

# Why Does My Computer Do That? Intro to Coding with Python—Code Reuse

Dr. Ab Mosca (they/them)

Slides based off slides courtesy of Jordan Crouser (<https://jcrouser.github.io/>)

# Plan for Today

- Quick review of “coding best practices”
- Some ethical questions
- Using online resources ethically
- How to attribute someone else’s code

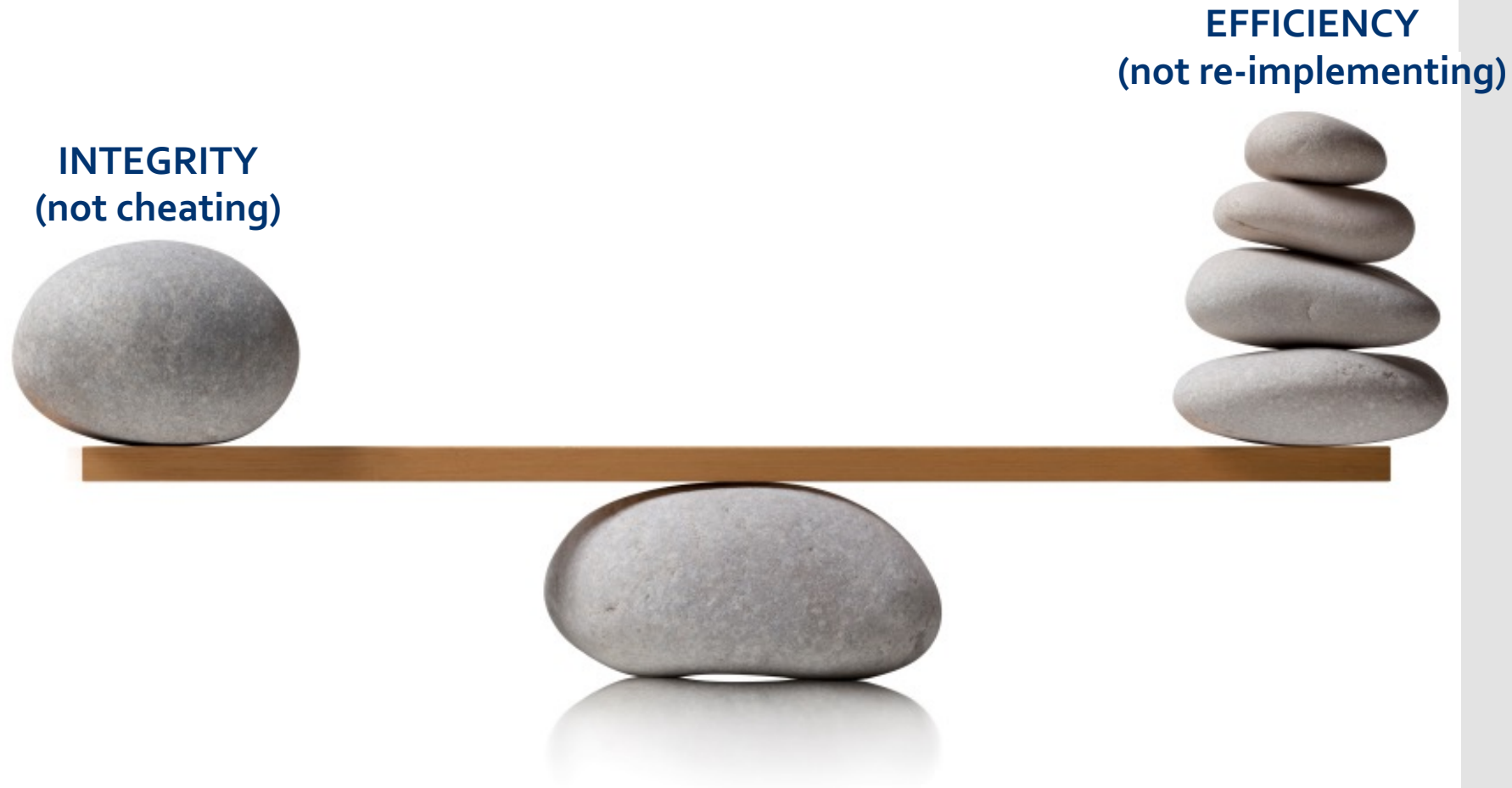
## Discussion

We've talked a lot about ideas like:

- "S4": start small | slow | simple
- Organizing your code so it's easy to reuse pieces
- Documenting your code so it's easy to come back to it
- Forking code from other people's repositories

Can you think of **ethical concerns** for any of this?

