

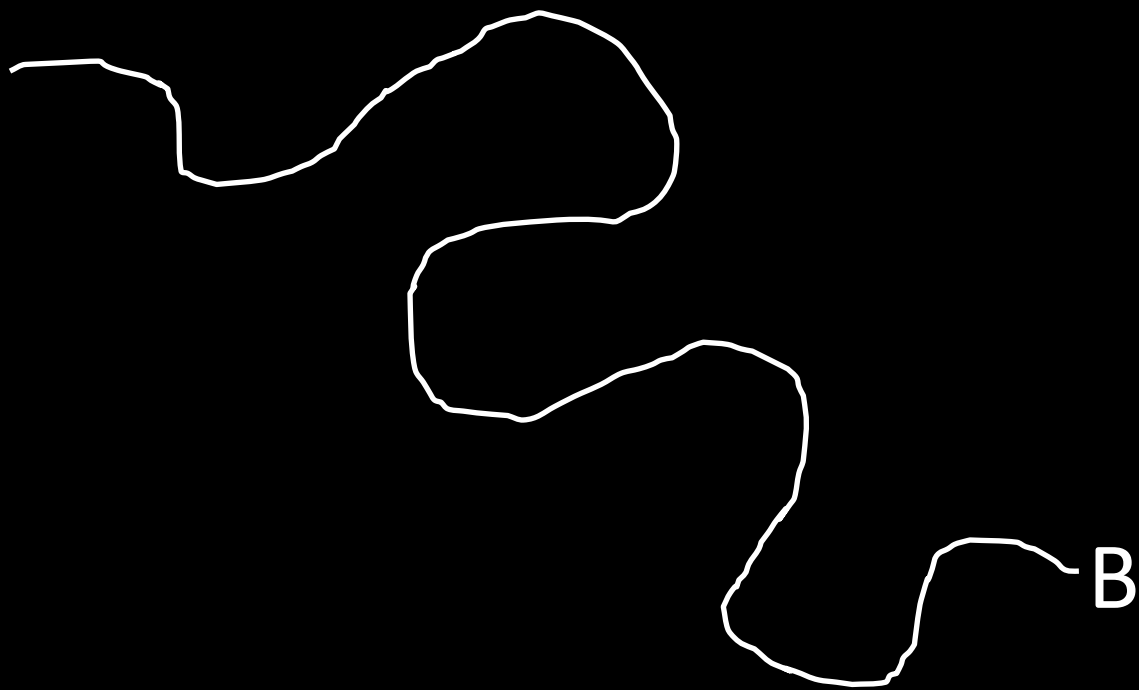
Python in
style!



A

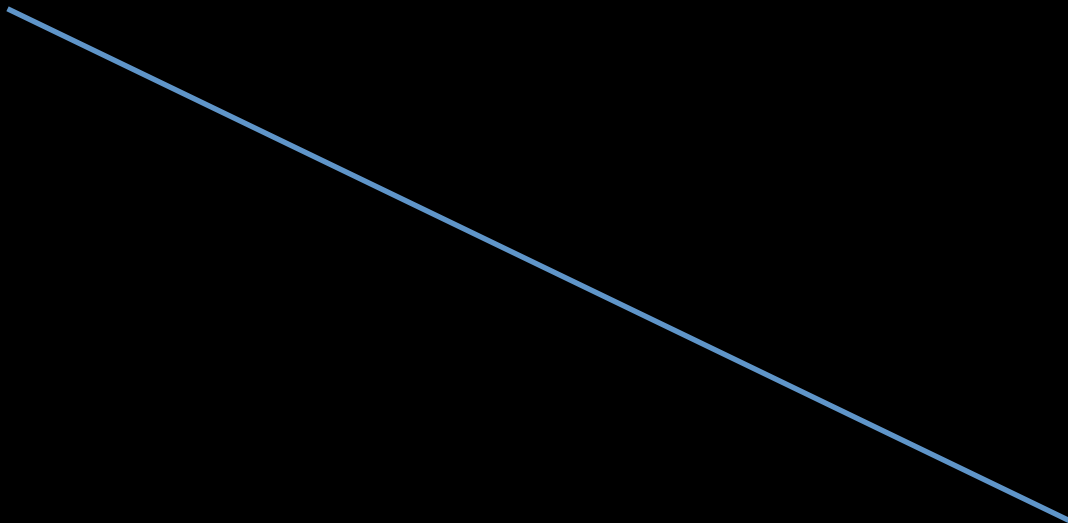
B

A



B

A



B

Shorter = Faster = Clearer

Why not aiming for a straight line?

- As long as you code you code...
- It will never be perfectly straight anyway.
- I can copy paste parts later.
- I will make it straight when I am done.

Why a straighter line?

- Make your code understandable and readable by you:
 - Reuse (now or in a year)
 - Debug
 - Avoid errors
 - What if you wanted to go to point C?
- Make the code understandable for others:
 - Reviewers
 - Members of your group
 - The whole world

Exercise:
`pythonstyle_exerciseA.py`
has some bad code

1. Make groups of 3-4
2. What does it do?
3. Why is it bad?

Summary of exercise

- Very linear
- Redundancy
- Some things should be defined as parameters
- Names of variables are cryptic
- Hard to read
- No comments
- Cryptic output
- Read from within the script

Structure your code

- *A good code is often read from the end of the file.*
- *Avoid redundancy!*
- Structure
 - Import modules at the top
 - Parameters
 - Functions
 - Main code

Aim for functional style

- It is very hard to see B in linear code; try to have higher order
- functions are great
- List comprehensions can be enough!

pythoninstyle_ExerciseB.py

Comments and docstrings

Block comment:

- Applies to all the code that follows it.

"""Docstrings"""

- modules, functions, classes.
- can be multiline.

Same line comment, can be very redundant

Pseudocode!

- Structure is hard, if you go straight in the writing your code will be more likely to be very linear and maybe even get confused.
- See pseudocode as a sort of “to do” list!

Pseudocode

- Pseudocode help structure a complex task.
- It does not have to follow any standard.
- You can gradually fill it in with real code!

Initialize total to zero

Initialize counter to zero

Input the first grade

while reading grades:

 add this grade into the running total

 add one to the grade counter

set the average to the total divided by the counter

print the average