



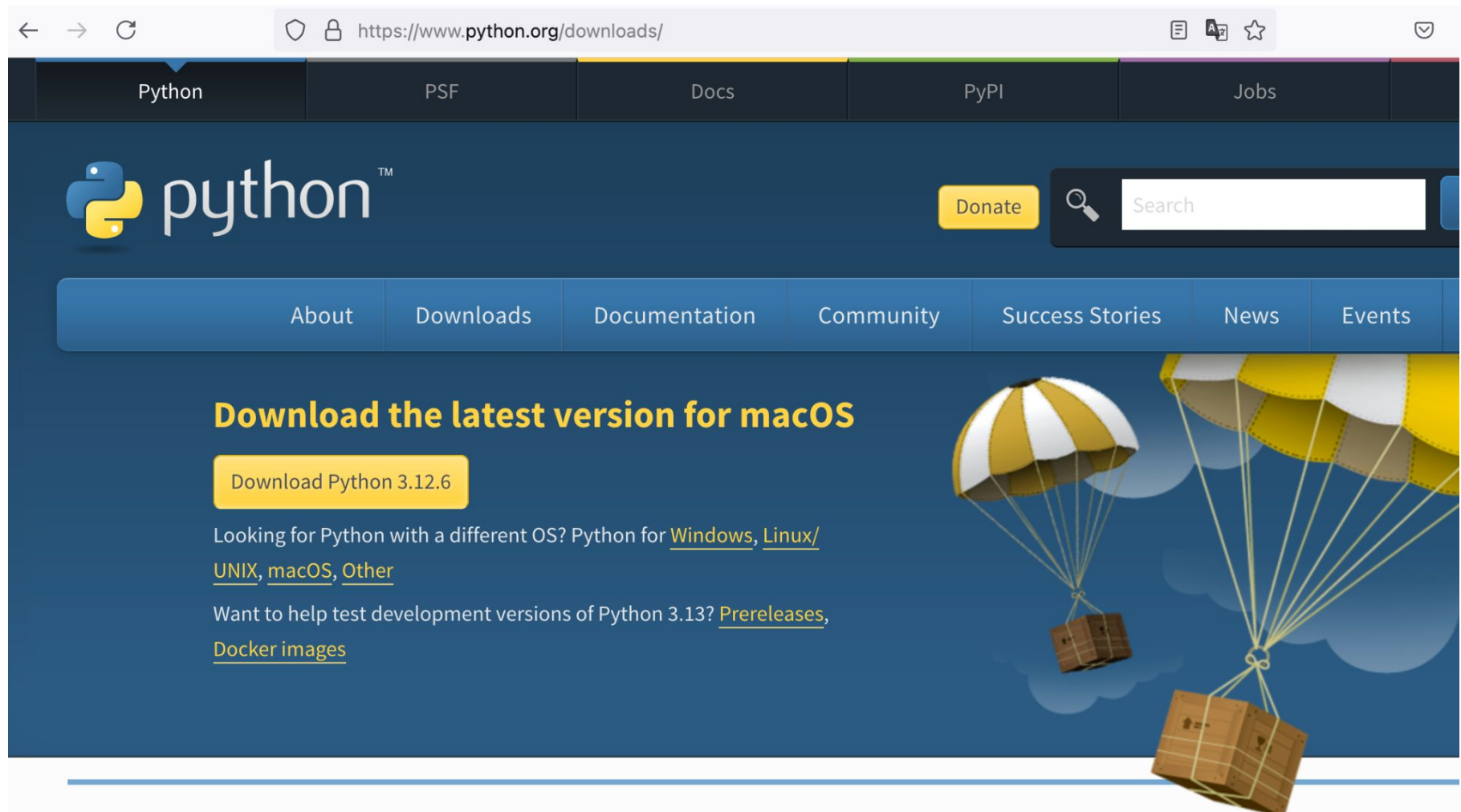
ENGR 21:

