

HW 2: Python Part 1

We've learned throughout the class that it's not about learning the tool, it's about learning how to think like an IS professional. For our first major HW, we will work with a very basic technical component. If you are interested in or already pursuing IS, consider this a stepping stone to future classes and career options.

What your Python program should do

Create a new repl for this homework.

1. Get three numbers from the user. Assume the user will type in only whole numbers. **The numbers could be negative or positive.** Don't forget to add helpful prompts for the user.
2. Using your knowledge of Python...
 - a. Determine which number is the smallest. Print that number along with descriptive text to inform the user of what is being printed on the screen.
 - b. Determine how many numbers are equal. Print that number (zero, two, or three) along with some descriptive text to inform the user of what is being printed to the screen.

Sample output

If the user enters:

3
4
6

Your program should print:

```
The smallest number is 3.  
There are zero equal numbers.
```

If the user enters:

3
6
6

Your program should print:

```
The smallest number is 3.  
There are two equal numbers.
```

If the user enters:

6
6

6

Your program should print:

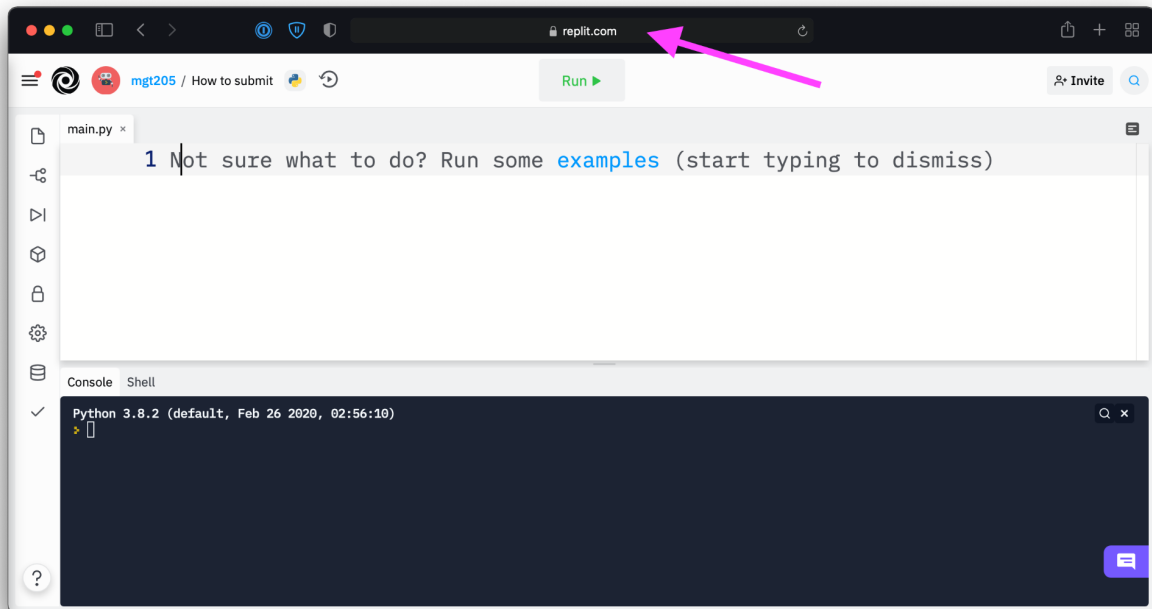
The smallest number is 6.

There are three equal numbers.

These are just examples of input and output. During grading, the user will be entering different numbers. If coded correctly, your program should work with any three whole numbers. I encourage you to try different sets of input to make sure your program can handle any valid input.

How to submit your Python code

When you are done with your code, click into your browser's address bar and copy that entire link.



Create a new text submission in iLearn. Paste your link into the text box (not the comments box) and submit.

Grading rubric

50 points total.

The TAs will be checking your output and your code.

There will be four sets of input known as test cases.

Test case J: 2, 14, 2

Test case K: 2, 2, 2

Test case L: -1, 2, -3

Test case M: 2, 1, 1

Enter those numbers in your program and you should get the appropriate output if your program is coded correctly. **Be careful: the wrong way** to do this HW is to explicitly check for those test cases. You will not earn any points for the test cases if you do this. Your program, if coded correctly, should organically work with any valid input.

Getting the correct smallest number is worth 2 points each.

Getting the correct number of equal numbers is worth 2 points each.

If you submit a program that either:

- a. Doesn't run
- b. Runs but misses all of the test cases

You will earn a 34 out of 50.

Blank files will earn a 0. Invalid submission links will receive a penalty as noted in the syllabus.