy file for today

```
emasroo1@iMac Documents % python test1.py
No such file or directory
emasroo1@iMac Documents % pwd
/Users/emasroo1/Documents
```

file test1.py is in Downloads, but you are in Documents

'change directory'

either

```
emasroo1@iMac Documents % cd /Users/emasroo1/Downloads
emasroo1@iMac Downloads % python test1.py
this is a string
```

gives full path

or

```
emasroo1@iMac Documents % python /Users/emasroo1/Downloads/test1.py
this is a string
```

Command-Line Arguments

- Information passed at runtime to a Python program
- Similar to functions
- Arguments interpreted as strings by default

```
emasroo1@iMac Documents % python test1.py hello
emasroo1@iMac Documents % python test1.py 'number 1'
emasroo1@iMac Documents % python test1.py 40 num
```

1 CL argument, 'hello'

1 CL argument, 'hello 1'

2 CL arguments, '40' and 'num'

- How do you access command-line arguments inside the program?
- `sys.argv` is a list containing the CL arguments as strings
- `sys.argv[0]` is the file name

argument vector

test1.py ?

```
import sys

a = sys.argv[1]
```

Equivalence of Command-Line Arguments and Function Arguments Approach

```
>>> def double(x):
      return 2*x

>>> double(5)
10
```

```
# file script1.py
import sys

num = int(sys.argv[1])
print(2*num)
```

```
python3 script1.py 5
10
```

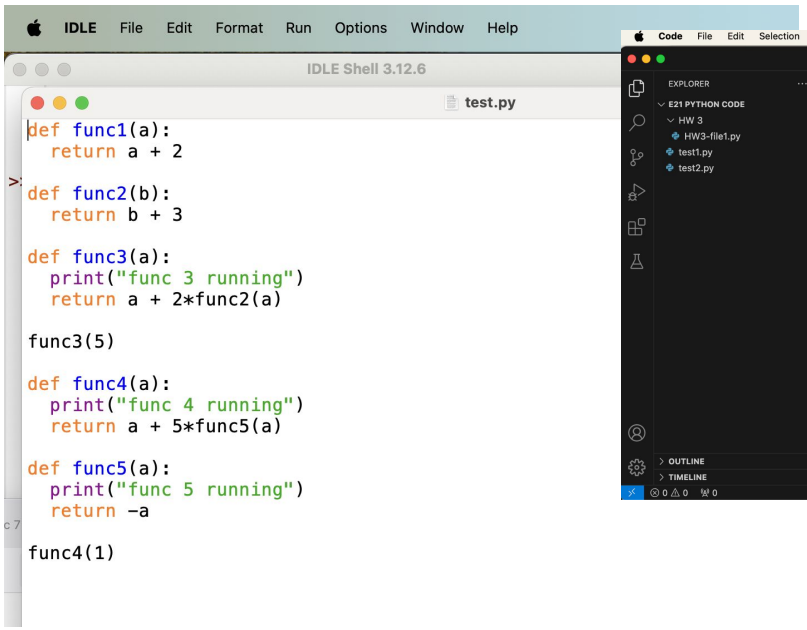
The same functionality can be achieved by using

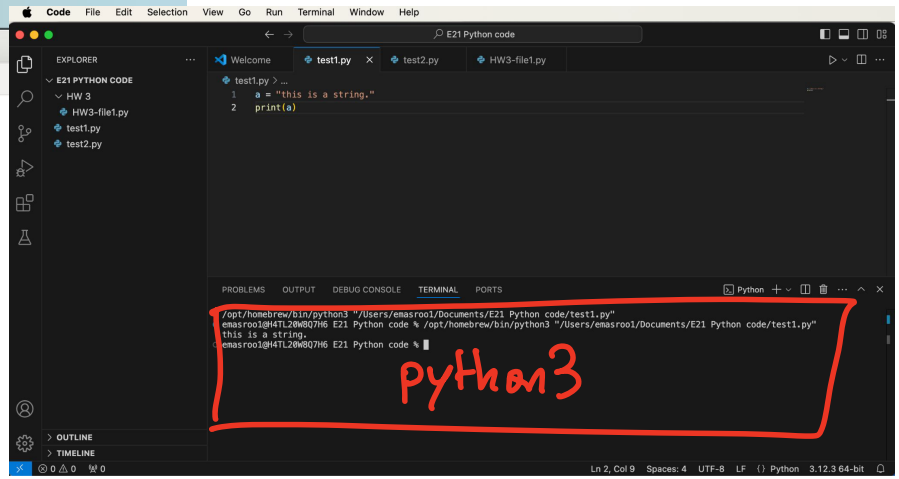
1) CL arguments to a Python script
2) Arguments to a Python function