Computer Engineering Fundamentals

Lecture 7
Tuesday, September 23, 2025

Review of Python Installation

Installing Python on your computer





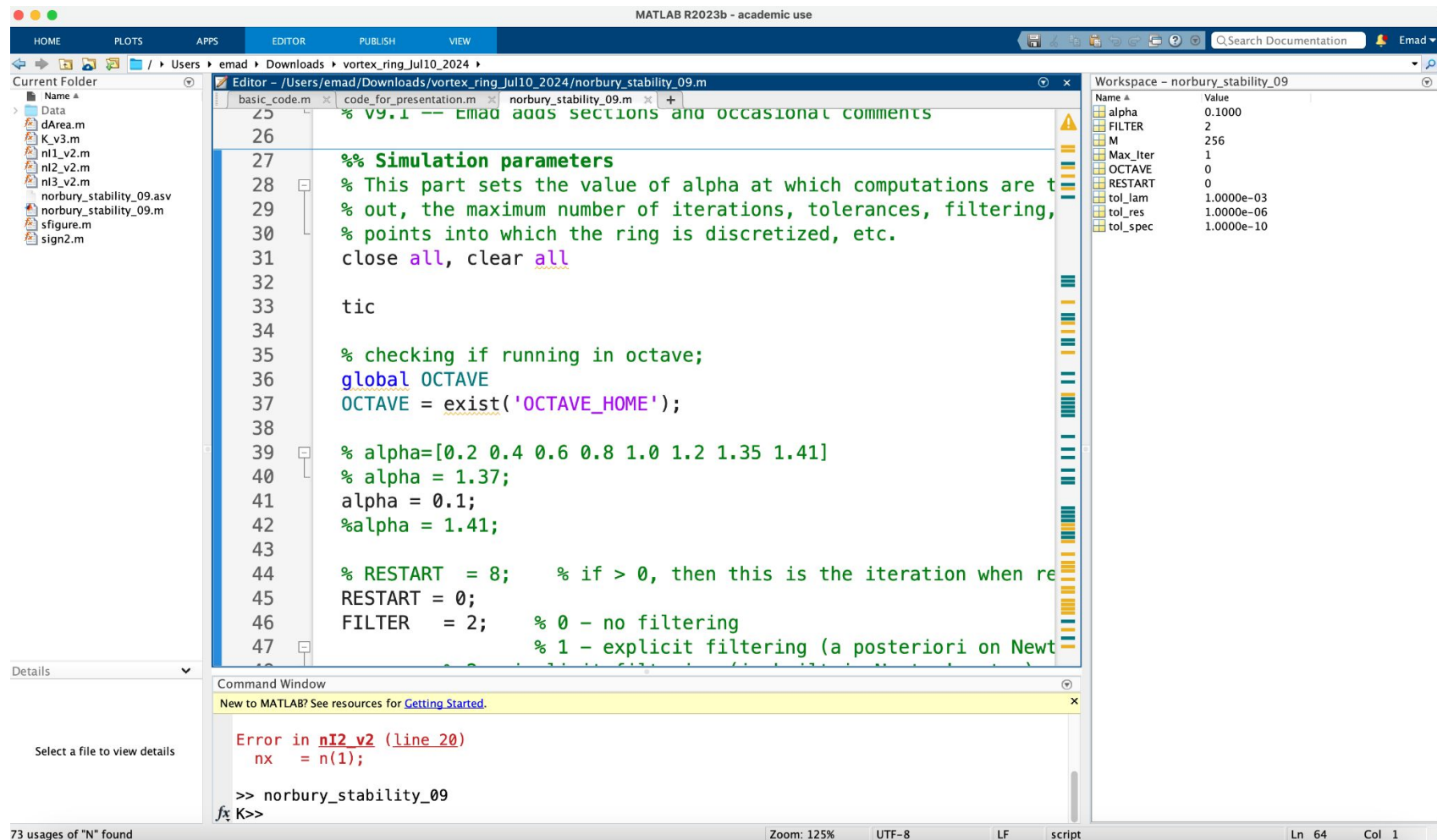
Choice of IDE (Integrated Development Environment)

