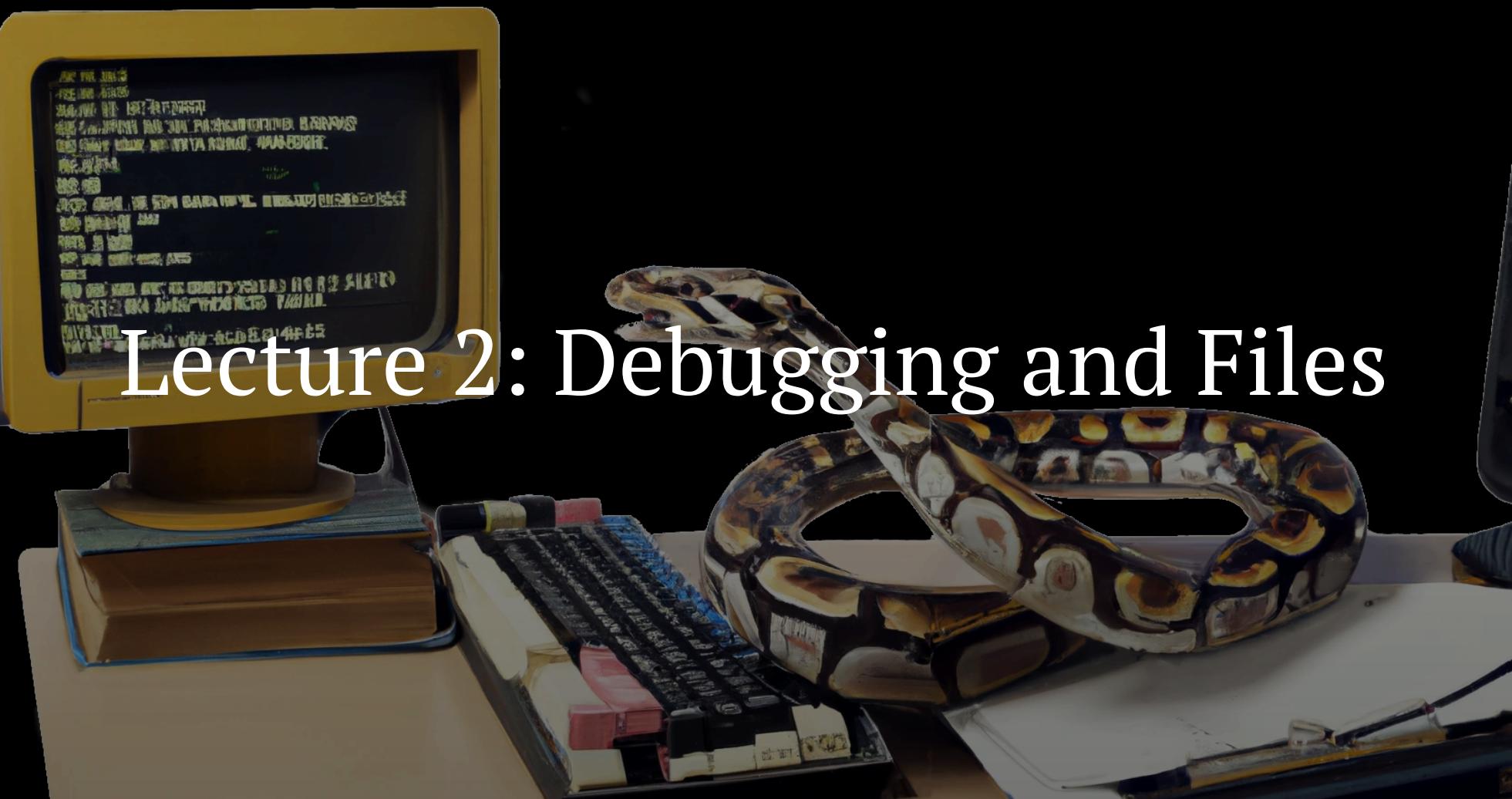


Lecture 2: Debugging and Files





Roadmap

- Homework assignment
- How to level up as a programmer
- Debugging
- File I/O
- Homework Assignment



Homework Assignment: Hangman

```
✓ TERMINAL zsh - code + □
● (.venv) $ /Users/sjoner/src/pyinter-2025/.venv/bin/python
  /Users/sjoner/src/pyinter-2025/code/hangman.py
Let's play hangman!

-----
Incorrect guesses:
You have 6 guesses left.
Guess a letter: e
Good guess! e is in the word.

_ _ _ e

Incorrect guesses:
You have 6 guesses left.
Guess a letter: s
Sorry, s is not in the word.

_ _ _ e

Incorrect guesses: s
You have 5 guesses left.
Guess a letter: c
Good guess! c is in the word.

c _ _ e

Incorrect guesses: s
You have 5 guesses left.
Guess a letter: o
Good guess! o is in the word.

c o _ e

Incorrect guesses: s
You have 5 guesses left.
Guess a letter: d
Good guess! d is in the word.
Congratulations! You guessed the word: code
diamond (.venv) $
```

