Assignment 3 – Functions

Assignment Due October 2, 2013

For this assignment, please note that NO jQuery may be used. You must write everything in core JavaScript.

1) Write a script that prompts the user for the radius of a circle, uses a function circleArea to calculate the area of the circle, and prints the area of the circle.

2) A parking garage charges a $2.00 minimum fee to park for up to three hours. The garage charges an additional $0.50 per hour for each hour or part thereof in excess of three hours. The maximum charge for any given 24-hour period is $10.00. Assume that no car parks for longer than 24 hours at a time. Write a script that calculates and displays the parking charges for each customer who parked a car in this garage yesterday. You should input from the user the hours parked for each customer. The program should display the charge for the current customer and should calculate and display the running total of yesterday's receipts. The program should use the function ***calculate-Charges*** to determine the charge for each customer. Use a text input field to obtain the input from the user.

3) Write function distance that calculates the distance between two points (x1, y1) and (x2,y2). All numbers and return values should be floating-point values. Incorporate this function into a script that enables the user to enter the coordinates of the points through an XHTML form.

4) Write a function integerPower(base, exponent) that returns the value of base exponent For example, integerPower(3, 4) = 3 * 3 * 3 * 3. Assume that exponent and base are integers. Function integerPower should use a for or while statement to control the calculation. Incorporate this function into a script that reads integer values from an XHTML form for base and exponent and performs the calculation with the integerPower function. The XHTML form should consist of two text fields and a button to initiate the calculation. The user should interact with the program by typing numbers in both text fields then clicking the button.

5) Write a function multiple that determines, for a pair of integers, whether the second integer is a multiple of the first. The function should take two integer arguments and return true if the second is a multiple of the first, and false otherwise. Incorporate this function into a script that inputs a series of pairs of integers (one pair at a time). The XHTML form should consist of two text fields and a button to initiate the calculation. The user should interact with the program by typing numbers in both text fields, then clicking the button.

6) Write a script that inputs integers (one at a time) and passes them one at a time to function isEven, which uses the modulus operator to determine whether an integer is even. The function should take an integer argument and return true if the integer is even and false otherwise. Use sentinel-controlled looping and a prompt dialog.

7) Write a function squareOfAsterisks that displays a solid square of asterisks whose side is specified in integer parameter side. For example, if side is 4, the function displays

****
****
****
****

Incorporate this function into a script that reads an integer value for side from the user at the keyboard and performs the drawing with the squareOfAsterisks function.

8) Write program segments that accomplish each of the following tasks:
    a. Calculate the integer part of the quotient when integer a is divided by integer
    b. Calculate the integer remainder when integer a is divided by integer b.

Then, use the programpieces developed in parts (a) and (b) to write a function displayDigits that receives an integer between 1 and 99999 and prints it as a series of digits, each pair of which is separated by two spaces. For example, the integer 4562 should be printed as 4 5 6 2

9) Implement the following functions:
    a. Function celsius returns the Celsius equivalent of a Fahrenheit temperature, using the calculation C = 5.0 / 9.0 * (F - 32);
    b. Function fahrenheit returns the Fahrenheit equivalent of a Celsius temperature, using the calculation F = 9.0 / 5.0 * C + 32;

Use these functions to write a script that enables the user to enter either a Fahrenheit or a Celsius temperature and displays the Celsius or Fahrenheit equivalent. Your XHTML document should contain two buttons—one to initiate the conversion from Fahrenheit to Celsius and one to initiate the conversion from Celsius to Fahrenheit.

10) Write a function minimum3 that returns the smallest of three floating-point numbers. Use the Math.min function to implement minimum3. Incorporate the function into a script that reads three values from the user and determines the smallest value.

