



# BIOE50010 – Programming 2

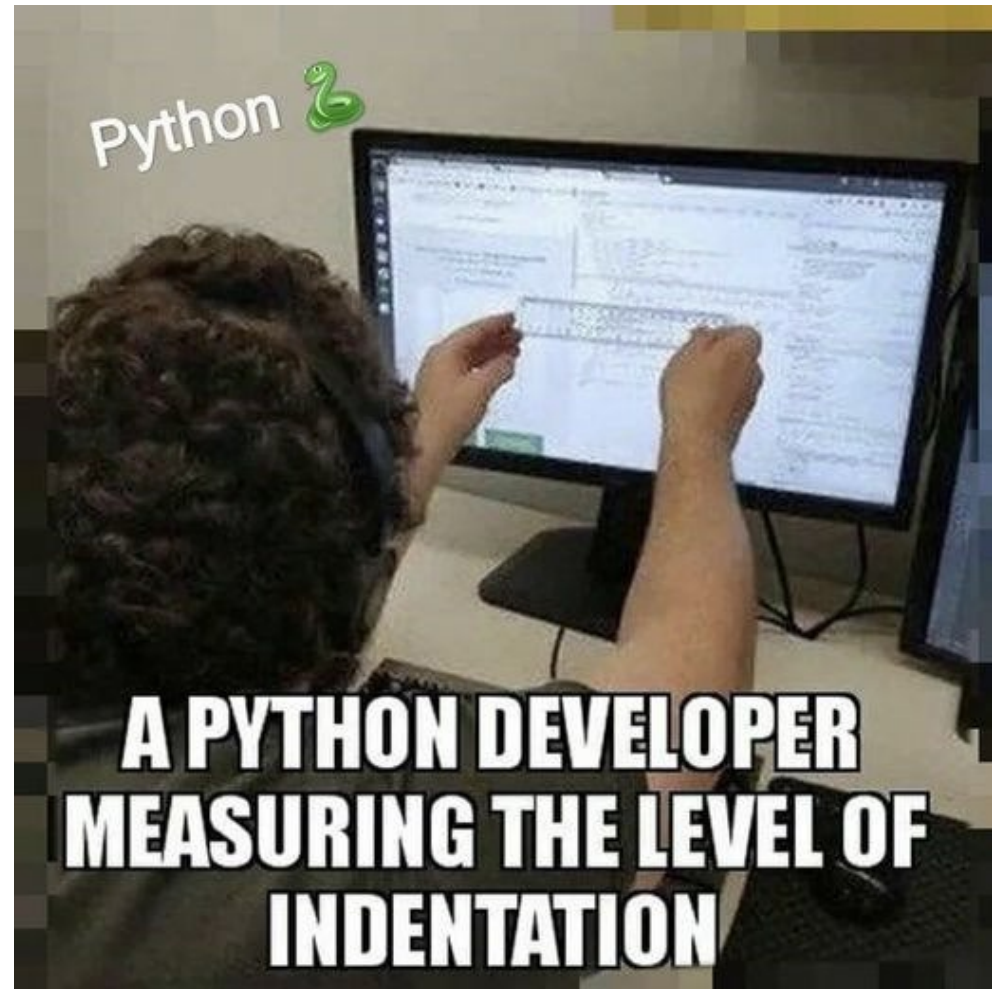
## *Computer Lab 2*

*Binghuan Li* | Department of Chemical Engineering

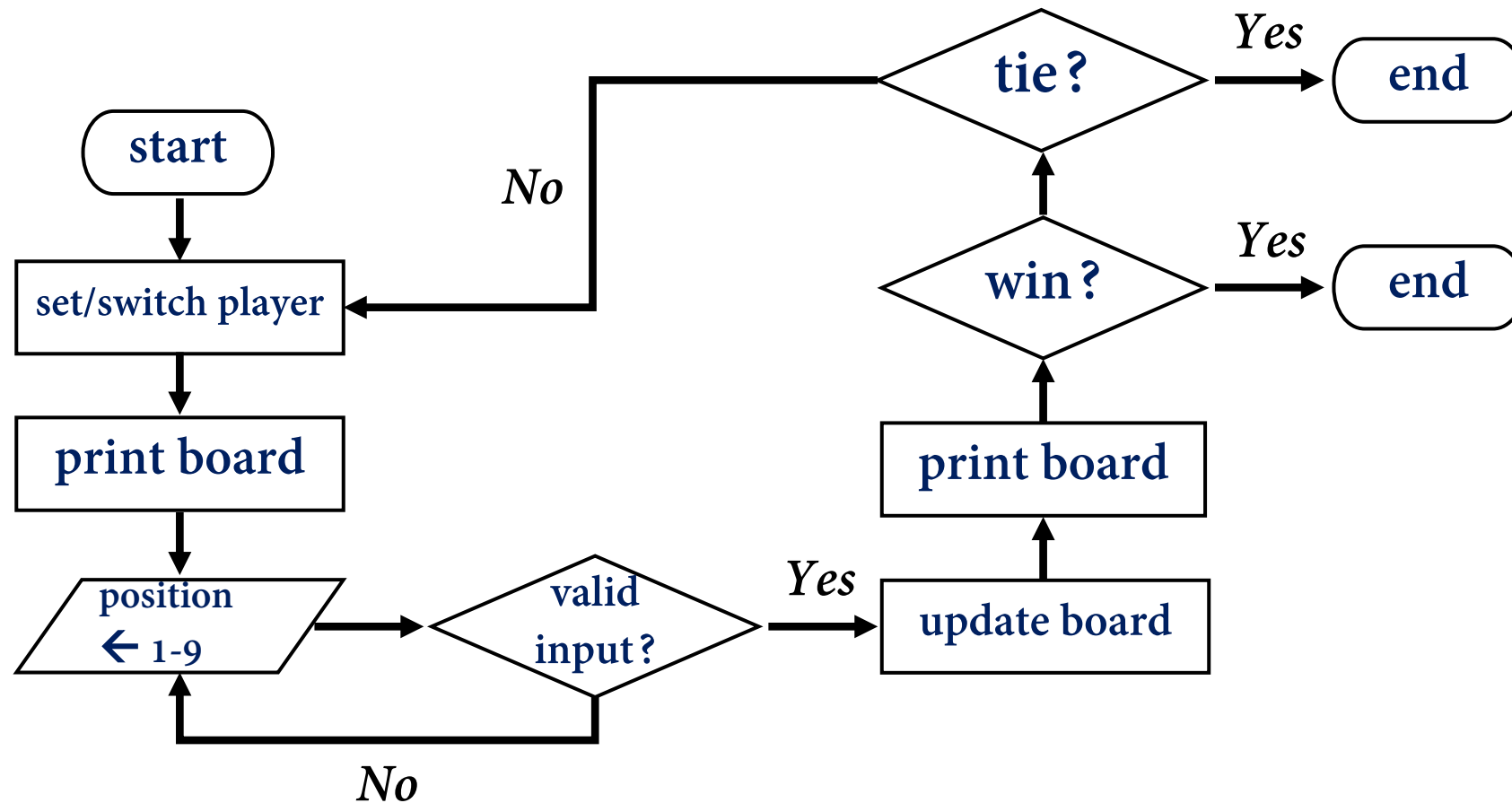
[\*binghuan.li19@imperial.ac.uk\*](mailto:binghuan.li19@imperial.ac.uk)

**October 16, 2023**

# Meme of the week 😊



# Last Week – Tic Tac Toe



## Your self-checklist

- *Data structures*
- *Typecasting*
- *Control flow: conditions and loops*
- *Input*
- *Functions, scopes*

# displayBoard

Code snippet from `tictactoe_sol1.py`

```
board = [" ", " ", " ", " ", " ", " ", " ", " ", " "]

def displayBoard(board):
    print(board[0] + '|' + board[1] + '|' + board[2])
    print('-----')
    print(board[3] + '|' + board[4] + '|' + board[5])
    print('-----')
    print(board[6] + '|' + board[7] + '|' + board[8])
```

Initialize a 1-D list with 9  
elements (spaces), also  
`[" "]*9`

This is just formatting!

- Format your output, rather than directly printing out a 2-D list...

```
board = [[1, 2, 3],
         [4, 5, 6],
         [7, 8, 9]]
print(board)
```



This won't format  
your board...  
unfortunately

# Namespace

## Example

```
myNumber = 1 ← Global space

def myFunc1():
    myNumber = 2 ← Local space
    print(myNumber)

def myFunc2():
    myNumber = 3 ← Local space
    print(myNumber)

def main():
    myFunc1()
    myFunc2()
    print(myNumber)
```

