```python
def sora_is_dead_1_30_2018(dice_size):
    """Function to find the odds of
    Marc KOing Sora from full health
    in 100% Orange Juice.
    dice_size: The amount of sides on the dice.
    Post-conditions: None
    Returns: A string representation of fraction of events where Sora gets
        one-shot from full health.
    Also Mark's tears."""

    sora_health = 4
    marc_atk_mod = 1

    loss_count = 0
    fight_count = 0
    for attack in range(1, dice_size + 1):
        for defend in range(1, dice_size + 1):
            if (attack + marc_atk_mod) >= (defend + sora_health):
                loss_count += 1
            fight_count += 1

    return str(loss_count) + "/" + str(fight_count)
```

Modules, Slicing, Branching, Repetition

By Mark Luu, for CMPT 141

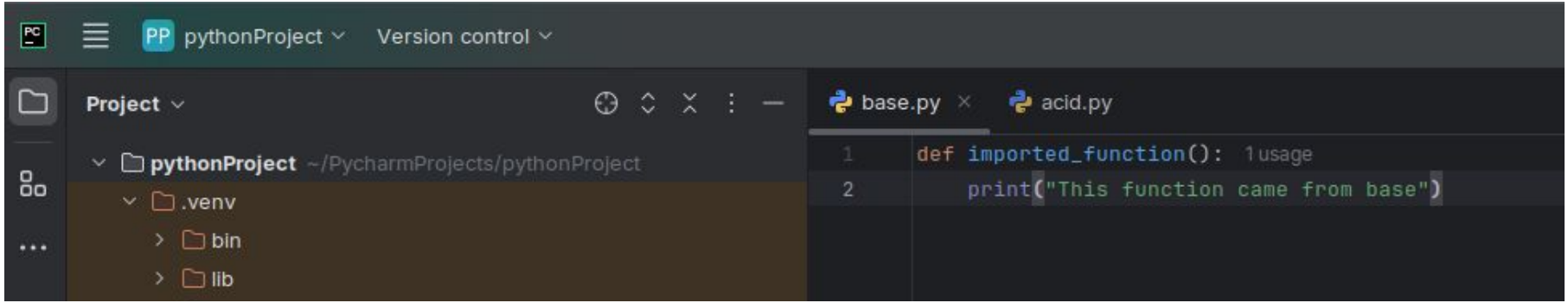
Carryover from last lab:

Parameters are what the function expects to receive

Arguments are what you actually give the function

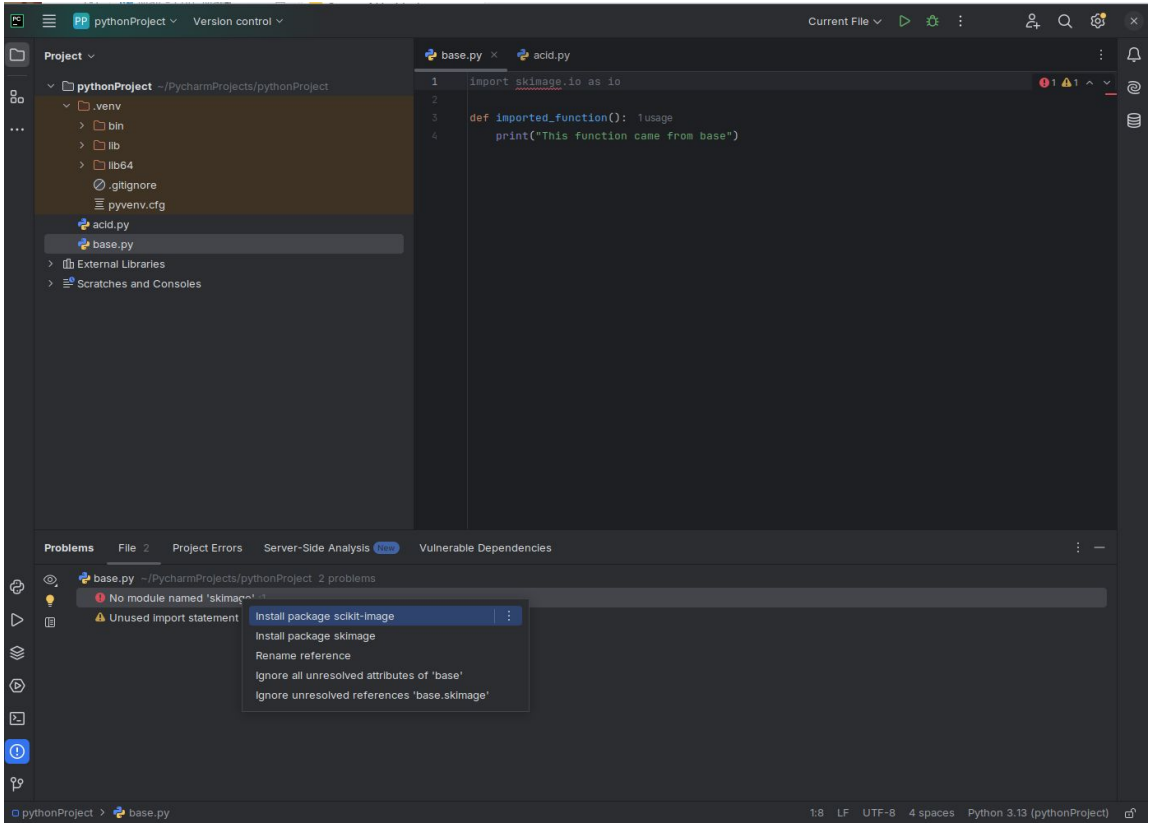
```
def function_name(param1):
    print(param1)

function_name(arg1)
```

Creating your own module:



Importing your own module:

Importing skimage, textbook style



Exercise 1: Creating some imported functions

Create a file named “start”.

In start, import the math library.

In start, create a function named “combinations” that returns the result of comb(16, 3).

In start, create a function named “power” with two parameters that prints pow(arg1, arg2).

In the same project, create a file named “end.”

In end, import start as s.

In end, print combinations().

In end, call power with the arguments 2 and 4 in that order.

WARNING!!!!!!!!!!!!!!!!!!

INDEXES START AT 0!!! NOT 1!!!! 0!!!!!!!!!!!!!!!!

Exercise 2: Index identification

```
srimp = ("I am the way, the truth, and the life. "
    + "No one comes to the father except through - $6 SRIMP SPECIAL!")

print(srimp[83:])
print(srimp[6:10])
print(srimp[1])
print(srimp[-8:])
print(srimp[-7:-5])

print(srimp[-6:-8])
print(srimp[999])
print(srimp[-6.9])
```

Exercise 3: Index and slice a string

Create a variable named “container” with the value “I am not a happy camper.”

Without creating another new variable, print the second word of container.

Print the last word of container.

Print the seventh last letter of container.

Exercise 4: Turning math into true and false

```
x = 83

y = 83

print(x < 82)
print((x > 82) and (x < 84))
print(x == 82)
print(x is 83)
print(y != x)
```

Exercise 4.5: Branching examples

```
x = 83
y = 85
if (x <= 82) and (y < 86):
    print("We're in if")
elif (x >= 84) or (x == 83):
    print("We're in elif")
else:
    print("X is 83")
```