Pretty much any function can be re-written as a script that accepts command-line arguments.

Three approaches to writing code

	IDLE	VS Code	Text Editor +terminal
edit files	Open a separate IDLE window for each *.py file	Tabs + folder tree to organize *.py files	Use notepad, Vim, textedit, gedit, nano, etc.
interactive	REPL window	Open REPL from within VS code	Type 'python' or 'python3'



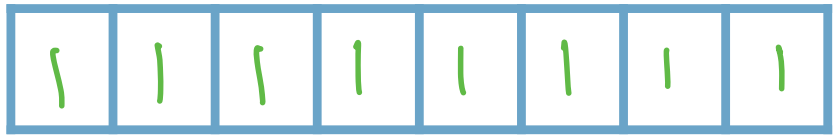


.py extension is only a convention

Floating-point system

What range of natural numbers can a computer store?

- With 8-bit binary numbers



biggest positive number available : $2^8 - 1$

$999 : 10^3 - 1$

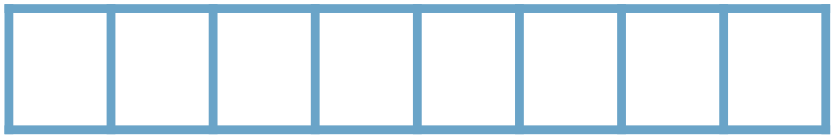
The largest possible 8-bit binary integer:

.... 0 or 1 0 or 1 0 or 1

$\underbrace{2 \times 2 \times \cdots \times 2}_{8 \text{ times}}$ choices : 2^8

What about positive & negative integers?

- With 8-bit binary numbers



Some options:

- 0 to 255
- -1 to 254
- -2 to 253
- -10 to 245
- -128 to 127

Magnitude of largest 8-bit number is ?

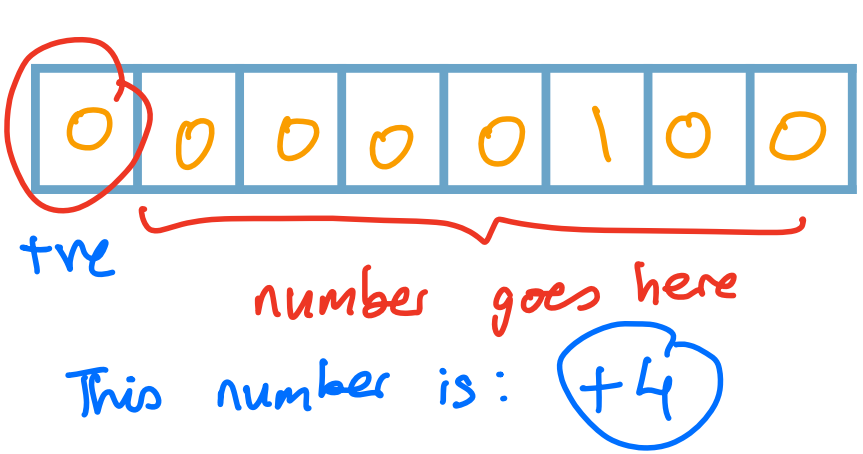
256 options.