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

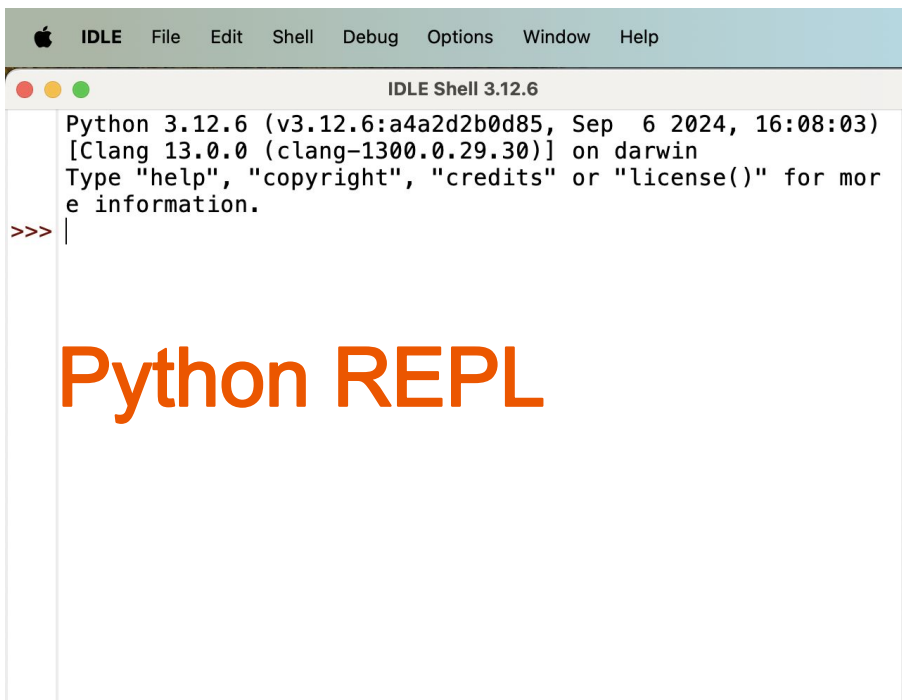
MATLAB has a built-in IDE



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

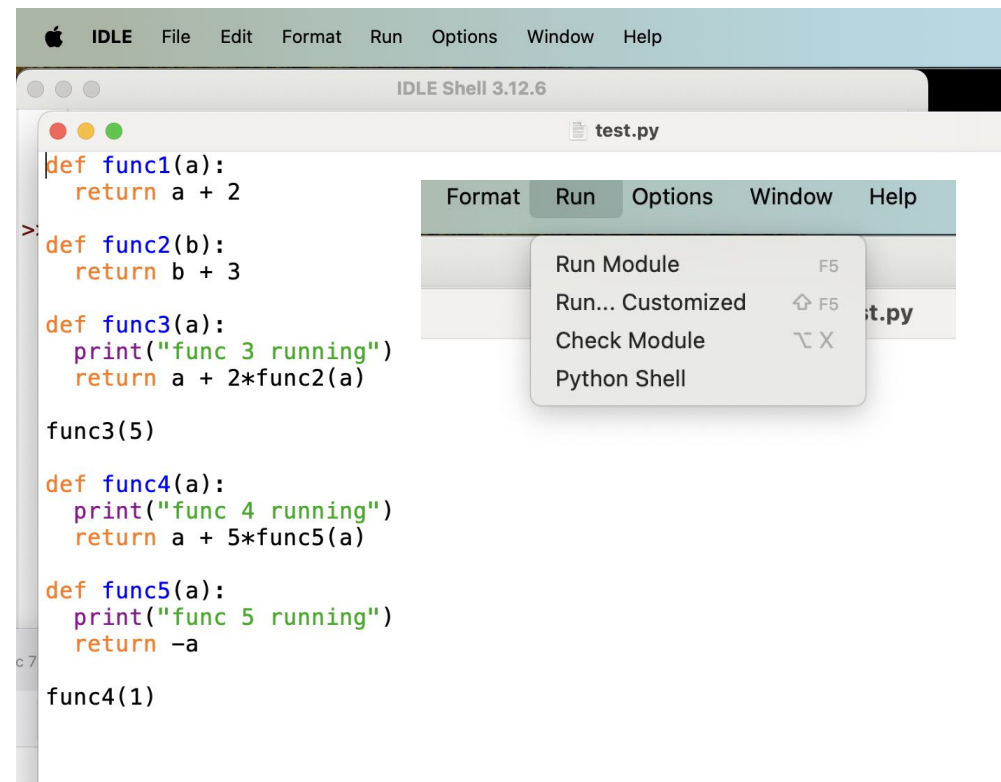
Comes pre-installed with Python if you get it from www.python.org/downloads

Search for 'IDLE' in start menu or Launchpad



```
Python 3.12.6 (v3.12.6:a4a2d2b0d85, Sep 6 2024, 16:08:03)
[Clang 13.0.0 (clang-1300.0.29.30)] on darwin
Type "help", "copyright", "credits" or "license()" for more
>>> |
```

Python REPL



```
def func1(a):
    return a + 2

def func2(b):
    return b + 3

def func3(a):
    print("func 3 running")
    return a + 2*func2(a)

func3(5)

def func4(a):
    print("func 4 running")
    return a + 5*func5(a)

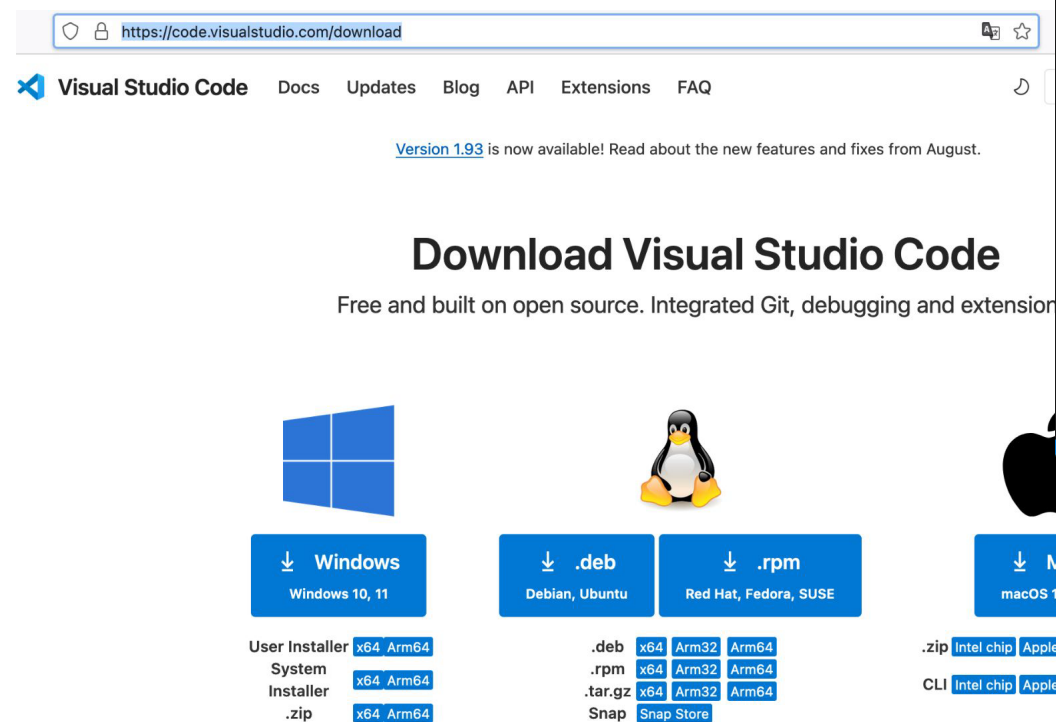
def func5(a):
    print("func 5 running")
    return -a

func4(1)
```

Run Module F5
Run... Customized ⇧ F5
Check Module ⌘ X
Python Shell

Visual Studio Code (VS Code)

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



The screenshot shows the Visual Studio Code download page. At the top, the URL <https://code.visualstudio.com/download> is in the browser address bar. The page features the Visual Studio Code logo and navigation links like Docs, Updates, Blog, API, Extensions, and FAQ. A message states "Version 1.93 is now available! Read about the new features and fixes from August." Below this is the heading "Download Visual Studio Code" with the tagline "Free and built on open source. Integrated Git, debugging and extension". The page is divided into three main sections for Windows, Linux, and Mac. The Windows section offers a download for Windows 10, 11. The Linux section provides download links for .deb (Debian, Ubuntu) and .rpm (Red Hat, Fedora, SUSE), along with a table of system installers and Snap Store options for various architectures. The Mac section offers a download for macOS 10.15+.