Exercise 5: Creating branches in functions

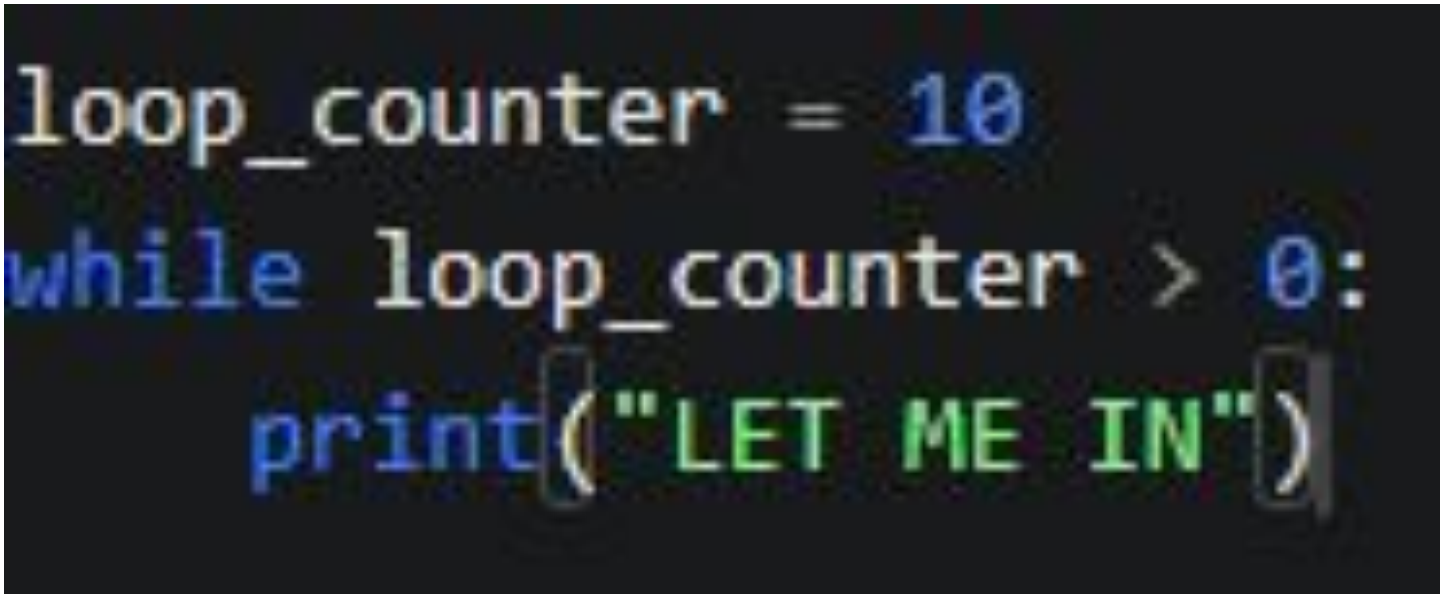
The len(arg1) function returns the length of the string argument in the function.

Create a function named len_category(arg1, arg2).

If the length of arg1 is less than 5 characters longer than arg2, len_category should return “Looks close enough to me.”
If it’s between 5 and 10 characters longer than arg2, len_category should return “We’re getting a little far...”

Otherwise, it should return “Oops! Too far!”

Exercise 6: Infinite loop



Exercise 7: Normal while loop

```
loop_counter = 10
while loop_counter > 0:
    print("LET ME IN")
    loop_counter -= 1
```

Exercise 8: Normal for loop

```
for step in range(1, 10):
    print(step)
```

Exercise 9: Nested for loop

```
for step in range(1, 4):
    for tiptoe in range(1, 6, 2):
        print(tiptoe)
    print(step)
```

Exercise 10: Creating repetition

Create a function named while_counter(step_size) that prints every step_size’th number from 1 to 100 using a while loop.

Create a similar function named for_counter(step_size) that prints with a for loop.

Import these functions into another file, and call those functions from that file.

Exercise in pain, revisited

```
def sora_is_dead_1_30_2018(dice_size):
    """Function to find the odds of
    Marc KOing Sora from full health
    in 100% Orange Juice.
    dice_size: The amount of sides on the dice.
    Post-conditions: None
    Returns: A string representation of fraction of events where Sora gets
        one-shot from full health.
    Also Mark's tears."""

    sora_health = 4
    marc_atk_mod = 1

    loss_count = 0
    fight_count = 0
    for attack in range(1, dice_size + 1):
        for defend in range(1, dice_size + 1):
            if (attack + marc_atk_mod) >= (defend + sora_health):
                loss_count += 1
            fight_count += 1

    return str(loss_count) + "/" + str(fight_count)
```