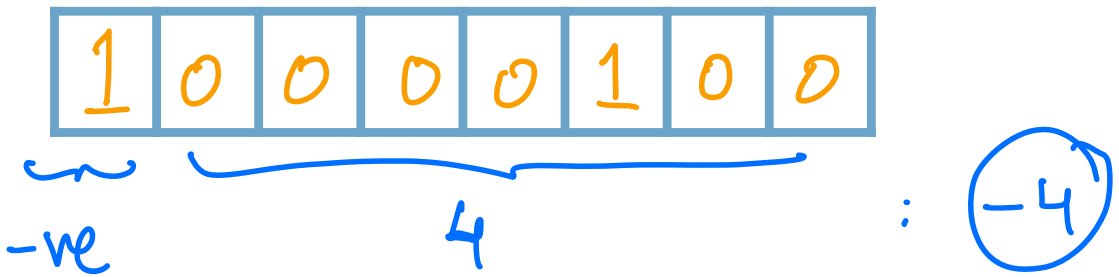
unevenly distributed around 0

evenly distributed around 0

"Sign-Magnitude" Representation: One way that signed integers can be stored in computers

- The most significant bit is reserved for indicating the sign of the number
- `0 → positive, 1 → negative`

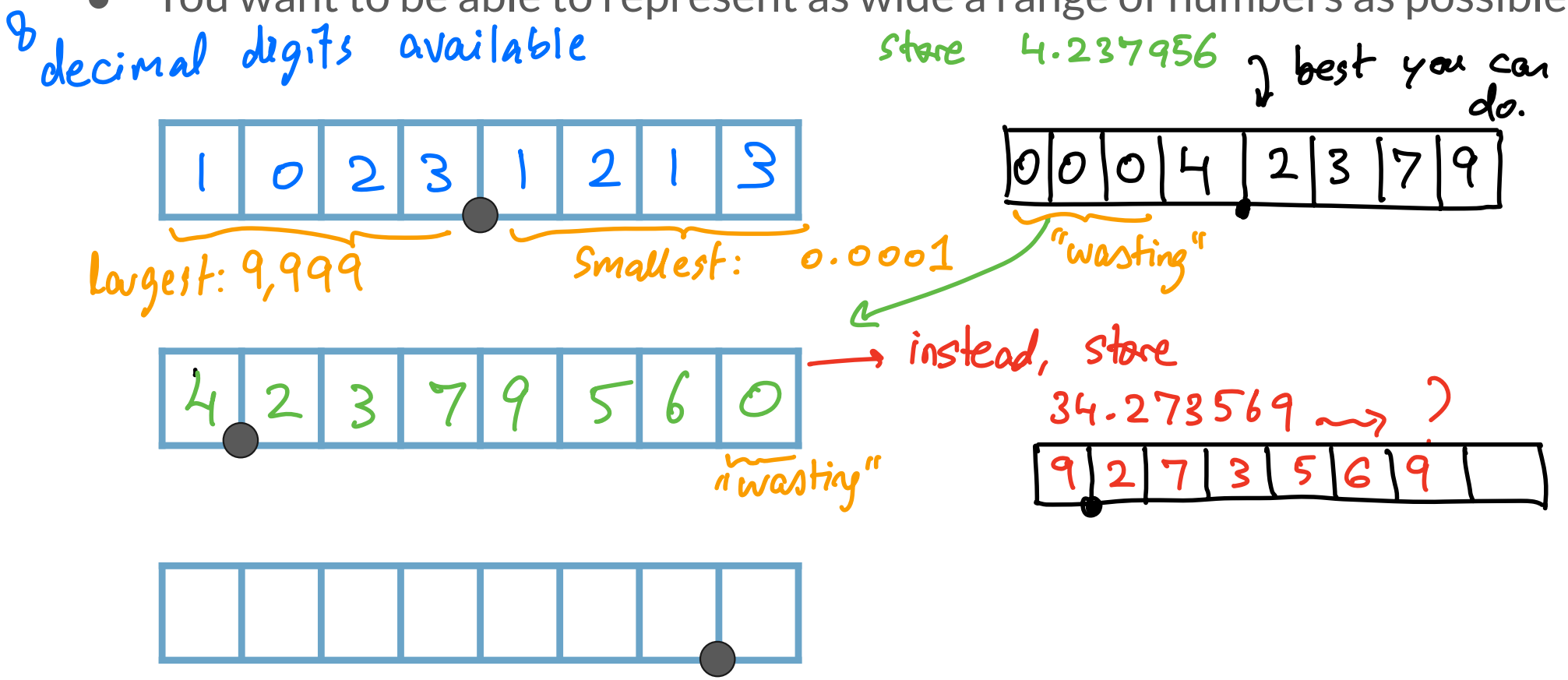




Different ways of storing "real" numbers: Fixed-Point

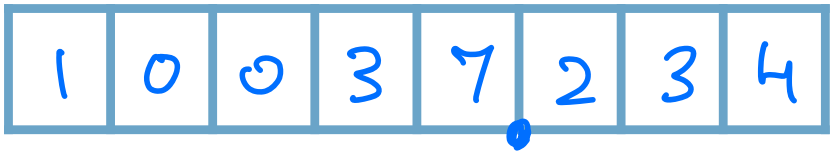
"decimal point"

- Suppose you only have room for 8 decimal digits to store a number.
- You want to be able to represent as wide a range of numbers as possible.



The Floating-Point idea (applied to decimal numbers)

- The “decimal” point floats i.e., it can move places