<https://code.visualstudio.com/download>

Visual Studio Code Docs Updates Blog API Extensions FAQ

Version 1.93 is now available! Read about the new features and fixes from August.

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extension

Windows
Windows 10, 11

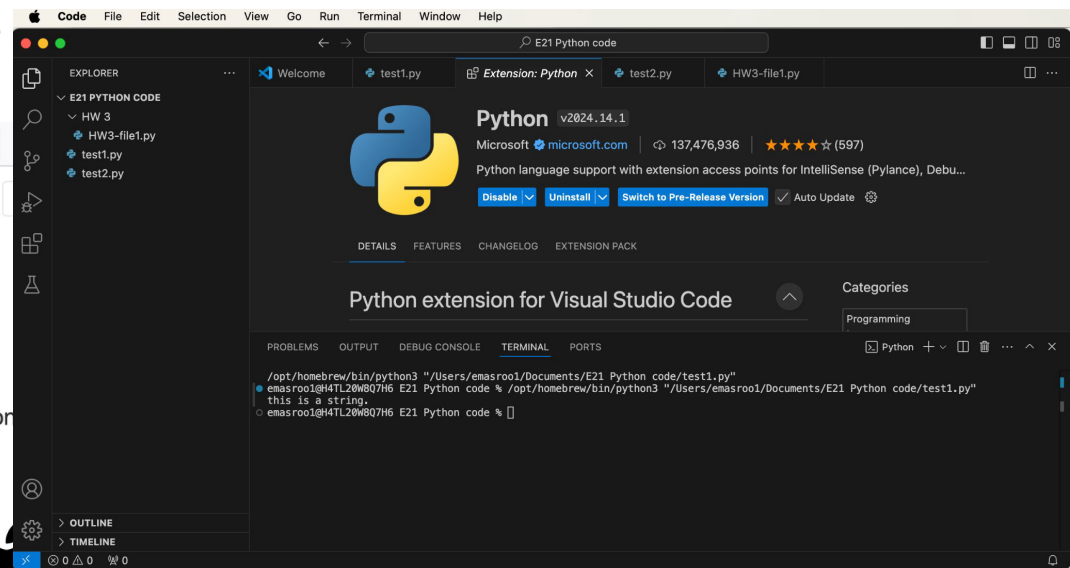
Linux

- .deb**
Debian, Ubuntu
- .rpm**
Red Hat, Fedora, SUSE

System Installer	x64	Arm32	Arm64
User Installer	x64	Arm32	Arm64
System Installer	x64	Arm32	Arm64
.zip	x64	Arm32	Arm64