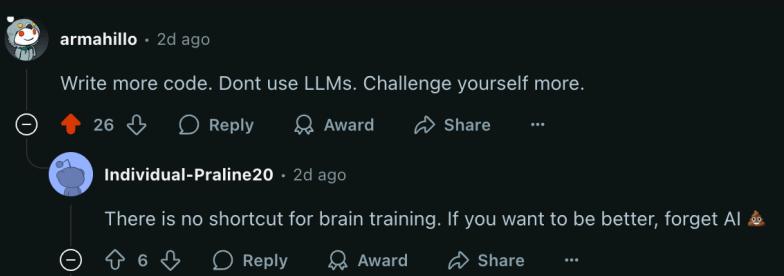
How to Level Up as a Programmer (1/4)

- How to go from ok programmer to good programmer?



How to Level Up as a Programmer (2/4)

- How to go from ok programmer to good programmer?



How to Level Up as a Programmer (3/4)

- How to go from ok programmer to good programmer?



LateAeon · 2d ago

Honestly my biggest periods of growth come from when I feel the most stuck on something for the longest time. If I keep at it persisting through, I will eventually breakthrough and grow tremendously in my problem solving abilities and skills as a programmer.

Like many others here have said, just keep at it. Keep building different things, build whatever interests you for fun on the side. Not all projects I build go great. Some flop, some never finish. But in the end the longer I program the better my dev setup, workflow, language familiarity all get... and I become a better programmer.



2



Reply



Award



Share

...

How to Level Up as a Programmer (4/4)

- How to go from ok programmer to good programmer?



motific · 2d ago

Understanding software engineering vs just bashing out code. Get good at debugging other people's code. Understand secure development practices.

Write the hard code - If you write for Linux then make your code run on OpenBSD and FreeBSD too, cross platform development helps you find bugs you didn't know you had.



4



Reply



Award



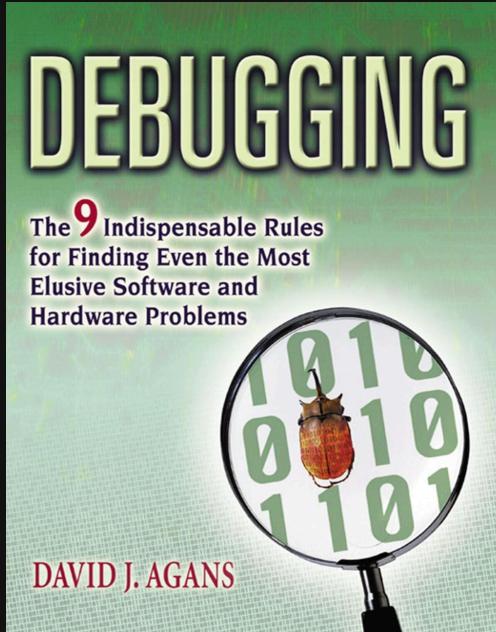
Share

...



Debugging

1. Understand the System
2. Make it fail
3. Quit thinking and look
4. Divide and conquer
5. Change one thing at a time
6. Keep an audit trail
7. Check the plug
8. Get a fresh view
9. If you didn't fix it, it ain't fixed



Debugging by David J. Agans

 Back to Hangman...

- What about more words?
- Let's read them from a file

Reading Files

```
built-in function  
f = open("file.txt", "r", encoding="utf-8")  
file object    file name    mode      encoding
```

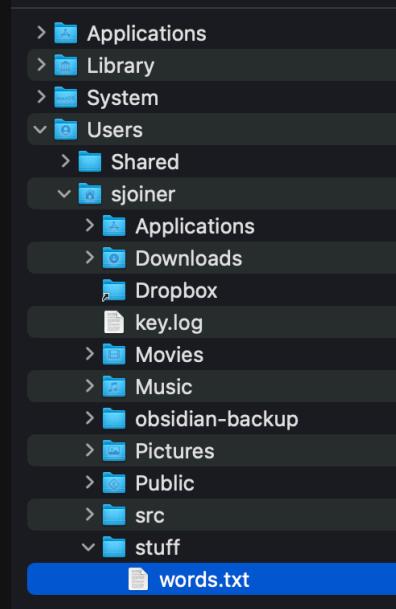
Example

```
file = open("file.txt", "r", encoding="utf-8")  
# do stuff with file...  
file.close()
```

- File name:
 - If no path, then file is in same directory from which you ran the program
 - Path can specify other location
- Mode:
 - "r" : open for reading
 - "w" : open for writing (erases existing content)
 - "x" : open for exclusive creation (fails if file already exists)
 - "a" : open for appending
 - "b" : binary mode
 - "t" : text mode
 - "+" : open for updating (reading and writing)

