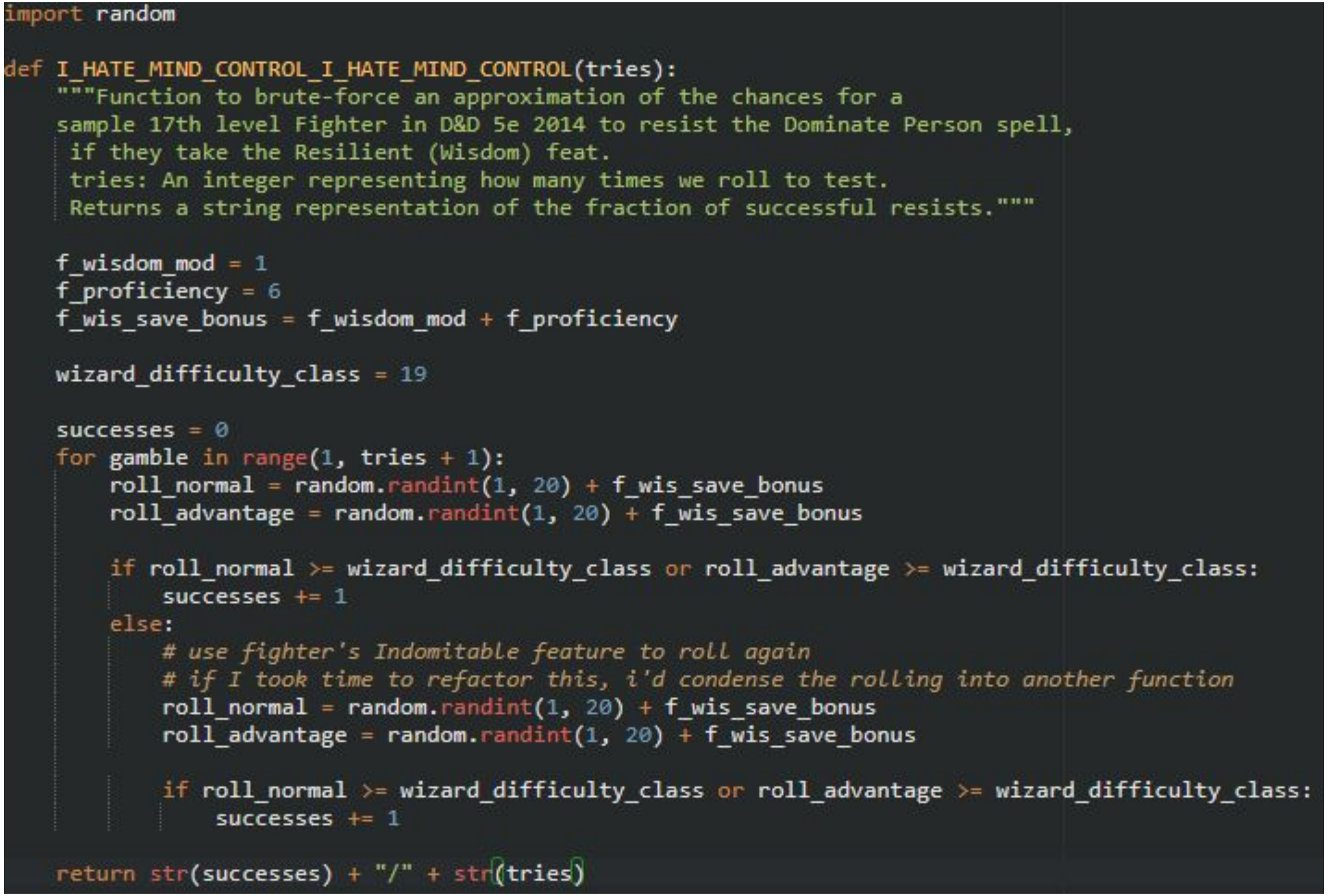
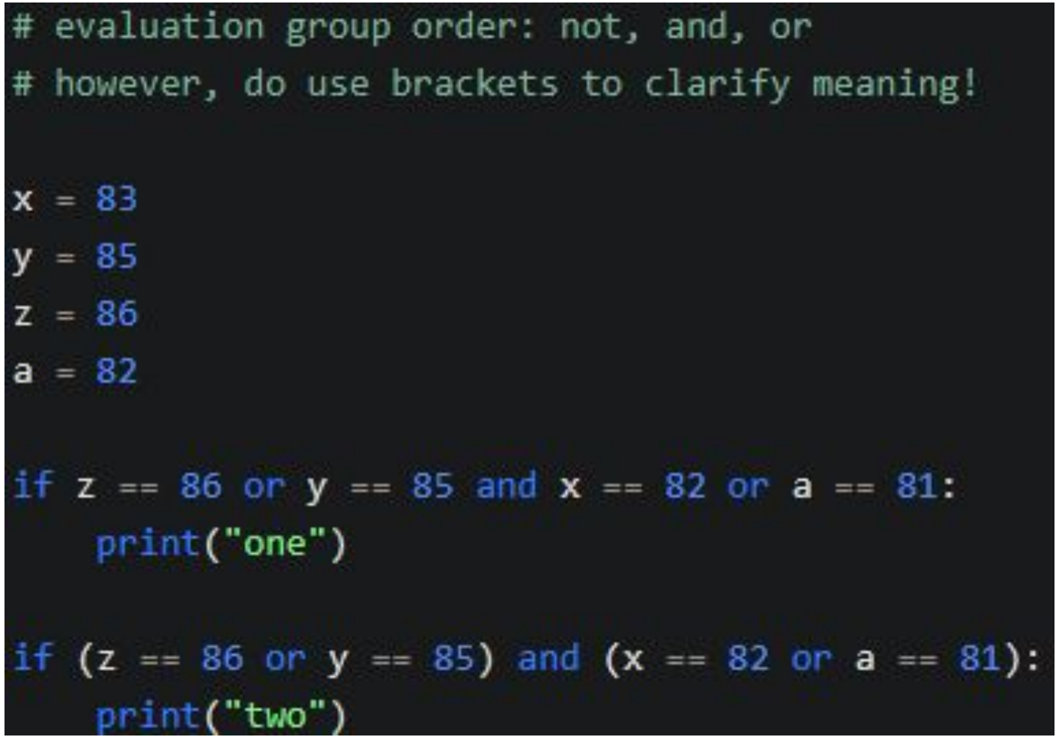


```python
import random

def I_HATE_MIND_CONTROL_I_HATE_MIND_CONTROL(tries):
    """Function to brute-force an approximation of the chances for a
    sample 17th level Fighter in D&D 5e 2014 to resist the Dominate Person spell,
     if they take the Resilient (Wisdom) feat.
     tries: An integer representing how many times we roll to test.
     Returns a string representation of the fraction of successful resists."""

    f_wisdom_mod = 1
    f_proficiency = 6
    f_wis_save_bonus = f_wisdom_mod + f_proficiency

    wizard_difficulty_class = 19

    successes = 0
    for gamble in range(1, tries + 1):
        roll_normal = random.randint(1, 20) + f_wis_save_bonus
        roll_advantage = random.randint(1, 20) + f_wis_save_bonus

        if roll_normal >= wizard_difficulty_class or roll_advantage >= wizard_difficulty_class:
            successes += 1
        else:
            # use fighter's Indomitable feature to roll again
            # if I took time to refactor this, i'd condense the rolling into another function
            roll_normal = random.randint(1, 20) + f_wis_save_bonus
            roll_advantage = random.randint(1, 20) + f_wis_save_bonus

            if roll_normal >= wizard_difficulty_class or roll_advantage >= wizard_difficulty_class:
                successes += 1

    return str(successes) + "/" + str(tries)
```