Mac
macOS 10.15+

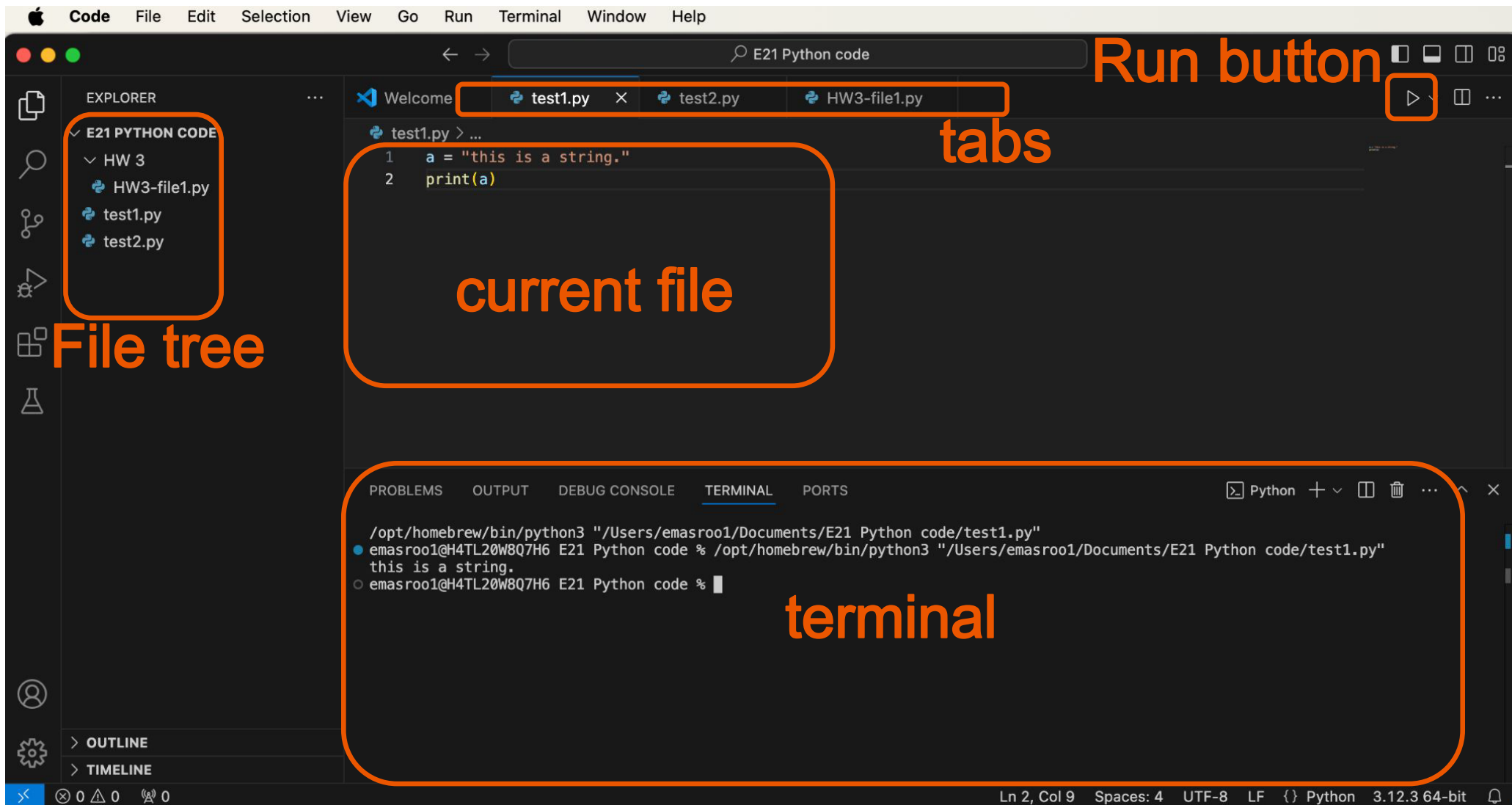
CLI
Intel chip Apple silicon Universal



<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”

Select Interpreter

Selected Interpreter: /opt/homebrew/bin/python3

Selected Interpreter: /opt/homebrew/bin/python3

Enter interpreter path...

Python 3.12.3 64-bit /opt/homebrew/bin/python3

Python 3.11.4 64-bit /usr/local/bin/python3

Python 3.9.6 64-bit /usr/bin/python3

Recommended

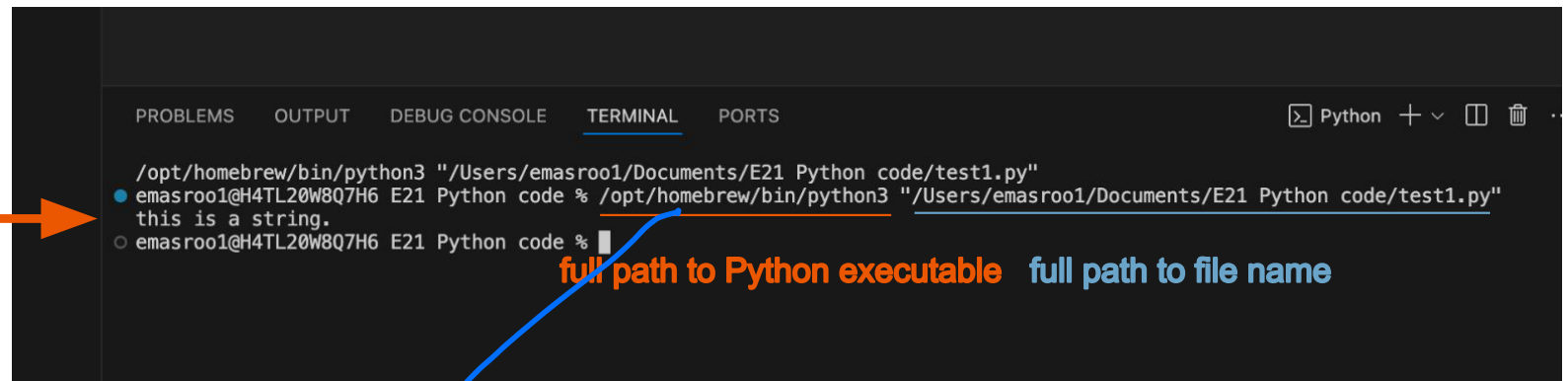
Global

VS Code found 3 instances of Python

Coding 'on the command line'

A closer look at “the terminal”