# The balancing act...



...so how do you know when it's **okay** to reuse code?

Let's consider  
a more familiar  
case...



Same or different?

## Dictionary

Enter a word, e.g. "pie"



# pla·gia·rism

/ˈplājəˌrizəm/ 

*noun*

the practice of taking someone else's work or ideas and passing them off as one's own.

*synonyms:* copying, infringement of copyright, [piracy](#), [theft](#), stealing; *informal* cribbing

"accusations of plagiarism"



Translations, word origin, and more definitions

## Scenario 0: self-reuse, not in a class

- You wrote a program that **solved a particular problem** as part of a project you're working on for fun
- Later on for **a different project** you're working on for fun, you need to solve the same problem
- Questions:
  - Can you reuse the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?

## Scenario 1: self-reuse, in- class work

- You wrote a program that **solved a particular problem** for a previous assignment in a course
- Later on for a **project you're working on for fun**, you need to solve the same problem
- Questions:
  - Can you reuse the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?



## Scenario 2: self-reuse, same course

- You wrote a program that **solved a particular problem** for a previous assignment in a course
- In a **later assignment for that same course**, you need to solve the same problem as part of a larger process
- Questions:
  - Can you reuse the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?
  - Does it matter if you copy/paste or **import** it?

## Scenario 3: self-reuse, different course

- You wrote a program that **solved a particular problem** for a previous assignment in a course
- In **an assignment for a different course**, you need to solve the same problem
- Questions:
  - Can you reuse the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?
  - Does it matter if it's the **whole assignment**, or just one part?

## Scenario 4: self-reuse, academic work →

- You wrote a program that **solved a particular problem** for an assignment in a course
- You later get a **job as a software engineer**, and you need to solve the same problem
- Questions:
  - Can you reuse the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?
  - Does it matter if it's the **whole assignment**, or just one part?

## Scenario 5: professors and TAs

- You are trying to **solve a particular problem** for an assignment in a course, but you are stuck
- You ask the professor or TA for advice, they walk you through **how to implement one of the functions**
- Questions:
  - Can you use the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?

## Scenario 6: peers

- You are trying to **solve a particular problem** for an assignment in a course, but you are stuck
- You ask a friend who took the class last year, they walk you through **how to implement one of the functions**
- Questions:
  - Can you use the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?

## Scenario 7: online sources

- You are trying to **solve a particular problem** for an assignment in a course, but you are stuck
- You look online to try to understand a concept, someone walks through **how to implement one of the functions**
- Questions:
  - Can you use the code?
  - Does it matter what the code **does**?
  - Do you need to **attribute** the code?

Common  
online Q&A  
resources



Demo: using  
StackOverflow





# How to attribute online code

```
# ----- START ATTRIBUTED CODE SECTION -----  
-  
# Code created with the help of Stack Overflow  
# https://stackoverflow.com/questions/49581417  
#  
# Question by Alden:  
#  
https://stackoverflow.com/users/9378177/alden  
#  
# Answer by CD Lane:  
#  
https://stackoverflow.com/users/5771269/cdlane  
  
# ...THE ACTUAL CODE GOES HERE...  
  
# ----- END ATTRIBUTED CODE SECTION -----  
--
```

# Discussion

For educational purposes, does copyright matter?  
Isn't everything covered by "**fair use**"?

## Rules for code reuse

- Always **attribute**
- Only use code you **actually understand**
- If it's for a course, **talk to the professor first**
- Understand **the license**, e.g. for StackOverflow



**Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0)**

## Activity: code reuse

- With your group, pick a simple game
- Use **online resources** to find a working python implementation
- **One partner:** make a new repl containing the code you found, add your partners as **collaborators**
- Write up the **attribution**
- Make sure that you **understand each piece of the code** (add lots of comments!)

Show and tell

What did you **find**?  
How does it **work**?