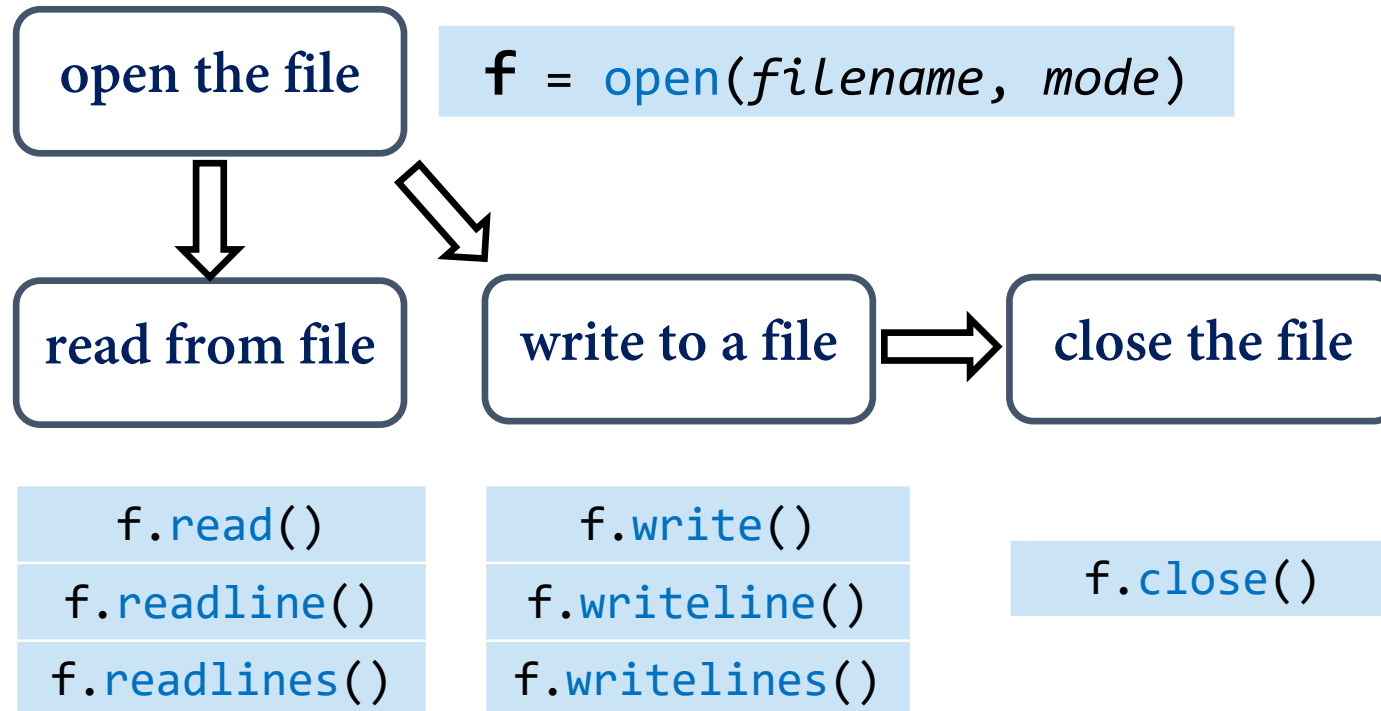
- Q: what will the output be like when I trigger `main()`?

## Console

```
>> 2
3
1
```

*Shout your questions from Lab 1!*

# File I/O



- `f=open(...)` returns a file object:
  - You work with the object: `f.read()`, `f.write()`
- Modes in `open`
  - 'r' read (default)
  - 'w' write
  - 'a' append
  - 'x' create
  - 'b' binary
- Always close your file once finished all operations!

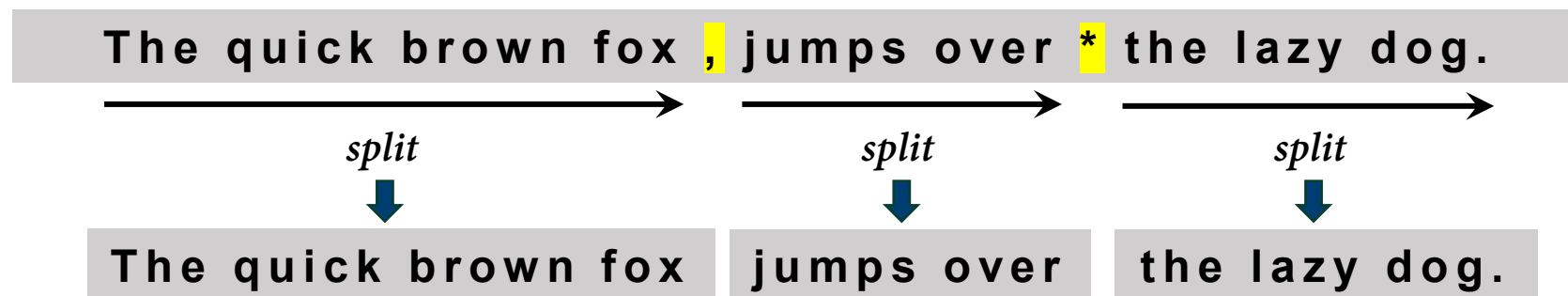
# Why do I need to close the file?

- more space used in the RAM, hence impact the performance
- many changes to files in python do not go into effect until *after* the file is closed
- theoretically, run in to limits of how many files you can have open
- likelihood for data corruption



# Your tasks today

- 3 mini tasks on **file I/O, formatting, and use of Python data structures**
  - Task 1: read a poem from a `.txt` file
  - Task 2: read and format the DNA to protein data from a `.csv` file
  - Task 3: read and process nucleotide sequences
- Read the sample console output carefully
- Hint: split a string with *delimiters*: consider using `split()` function



*Questions?*

*That's it for now.*

*You can now proceed to the Lab 2 exercise.*