output prints to terminal →



The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
/opt/homebrew/bin/python3 "/Users/emasroo1/Documents/E21 Python code/test1.py"
• emasroo1@H4TL20W8Q7H6 E21 Python code % /opt/homebrew/bin/python3 "/Users/emasroo1/Documents/E21 Python code/test1.py"
  this is a string.
○ emasroo1@H4TL20W8Q7H6 E21 Python code %
```

full path to Python executable full path to file name

Command argument

full path to python
shortcut or 'python3'

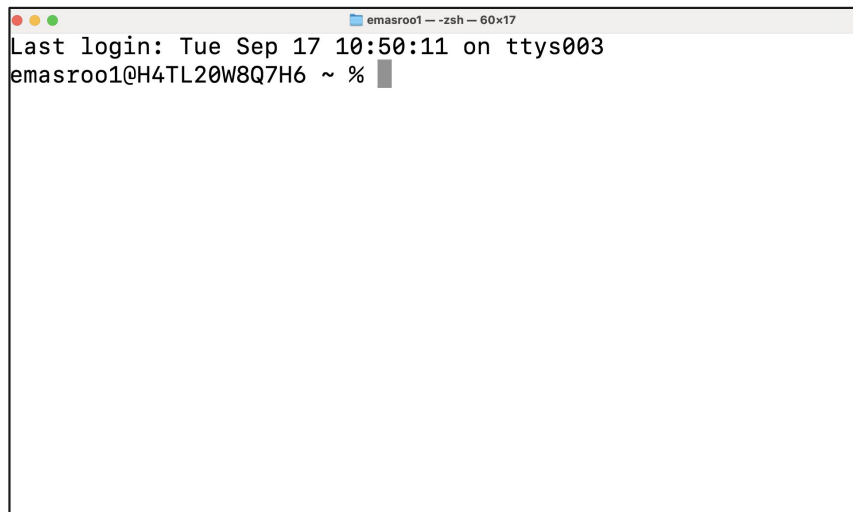
option: use "full path" or shortcut

some file containing python code

Terminals & Command-Line Interfaces today

Mac

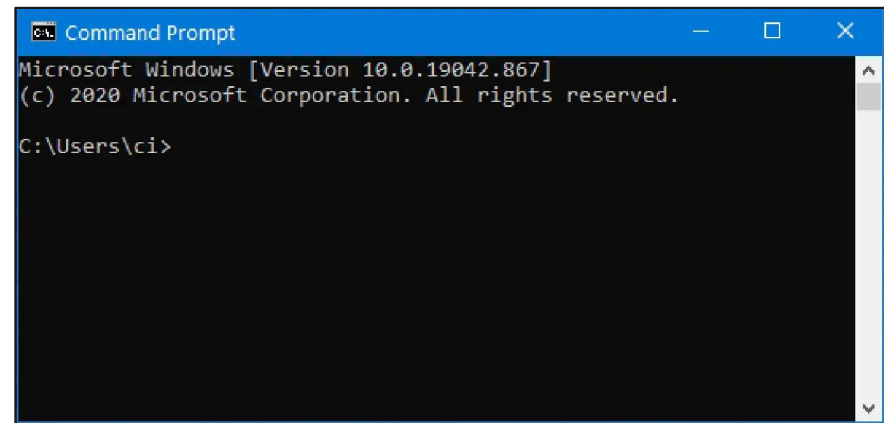
Terminal

A screenshot of a Mac Terminal window. The title bar shows 'emasroo1 - zsh - 60x17'. The terminal text reads: 'Last login: Tue Sep 17 10:50:11 on ttys003' followed by 'emasroo1@H4TL20W8Q7H6 ~ %' with a cursor.

```
emasroo1 - zsh - 60x17
Last login: Tue Sep 17 10:50:11 on ttys003
emasroo1@H4TL20W8Q7H6 ~ %
```

Windows

Command Prompt, aka `cmd.exe`

A screenshot of a Windows Command Prompt window. The title bar shows 'Command Prompt'. The text inside reads: 'Microsoft Windows [Version 10.0.19042.867] (c) 2020 Microsoft Corporation. All rights reserved. C:\Users\ci>'.

```
Command Prompt
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.
C:\Users\ci>
```

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
```

or

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

gives full path

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
```

1 CL argument, 'hello'

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

1 CL argument, 'hello 1'

```
emasroo1@iMac Documents % python test1.py 40 num
```

2 CL arguments, '40' and 'num'

- How do you access command-line arguments inside the program? *test1.py?*
- `sys.argv` is a list containing the CL arguments as strings
- `sys.argv[0]` is the file name

→ argument vector

```
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.

