| Unit 5 Loop Python Problems [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing)    *Unit 05 of Python Programming - Loop Python Problems* | |
| --- | --- |
| Learning Targets  This unit we will…  Explore the different types of loops and build games using the combined knowledge of control structures, loops and if statements.  I can…   * Use for loops when there is a definite way of knowing how many times you will need to loop, so that you can control your loop with a counter. * Use for loops with and without the range function. * Use while loops when you need a loop, but are not sure how many times the loop will have to happen, but instead use a condition to stop the loop. * use Boolean logic and comparison operators to control loops. * Combine loop and if control structures to make more advanced programs.   Vocabulary: control structures, for loop, while loop, range(), break, continue, Boolean | |
| Learn About It!  *You can explore some, or all of these resources. If you want to see a resource again, go for it!*  [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing) *These Collab documents review the concepts of each unit with code you can run and modify.* | |
| Evidence of Learning  *Complete the following programming exercises.*  [Grading Rubric](https://docs.google.com/document/d/1shjqolaw_5tSX9T5OJ2FZuBeon7K3hDrYEJ5m1ltSEw/edit?usp=sharing) | |
| Unit Programs  Review:   1. [Colab - For Loops](https://colab.research.google.com/drive/1E0wFPlbPzdNTtiy8EiHJ9uUrxE-zjEsY) 2. [Colab - While Loops](https://colab.research.google.com/drive/1i7mHdntcI_pHdSpUteEjS3GYLEchcqxr)   Once you have reviewed the Colab documents complete the problems below. There are tips, sample code, and links to sample code that you will use within the Colab documents, you also may want to refer back to early colabs. There will be two sets of problems to do, the first group can be done in a single file and the turtle program should be done in a separate file. There are pictures of what your output should look like below. Name the files **Unit05\_YourLastName.py and Unit05Turtle\_YourLastName.py**, if you do this set of problems in [repl.it](https://repl.it/) name the repl.it Unit05\_YourLastName and Unit05Turtle\_YourLastName and turn the share links into the classroom.  **YOU MAY NOT USE LISTS OR SEQUENCES TO SOLVE ANY OF THESE PROBLEMS!!**  **Unit05\_YourLastName**  **Sample output is shown below**   1. Countdown - (5 points) - Using [time.sleep()](https://docs.python.org/3/library/time.html#time.sleep) ask your user to input a number for a countdown and then your program should count down from that number to 1, there needs to be a delay of one second between each countdown number appearing on the screen.      1. 100 to 1000 Divisible by 5 or 6 - (15 points) - Write a program that will output every number between 100 - 1000 that is evenly divisible by the numbers 5 or 6. Output those numbers 10 per line.      1. Pyramid 1-15 - (20 points) - ask the user to end a number from 1-15, which will determine the size of the pyramid you will output. If the user inputs a number outside of the range ask again for the correct input until you get it. The pyramid will be composed of numbers like the two samples below. The pyramid cannot have sloping sides. If the user inputs a 1 the output will be a 1. You must solve the problem using loops.   Pyramid of 9  Pyramid of 15   1. Math Game - (20 points) - This is a simple addition/subtraction game. If the user gets five problems correct they win and if they get three wrong they lose. First ask the user if they want addition, subtraction, or both types of problems. If the user wants subtraction or both problems ask them if they want problems with negative answers or not. Use the same inputs, A, S, and B, as the sample output below. The A, S, and B inputs need to be [case insensitive](https://www.geeksforgeeks.org/isupper-islower-lower-upper-python-applications/#:~:text=In%20Python%2C%20upper()%20is,it%20returns%20the%20original%20string.). Depending on what the user selected you will generate random simple math problems with two numbers between 0-9. Give the user feedback on whether their problem is correct or not and keep track of how many right and wrong they get. If they get five correct they win, let them know and let them know how many wrong they got. If they get three problems wrong they lose, let them know how many problems they got correct. There are three sample outputs from three different runnings of the program below, your inputs and outputs should be similar. An efficient program should not repeat code for outputting addition or subtraction problems.          1. Number Guessing Game - (20 points) - For this program ask the user to enter a number between 1-100. Your program needs to guess the users number without using the number they input. Program will guess a number and output so the user can let it know if the guess was High H, Low L, or Correct C. User feedback input needs to be [case insensitive](https://www.geeksforgeeks.org/isupper-islower-lower-upper-python-applications/#:~:text=In%20Python%2C%20upper()%20is,it%20returns%20the%20original%20string.). When the program is correct let the user know how many guesses it took. There are two sample runs of the program below. Efficient code will not make random guesses.        1. Random Turtle - (20 points) - With a turtle draw cartesian coordinate grid that goes out to 300 in all four directions using goto statements instead of forward and turn. None of the random drawings can cross any of the X,Y axis’ and should stay roughly within the 300 outer limit.    1. In the upper left quadrant draw 100 randomly colored and placed circles with a radius of 10.    2. In the lower left quadrant draw 100 randomly colored and placed solid right triangles, with a hypotenuse of 60.    3. In the upper right draw 100 randomly placed Isosceles triangle and each side of every triangle has to be randomly colored. The side lengths are 60.    4. Finally in the lower right quadrant draw 100 random lines each a random color, the length can also be random.     Make sure you have a comment block at the top of your program with your name, the date and a list of the programs that are being run in the program. Also make sure to comment your variables, control structures, and each problem. Also use white space between the problems.  ############################################################  # Name : Date: #  # Unit 5 Problems #  # clock countdown, Divisible by 5 or 6, #  # display a pyramid, math game, Number Guessing Game #  # random turtle #  ############################################################  When your code works and is commented, turn it into the classroom. | |



That is, if your programs work!!