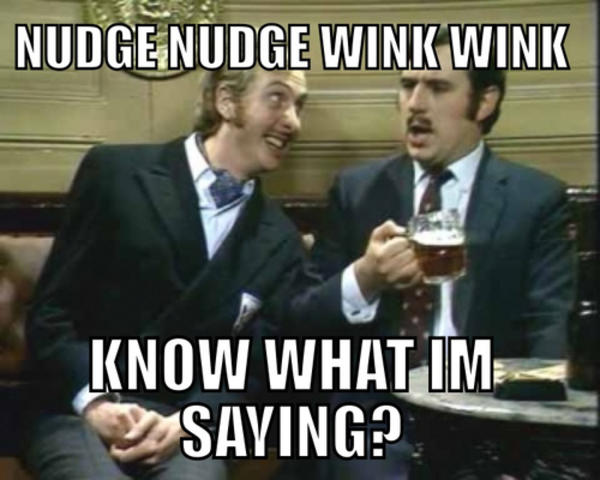
| Unit 4 Selection Python Problems [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing)    *Unit 04 of Python Programming - Selection Python Problems* | |
| --- | --- |
| Learning Targets  This unit we will…  Explore “if” statements and conditional Boolean logic, as well as random numbers.  I can…   * use “if” statements with boolean logic to make decisions in my programs * use “if” combined with “elif” statements or nested “if” statements to make a [logic decision tree](https://drive.google.com/file/d/13weRLFty6benE129gdczqlEzO3K5keMx/view?usp=sharing) within my program. * use Boolean logic and comparison operators to make decisions within my programs. * use random number generators within programs when needed.   Vocabulary: control structures, if and if elif statements, Boolean logic, logical operators, truth tables, comparison operators, operator precedence, random numbers, inclusive, exclusive. | |
| Learn About It!  *You can explore some, or all of these resources. If you want to see a resource again, go for it!*  [Collab Doc Index](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 - If Statements](https://colab.research.google.com/drive/1rQ7ou-A-_XkRKKr50JOa90efH_YZv-8z) 2. [Colab - Boolean Logic and Boolean Variables](https://colab.research.google.com/drive/1R_jyODc-fgKWV7YFVrNjH_ueU4A7qy1f) 3. [Colab - Random Numbers](https://colab.research.google.com/drive/1biSdIW4_Abk_WatIpWOTTCiOJchUWsJG)   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 **Unit04\_YourLastName.py and Unit04Turtle\_YourLastName.py**, if you do this set of problems in [repl.it](https://repl.it/) name the repl.it Unit04\_YourLastName and Unit04Turtle\_YourLastName and turn the share links into the classroom.  **YOU MAY NOT USE LISTS OR SEQUENCES TO SOLVE ANY OF THESE PROBLEMS!!**  **Unit04\_YourLastName**  **Sample output is shown below**   1. Sort three numbers - (15 points) - with one input get three numbers from the user and store them in three different variables. Output the numbers in a sorted order from the smallest number to the largest. **You may not use the min or max functions**. 2. Divisible by 5 and/or 6 - (15 points) - get a number input from the user and check to see if the number is equally divisible by 5 and/or 6. Output will tell the user one of the following: the number is equally divisible by 5 and 6, the number is equally divisible by either 5 or 6, or the number is not equally divisible by 5 or 6. 3. Rock, Paper, Scissor - (15 points) - The program will select a random number between 0-2 with scissor = 0, rock = 1 , and paper = 2. Ask the user: “Select Rock, Paper, or Scissors by entering a number scissor (0), rock (1), paper (2):”. Compare the computer's random number to the user’s input and output the result of the match. If they are the same it is a tie, scissor beats paper, rock beats scissor, and paper beats rock. Make sure to output what the computer got. 4. Random Card - (20 points) - Using a random number output a random card from a playing card deck like King of Hearts or 7 of Spades, any one of the 52 cards of a deck have to be possible. 5. Palindrome Number - (15 points) - a palindrome is something that is the same forwards or backwards. Ask the user to input a three digit number, create a reversed version of the number and compare them so you can output whether the number is a palindrome or not. No lists, sequences, or string methods allowed for solving this problem. 6. Turtle Circle Overlap - (20 points) - First draw the cartesian coordinate X,Y axis’ using goto statements instead of forward and turn. Both X and Y should go from 250 to -250. Centered on 0,0 draw a solid red circle with a black border and a radius of 2. With one input statement ask the user for the X,Y coordinates for the center point of a circle as well as the radius of the circle (ie "Enter circle1's center x-, y-coordinates, and radius: "). Then with a second input statement ask the same thing for a second circle. Draw the two circles and then write a message to the turtle window, to the lower left outside and a little away from the circles, telling the user if the second circle is completely inside the first, completely outside of the first, or if the circles overlap. Your program only needs to run once, outputting the two circles and the results. My example below ran three times to show all the possible results. For input in a repl.it turtle you cannot use eval. If you do x,y,z = input(''some message ").split(), you can input numbers with space inbetween, like 10 10 5. Those numbers will be strings so you will need to convert them x = str(x).    1. My input for my example was:       1. 100,100,10 and 50,50,20       2. -100,-100,75 and -80,-80,25       3. -100,125,75 and -50,75,20   To accomplish this task you will need to calculate the distance between the two points using the distance equation below and then factor in the radius of the circles to see their relationship. The distance equation requires finding a square root, you can use the radical to exponent rule shown below to raise the numbers to the power of 0.5.      Sample output of what your programs should look like is shown in the images below. 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 4 Problems #  # Sort 3 sort three numbers, divisible by 5 and/or 6 #  # Rock, Paper, or Scissors, Pick a card #  # Palindrome Number, turtle overlap #  ############################################################################  When your code works and is commented, turn it into the classroom. | |
| Above is what a sample input given to my turtle program looks like. Keep in mind your program only has to run once. My sample output of the turtle drawing has three outputs on one screen, your program does not have to do that. I did this so you could see three samples in one image, I used a loop and we haven’t covered loops yet so you do not have to add this feature. I only expect a single running of your program. | |



Are those circle directions clear? Say no more, say no more!!