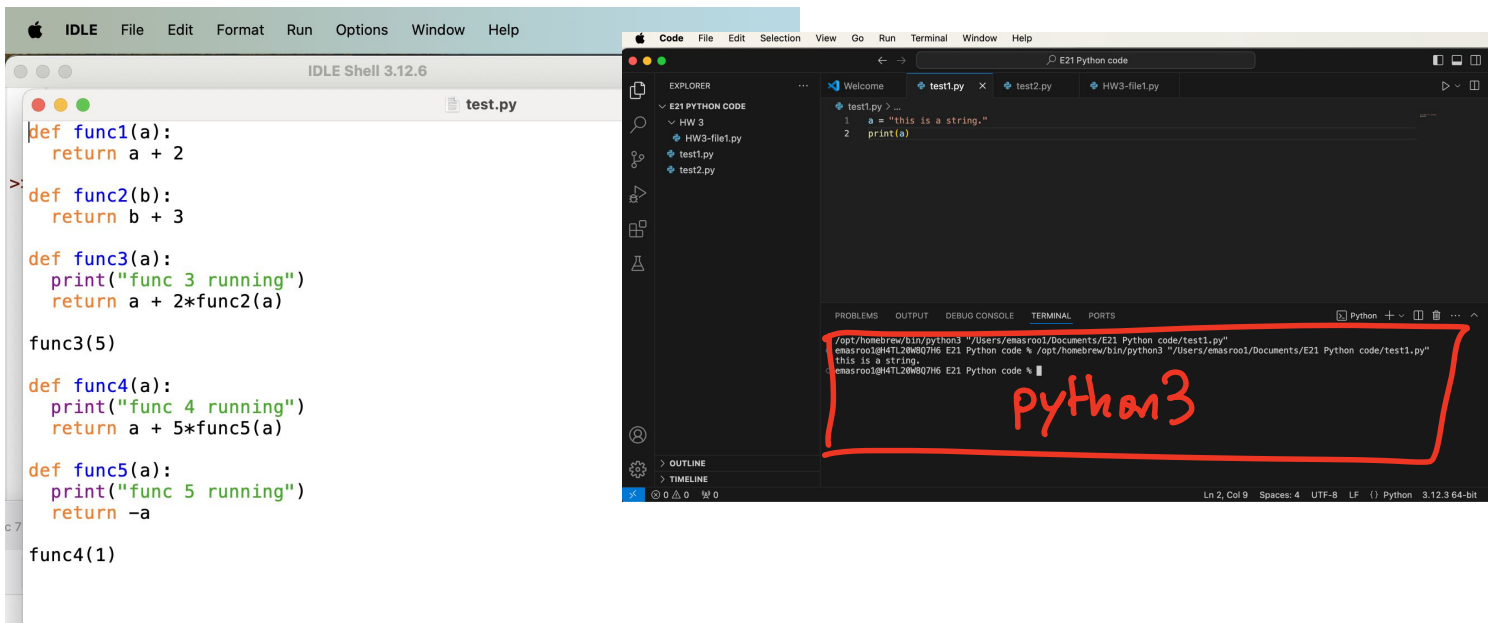
interactive

REPL window

Open REPL from within VS code

Type 'python' or 'python3'

*.py extension
is only a
convention*

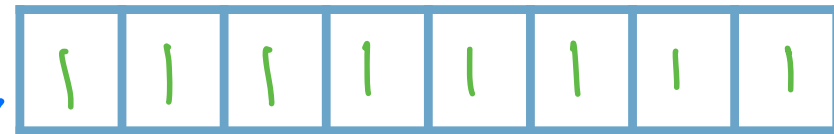


Floating-point system

What range of natural numbers can a computer store?

- With 8-bit binary numbers

$$999 : 10^3 - 1$$



biggest positive number available :

$$2^8 - 1$$

The largest possible 8-bit binary integer:



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

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

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 ?

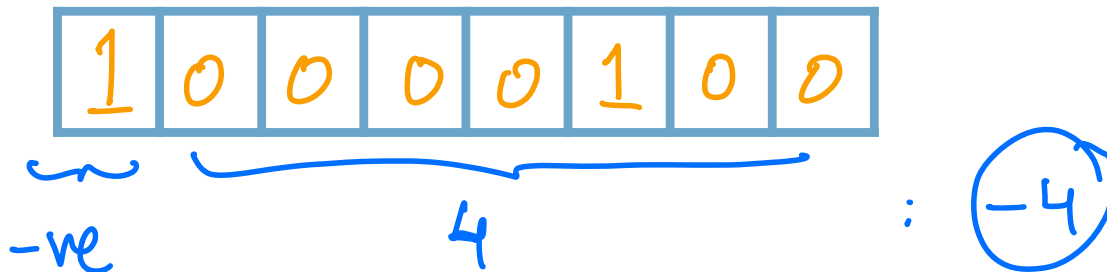
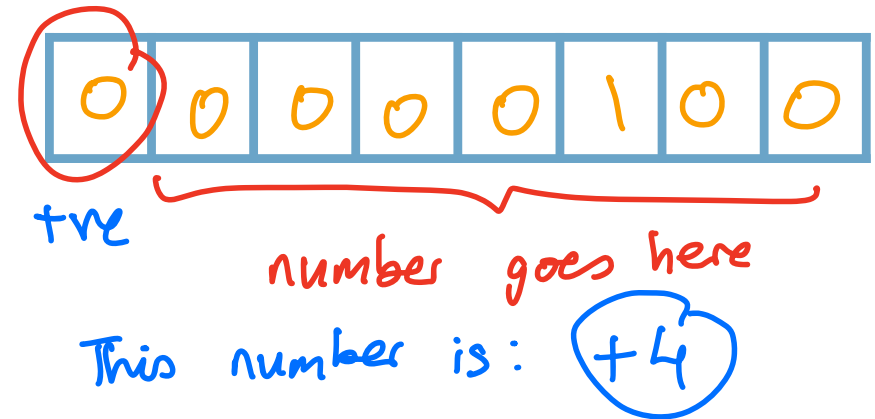
256 options.

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

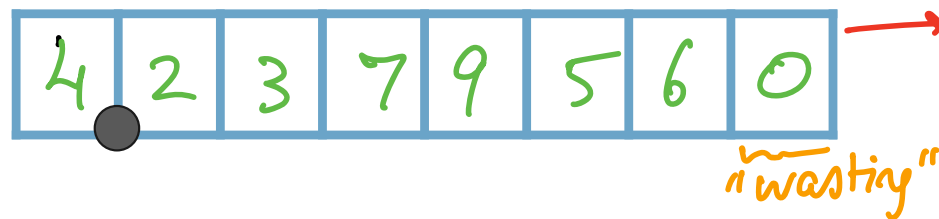
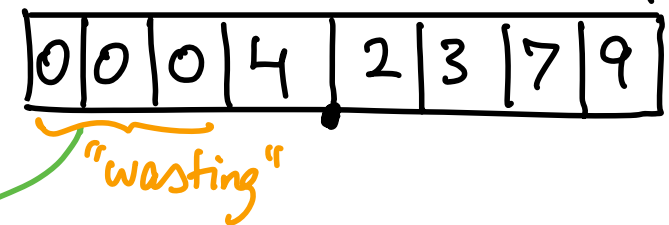
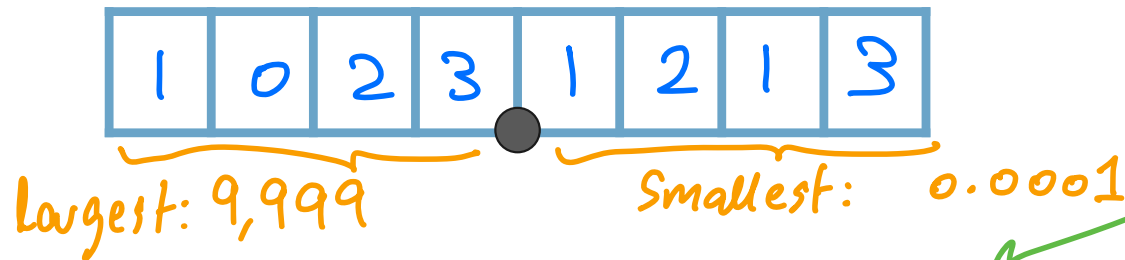


