



Python Programming for All

Lab 6

INSTRUCTOR: Dr. Albert Kahira
albert@jengaschool.com

LOOPS

1. We will compare the control of the start, end, and step values in a counter loop, both in for and while loops
 - a. Write a function that finds the sum of numbers from 1 to 100 (both included) using a **for loop**. Write another function that does the same operation using a while loop.
 - b. Modify your functions such that summation starts from 5 instead of 1. Then further generalize such that the start value is entered by the user.
 - c. Modify both functions such that the end value is not 100, but it is entered by the user. You can use either `<` or `<=` in the while condition. Consider both cases, how would the boundary value change for both cases?
 - d. Modify the functions such that numbers increment by 2.
2. If you continually divide a number by 2, it will equal to zero after you do the division an infinite number of times: The sequence $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \dots \frac{1}{2^n}$ approaches 0 as n approaches infinity. However, when you use a float, because of the rounding errors, the result becomes zero after a number of steps (before infinity). Let's see this using a while loop.
 - a. In the beginning of the program, initialize a float variable to 1 (e.g. `x=1.0`). Then, in a while loop, continually divide `x` by 2 (`x=x/2`). When the loop

-
4. A user wants to purchase a list of items as much as the budget allows. Let's say the budget is 100TL. Write a program that reads item prices from the user one by one; prints the total up to that point after reading each price; and prints a message and stops when the total exceeds 100TL.
 5. We can use a while loop to make sure that the user inputs a value that satisfies certain conditions. We made a similar check before using if statements. But in order to be able to ask for a new input, you need a loop. Checking that an input satisfies certain conditions (e.g. a number is positive, the password is long enough, the **date** is valid, etc.) is called 'input validation'.
 - a. In the following program, input validation is used to make sure the number for the square root is non-negative. A while loop is used to wait until a valid input is entered.

```
def valid_squareroot():
    x = int(input("Enter the number for SquareRoot:"))

    while x<0:
        print("Number cannot be negative...")
        x = int(input("Enter the number for SquareRoot:"))
    print("The squareroot of", x, "is", x**0.5)
valid_squareroot()
```

- b. Based on the above example write a function that asks the user for a password, and repeats this until the user enters a password that you determine.