File Paths

- Files are organized on your hard drive in *directories* (also called folders)
- Separated by / on Mac and Linux, \ on Windows
- Mac and Linux example:
 - `/home/Users/steve/stuff/words.txt`
- Windows Example:
 - `C:\Users\Steve\stuff\words.txt`
 - Note: Windows also includes drive letter
- These are *absolute* paths
 - Specify exact location starting from system root
- Relative paths specify location relative to your current directory
 - .. means go up one directory
 - Example: `../../other-files/words.txt`





os.path Module

- Lets us deal with paths, files, and directories in an OS-agnostic way

```
from os import path

# results in:
#     ../stuff/words.txt on Mac/Linux
#     ..\stuff\words.txt on Windows

mypath = path.join('..', 'stuff', 'words.txt')

path.dirname(mypath) # returns ../stuff
path.exists(mypath) # returns True if the file exists

path.isdir(mypath) # returns True if path is a directory
path.isfile(mypath) # returns True if path is a file
```

- See documentation for `os.path` for more useful functions



with Statement

```
from os import path

words_path = path.join('stuff', 'words.txt')

with open(words_path, 'r', encoding="utf-8") as file:
    # do stuff with file
```

```
from os import path

words_path = path.join('stuff', 'words.txt')

file = open(words_path, 'r', encoding="utf-8")
# do stuff with file
file.close()
```

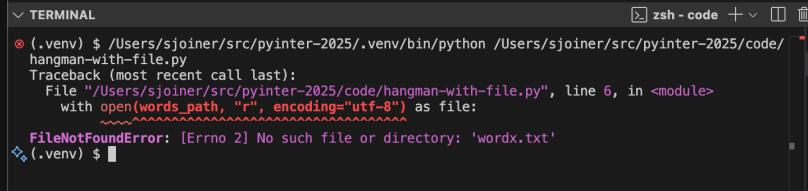
- Alternative to `open` and `close`
- Automatically opens and closes the file for us
- Prevents errors from forgetting to close the file
- Expresses intent

File Not Found Error

```
from os import path

words_path = path.join('stuff', 'words.txt')

with open(words_path, 'r', encoding="utf-8") as file:
    # do stuff with file
```



A screenshot of a terminal window titled "TERMINAL". The window shows a command being run: `./venv $ /Users/sjoiner/src/pyinter-2025/.venv/bin/python /Users/sjoiner/src/pyinter-2025/code/hangman-with-file.py`. The output shows a `FileNotFoundError`: [Errno 2] No such file or directory: 'wordx.txt'. The terminal window has a dark background with light-colored text.

- You'll see this if:
 - Your filename doesn't match
 - Your path is wrong
 - Your file isn't in the directory
 - Your code is in a different directory



Shell Commands

- List files and directories
 - Linux / Mac: `ls`
 - Windows: `dir`
- Print working directory
 - Linux / Mac: `pwd`
 - Windows: `cd` (no arguments)
- Change directory
 - Linux / Mac: `cd foo`
 - Windows: `cd foo`

```
▽ TERMINAL
● (.venv) $ pwd
/Users/sjjoiner/src/pyinter-2025
● (.venv) $ ls
assignments      layouts      node_modules    README.md      styles
bun.lock        lecture-01.md  package.json   setup
code            lecture-01.pdf  pages          snippets
components      lecture-02.md  public         sources
● (.venv) $ cd code
● (.venv) $ pwd
/Users/sjjoiner/src/pyinter-2025/code
● (.venv) $ ls
american-words.35.txt  hangman-with-file.py  selfref.py
dobracketsmatch.py     isogram-set.py       words.txt
example.py             isogram.py
● (.venv) $ cd ..
● (.venv) $ pwd
/Users/sjjoiner/src/pyinter-2025
○ (.venv) $ █
```



Reading from the File

- `file.read()`
 - read the entire file as a string
- `file.readline()`
 - read the next line as a string
- `file.readlines()`
 - read the file as a list of strings
- `for line in file:`
 - iterate over each line in the file
- line contains newline character(s) `\n`, or `\r\n`
- remove them with `string.strip()`

```
from os import path

words_path = path.join("stuff", "words.txt")
words = []

with open(words_path, "r", encoding="utf-8") as file:
    for line in file:
        sline = line.strip()
        if len(sline) >= 4 and len(sline) <= 7:
            words.append(sline)
```

- Now you can read words from a file for your hangman program
 - 10,000 most common english words
 - Scrabble word list
 - Large english words list



Improve Your Program

- Get words from file
- Handle invalid input
- Handle guessing a letter that was already guessed



Homework 2: Evil Hangman

Evil Hangman is a computer program that cheats at the classic game of Hangman. Normally, the picks a single word and accurately represents it as the human player tries to guess all of the letters. In Evil Hangman, the computer instead maintains a large list words, then continuously pares down the word list to try to dodge the player's guesses.

- Start with your improved hangman program
- Read words from file
- Determine number of letters in starting words
- Pare down list of words by word length
- When the player makes a guess, pare down word list by removing words containing guessed letter
 - If there are words remaining after paring down, then tell the player their guess was incorrect and update your word list
 - If there aren't any words remaining after paring down, then choose a word at random from the list before paring down. Use this word to continue playing hangman normally