Different ways of storing "real" numbers: **Fixed-Point** *"decimal point"*

- 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.

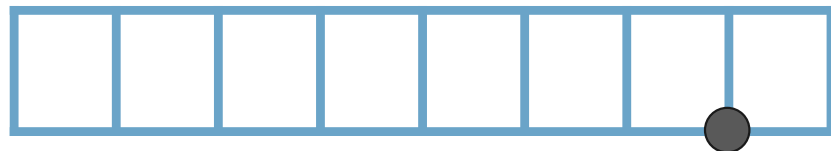
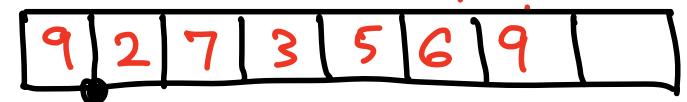
8 decimal digits available

store 4.237956 *best you can do.*



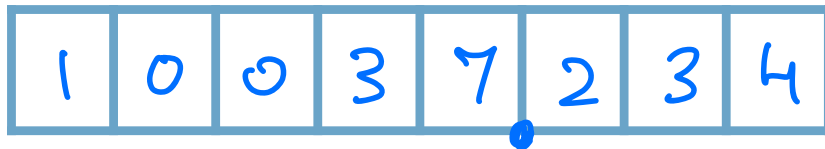
instead, store

34.273569 ~> ?



The **Floating-Point** idea (applied to decimal numbers)

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



→ currently, decimal point is at location 4

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

“Scientific notation” for numbers

Information conveyed by this:
→ Sign

→ 3 digits for number.

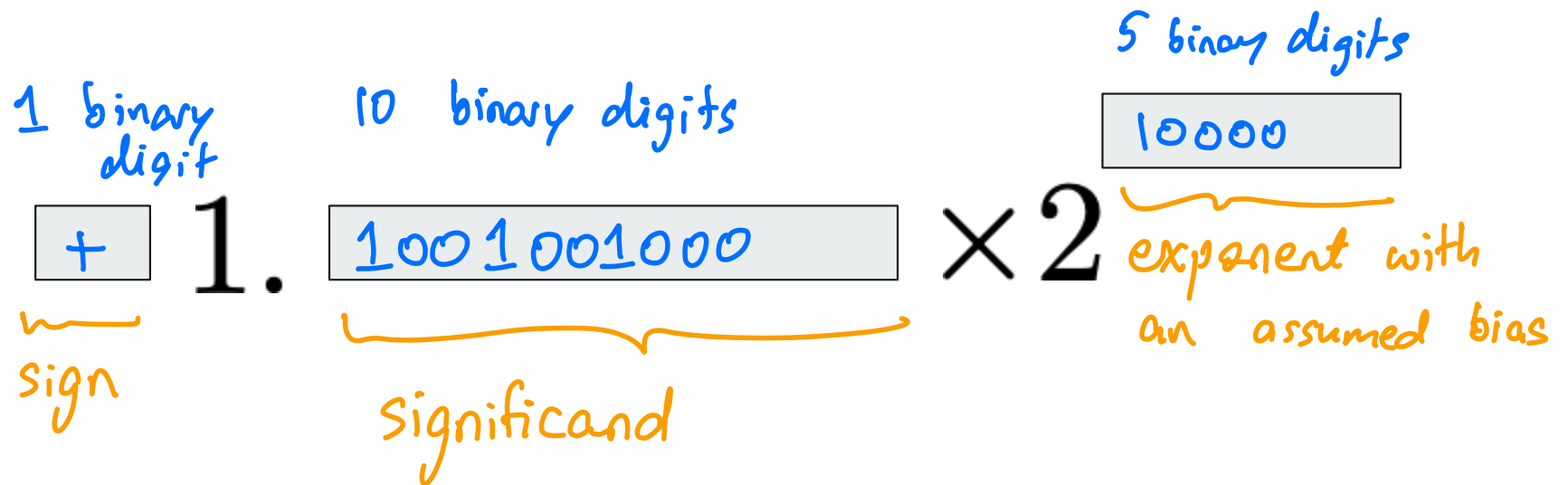
→ 1 digit for exponent

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

$$\left[-2.34 \times 10^5 \right]$$

from 0 to 9 }
-5 to 4 } or
choose by convention

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 14 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

instead of 1.234×10^{56}

sign bit. \leftarrow $1.$ \leftarrow some binary number. $\times 2$

assumed to be 1. \leftarrow some binary number

\square 1. \square $\times 2$ \square