- Need to store:
  - The digits
  - The location of the point

"Scientific notation" for numbers

$$-2.34 \times 10^{5}$$

from 0 to 9 } or
-5 to 4 }

Choose by convention

Information conveyed by this:

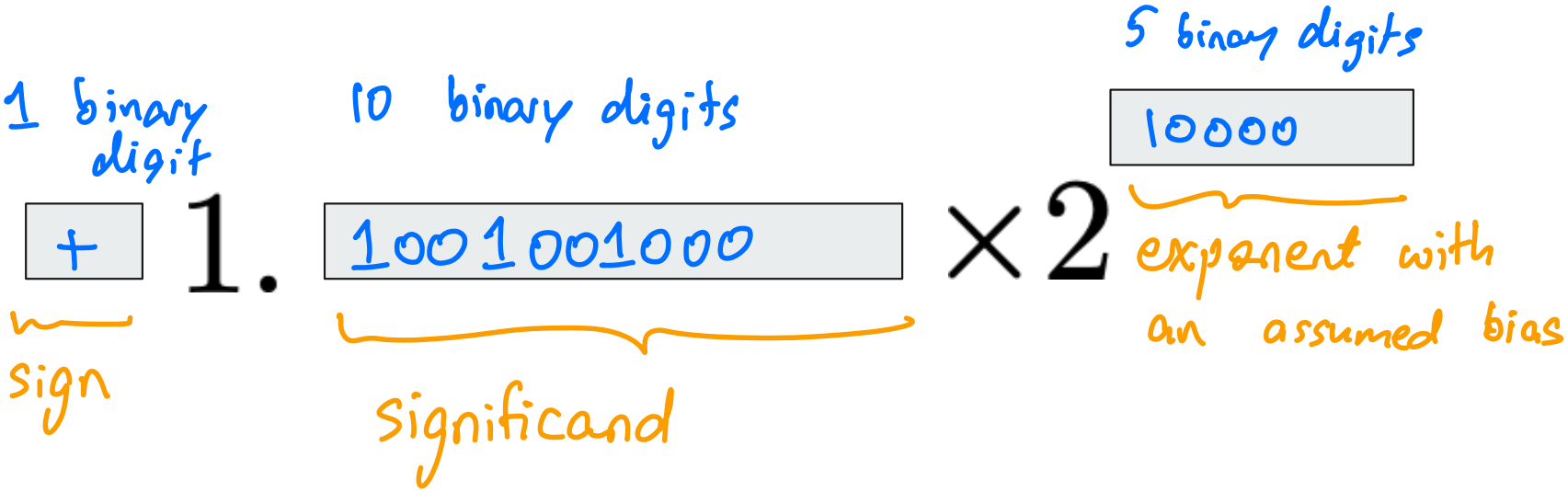
→ sign

→ 3 digits for number.

→ 1 digit for exponent

Assumes the $\times 10^{\square}$ structure.

The IEEE Standard for Floating-Point Binary Numbers



Exponent: 0 to 31 if we don't use a bias
-14 to 15 using a bias.

Bias: subtract from exponent to interpret it.
for 16-bit numbers, 15 is the standard bias

https://evanw.github.io/float-toy/

16-bit, 32-bit and 64-bit floats

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