| Unit 6 Function Python Problems [Learning Plan Index - Python](https://docs.google.com/document/d/1B5yWb6wCSRhqD42iWxCi7bmLPY2EqvU6pbiEQT0zs20/edit?usp=sharing)    *Unit 06 of Python Programming - Function Python Problems* | |
| --- | --- |
| Learning Targets  This unit we will…  Learn how to create and use functions, which will allow us to make our programs more modular.  I can…   * Create and call, or invoke, a function. * Pass parameters to a function and use them within the function while understanding the difference between pass by value and pass by reference. * Explain how functions make code modular and can eliminate redundancy. * Explain the difference between parameters and arguments. * Demonstrate the passing of parameters by position and keyword. * Create and use functions that either perform an action or return a value. * Use variables locally and globally within functions. * Understand and explain what scope means. * Use default arguments in my functions. * Explain what a function stub is and how to use one if needed.   Vocabulary: function, call or invoke, parameter, argument, pass by value, pass by reference, modular code, redundancy, pass by position, pass by keyword, return, local and global scope, default arguments, stub. | |
| 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 - Functions](https://colab.research.google.com/drive/1cIvFUOg2HgNVFUn0EUxvDZhVrPf2B2Ir)   Once you have reviewed the Colab document 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 **Unit06\_YourLastName.py and Unit06Turtle\_YourLastName.py**, if you do this set of problems in [repl.it](https://repl.it/) name the repl.it Unit06\_YourLastName and Unit06Turtle\_YourLastName and turn the share links into the classroom.  **YOU MAY NOT USE LISTS OR SEQUENCES TO SOLVE ANY OF THESE PROBLEMS!!**  **Unit06\_YourLastName**  **Sample output is shown below**   1. Sum of Digits - (10 points) -  Required function heading - **def sumdigits(numb):** Create a function that will take a single number parameter and add all the digits of that number and **return** the sum of those digits. Ask the user to enter a number, of any length, and pass that number into your function so that you can output the results from the **return** of that function. The output for problems 1-3 is below. 2. Sort 3 Numbers - (15 points) -  Required function heading - **def displaySortedNumbers(num1, num2, num3):** Create a function that will take three int parameters and for output will print those numbers in sorted order from smallest to largest. You may not use the min() and max() functions. Get input from the user for the three numbers and then pass those into the function. Sample output is in the image above. 3. Display Characters - (15 points) -  Required function heading - **def printChars(ch1, ch2, numPerLine):** Create a function that takes in two letters and an int as parameters from a user's input. Within the function you will have to make sure that ch1 is a letter that comes before ch2. Once you have confirmed, or set, the order of ch1 and ch2 print all the letters from ch1 to ch2, including ch1 and ch2, and use the int parameter to determine how many letters per line you should print. Assume letters are either lower to lower or upper to upper, there will be no case mixing. Sample output is in the image above. 4. Palindrome Number - (15 points) -  Required function heading - **def isPalindrome(inp):** Required function heading - **def reverseInt(inp):** The functions you create in this problem will be used in both problems #5 and #6 so keep the code you create here uncommented for use in those problems as well. For this problem you need to create two functions. reverseInt() will **return** a reversed int of its int parameter. isPalindrome() will use reverseInt() and with one line of code it will **return** a Boolean value that determines if its parameter is the same forwards and backwards, known as a palindrome. Get a number input from the user and pass it into isPalindrome() function to determine if it is a palindrome number or not. Use isPalindrome() in your output print statement. Sample output is in the image below.      1. Palindromic Prime - (15 points) -  Required function heading - **def isPrime(num):** Required function heading - **def PalindromicPrime(NumOfPrimes):** This problem will also use the isPalindrome() function you created in problem 4, so make sure you have not commented that problem out. The isPrime() function will return a Boolean value determined by whether or not the isPrime() parameter is a prime number or not. The PalindromicPrime() function will take an int parameter that will determine how many palindrome prime numbers it will output to the screen. A palindrome prime is a number that is the same forwards and backwards and is also prime. Ask the user to input the desired number of palindrome primes that should be output. The program will be tested for 100, make sure your output is 10 numbers per line and that you use format() so that the numbers are aligned in five spaces per number. Sample output is in the image above. 2. Emirp - Non Palindromic Prime whose reverse is prime - (15 points) - No function for this program, this will use the functions created in problems 4 & 5. For this problem you need to output 100 numbers that are prime, while not a palindrome, but the reverse of the number is also a prime number. Output 10 numbers per line using format to align them to five spaces per number. This problem does not require user input, a sample run is in the image below. 3. Turtle Functions - (15 points) -  Required function headings -  **def drawRectangle(color="black", x=0, y=0, width=30, height=30): def drawPolygon(color="black", x=0, y=0, numsides=4, length=30): def drawCircle(color="black", x=0, y=0, radius=50):** You will solve this problem in a separate file. You need to draw the cartesian coordinate system using goto statements instead of forward and turn. Next create the three functions based on their headings. These functions need to draw [solid shapes](https://docs.python.org/3/library/turtle.html#turtle.begin_fill) that have the attributes that are passed in the parameters or the default parameters if nothing is passed.    1. In the upper-left quadrant draw four pentagons using the drawPolygon() method, try and match the sample output in the image below.    2. In the upper-right quadrant draw five squares using the drawPolygon() method, try to match the sample output in the image below, you will need to adjust the turtle’s heading for each square and you should reset it to 0 before your next drawing.    3. In the lower-left quadrant draw four rectangles using the drawRectangle() method, create the same hashtag design with the same overlaps of the rectangles.    4. In the lower-right quadrant draw four circles using the drawCircle() method, try to match the sample output in the image below, you do need the white ring.     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 6 Problems #  # Sum of Digits, sort 3 numbers, Display Characters #  # Palindrome Number, Palindromic Prime, Emirp #  # Turtle Draw Shapes #  ############################################################  When your code works and is commented, turn it into the classroom. | |



I want my garden to look the same forwards and backwards!!