Branching and Repetition Expanded

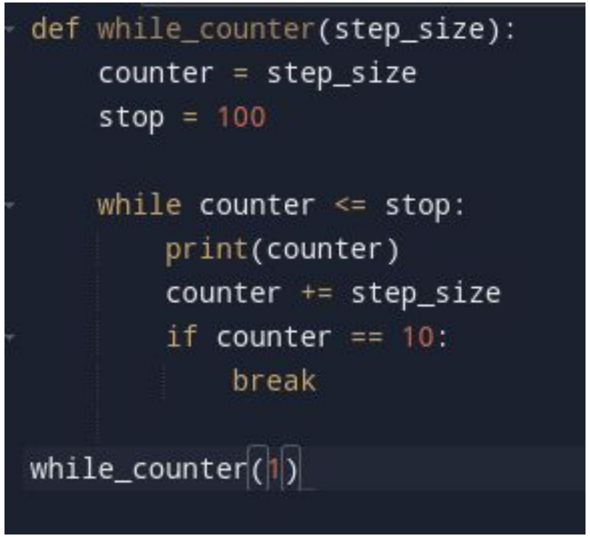
By Mark Luu, for CMPT 141

Exercise 1: Keyword priority



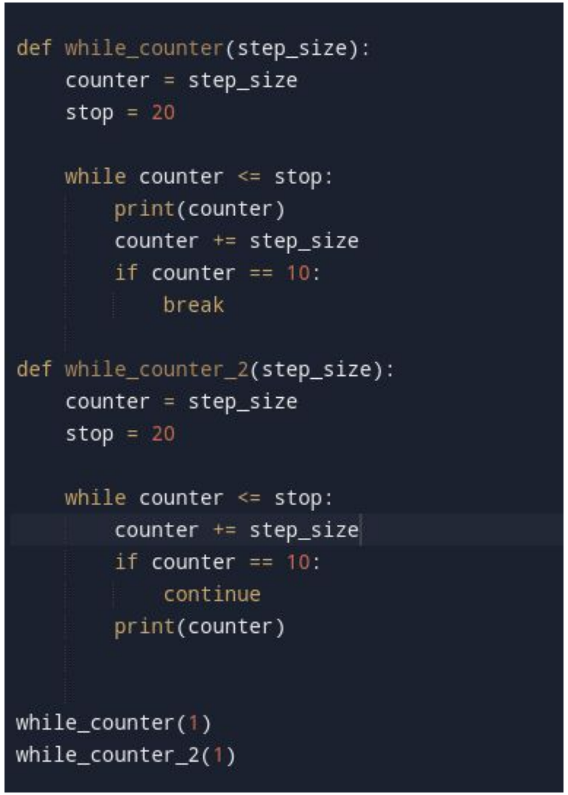
```python
# evaluation group order: not, and, or
# however, do use brackets to clarify meaning!

x = 83
y = 85
z = 86
a = 82

if z == 86 or y == 85 and x == 82 or a == 81:
    print("one")

if (z == 86 or y == 85) and (x == 82 or a == 81):
    print("two")
```

Exercise 2: Break and continue



```python
def while_counter(step_size):
    counter = step_size
    stop = 100

    while counter <= stop:
        print(counter)
        counter += step_size
        if counter == 10:
            break

while_counter(1)
```

Exercise 3: Break and continue



```python
def while_counter(step_size):
    counter = step_size
    stop = 20

    while counter <= stop:
        print(counter)
        counter += step_size
        if counter == 10:
            break

def while_counter_2(step_size):
    counter = step_size
    stop = 20

    while counter <= stop:
        counter += step_size
        if counter == 10:
            continue
        print(counter)


while_counter(1)
while_counter_2(1)
```