11) An integer number is said to be a perfect number if its factors, including 1 (but not the number itself), sum to the number. For example, 6 is a perfect number, because 6 = 1 + 2 + 3.Write a function perfect that determines whether parameter number is a perfect number. Use this function in a script that determines and displays all the perfect numbers between 1 and 1000. Print the factors of each perfect number to confirm that the number is indeed perfect. Challenge the computing power of your computer by testing numbers much larger than 1000. Display the results in a <textarea>. An integer is said to be prime if it is greater than 1 and divisible only by 1 and itself. For example, 2, 3, 5 and 7 are prime, but 4, 6, 8 and 9 are not.
    a. Write a function that determines whether a number is prime.
    b. Use this function in a script that determines and prints all the prime numbers between 1 and 10,000. How many of these 10,000 numbers do you really have to test before being sure that you have found all the primes? Display the results in a <textarea>.

c. Initially, you might think that n/2 is the upper limit for which you must test to see whether a number is prime, but you only need go as high as the square root of n. Why? Rewrite the program using the Math.sqrt method to calculate the square root, and run it both ways. Estimate the performance improvement.

12) Write a function that takes an integer value and returns the number with its digits reversed. For example, given the number 7631, the function should return 1367. Incorporate the function into a script that reads a value from the user. Display the result of the function in the status bar.

13) The greatest common divisor (GCD) of two integers is the largest integer that evenly divides each of the two numbers. Write a function gcd that returns the greatest common divisor of two integers. Incorporate the function into a script that reads two values from the user.

14) Write a function qualityPoints that inputs a student's average and returns 4 if the student's average is 90–100, 3 if the average is 80–89, 2 if the average is 70–79, 1 if the average is 60–69 and 0 if the average is lower than 60. Incorporate the function into a script that reads a value from the user.

15) Write a script that simulates coin tossing. Let the program toss the coin each time the user clicks the Toss button. Count the number of times each side of the coin appears. Display the results. The program should call a separate function flip that takes no arguments and returns false for tails and true for heads. [Note: If the program realistically simulates the coin tossing, each side of the coin should appear approximately half the time.]

16) Computers are playing an increasing role in education. Write a program that will help an elementary-school student learn multiplication. Use Math.random to produce two positive one-digit integers. It should then display a question such as **How much is 6 times 7?** The student then types the answer into a text field. Your program checks the student's answer. If it is correct, display the string "Very good!" and generate a new question. If the answer is wrong, display the string "No. Please try again." and let the student try the same question again repeatedly until the student finally gets it right. A separate function should be used to generate each new question. This function should be called once when the script begins execution and each time the user answers the question correctly.

17) Write a script that plays a "guess the number" game as follows: Your program chooses the number to be guessed by selecting a random integer in the range 1 to 1000. The script displays the prompt Guess a number between 1 and 1000 next to a text field. The player types a first guess into the text field and clicks a button to submit the guess to the script. If the player's guess is incorrect, your program should display Too high. Try again. or Too low. Try again. to help the player "zero in" on the correct answer and should clear the text field so the user can enter the next guess. When the user enters the correct answer, display Congratulations. You guessed the number! and clear the text field so the user can play again. [Note: The guessing technique employed in this problem is similar to a binary search, which we discuss in Chapter 10, JavaScript: Arrays.]

18) Write a recursive function power(base, exponent) that, when invoked, returns base exponent for example, power(3, 4) = 3 * 3 * 3 * 3. Assume that exponent is an integer greater than or equal to 1. The recursion step would use the relationship:

$$base^{exponent} = base \cdot base^{exponent - 1}$$

and the terminating condition occurs when exponent is equal to 1, because $base^1$ = base Incorporate this function into a script that enables the user to enter the base and exponent.

a. It is interesting to watch recursion in action. Modify the factorial function above to display its local variable and recursive-call parameter. For each recursive call, display the outputs on a separate line and add a level of indentation. Do your utmost to make the outputs clear, interesting and meaningful. Your goal here is to design and implement an output format that helps a person understand recursion better. You may want to add such display capabilities to the many other recursion examples and exercises throughout the text.