**Common Mistakes**

**Python Conditionals**

- Don’t forget the colon at the end of the conditional statement in an **if** or **elif**

**--- CORRECT**  **if num > 6**:

**# code here**

**XXX WRONG** **if num > 6** # ERROR can’t concatenate int to string

**# code here**

- Be extra careful with your conditional statements and double check to make sure you cover all bases. In the below example if **grossIncome** is equal to 500 it will be missed by both conditionals.

**if grossIncome < 500 :**

**# Code won’t execute if grossIncome == 500**

**elif grossIncome > 500 and grossIncome <= 700:**

**# Code won’t execute if grossIncome == 500**

- Understand when to use **and** and **or** especially when looking at specific number ranges. The below code will run EVERY TIME because num can be anything below 700 **or** anything above 500 which includes all numbers in existence (1, 0, -9999, 9999, etc.)

**XXX WRONG if num > 500 or num <= 700:**

**XXX WRONG** **# Code will ALWAYS execute**

**--- CORRECT if num > 500 and num <= 700:**

**--- CORRECT** **# Code will execute if (e.g. 501, 600, 650, 700)**

- Try to be neater with your interactions with the user and your outputs. While the answer below is correct, you can see that there is no consistency between the spaces around the equals sign.

Text

Description automatically generated

- When importing a python file as a module, you need to use the name of the import when referring to it. If there are variables or functions you want to use in your script you must use the imported module name, followed by a period, followed by the variable or function name **moduleName.variableName**.

The code below shows what is contained inside the python file called **myModule**. myModule

**# myModule.py**

**myString = “My name is James”**

**def myFunc():**

**print(“Hello”)**

**--- CORRECT**  **import myModule**

**--- CORRECT**  **print(myModule.myString)** **# “My name is James”**

**--- CORRECT**  **myModule.myFunc()** **# “Hello”**