

## proj02: Iteration

### Part I WHILE LOOPS:

Open proj02\_01.py.

Write a program that prompts the user to enter numbers, one per line, ending with a line containing 0, and keep a running sum of the numbers. Only print out the sum after all the numbers are entered (at least in your final version).

Example:

```
Enter a number to sum, or 0 to indicate you are finished: 4
Enter a number to sum, or 0 to indicate you are finished: 5
Enter a number to sum, or 0 to indicate you are finished: 2
Enter a number to sum, or 0 to indicate you are finished: 10
Enter a number to sum, or 0 to indicate you are finished: 0
```

The sum of your numbers is: 21

*Hint:* Start by creating a variable called “sum” that is set to zero.  
Then, create a while loop that keeps looping *while* user input is not 0.  
After the loop ends, print out the sum.

### Part II FOR LOOPS:

Open proj02\_02.py

Write a program that asks the user how many Fibonacci numbers to generate and then generates them.

The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13 ...

Example:

How many Fibonacci numbers would you like to generate? 8

```
1
1
2
3
5
8
13
21
```

*Hint:* you will need several variables to keep track of these numbers!

Here is one way to set it up.

Previous number = 0

Current number = 1

Next number

During the for loop

    Print out the current number

    Set the next number to the previous number + the current number

    Set the previous number to the current number

    Set the current number to the next number

Extensions:

- Use a while loop instead of a for loop ( $n = 0$ , while  $n$  is  $<$  user\_number:). Remember to increment  $n$  each time!
- It is possible to solve this problem using ONLY previous number and current number – you don't need a next number variable. See if you can solve the problem WITHOUT a next number variable!!
- Instead of Fibonacci numbers, generate powers of 2
- Instead of Fibonacci numbers, generate all divisors of a number (*Hint:* % gives the remainder of two numbers, so  $8\%4 = 0$ , and  $8\%5 = 1$ ).