

Assignment 6 – JavaScript Objects

Due Wednesday, June 19, 2013

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

- 1) Write a script that uses random number generation to create sentences. Use four arrays of strings called article, noun, verb, and preposition. Create a sentence by selecting a word at random from each array in the following order: article, noun, verb, preposition, article, and noun. As each word is picked, concatenate it to the previous words in the sentence. The words should be separated by spaces. When the final sentence is output, it should start with a capital letter and end with a period. The arrays should be filled as follows: the article array should contain the article "the", "a", "one", "some", and "any"; the noun should contain the nouns "boy", "girl", "dog", "town", and "car"; the verb array should contain the verbs "drove", "jumped", "ran", "walked", and "skipped"; the preposition array should contain the preposition "to", "from", "over", "under" and "on". The program should generate 2- sentences to form a short story and output the result to an HTML5 text area. The story should begin with a line reading "Once upon a time..." and end with a line reading "THE END".
- 2) A limerick is a humorous five-line verse in which the first and second lines rhyme with the fifth, and the third line rhymes with the fourth. Using techniques similar to those developed in question 1, write a script that produces random limericks. Polishing this program to produce good limericks is a challenging problem, but the result will be worth the effort!
- 3) Write a script that inputs a telephone number as a string in the form (555) 555-555. The script should use String method split to extract the code as a token, the first three digits of the phone number as a token, and the last four digits of the phone number as a token. Display the area code in one text field and the seven-digit phone number in another text field.
- 4) The availability of computers with string manipulation capabilities has resulted in some rather interesting approaches to analyzing the writings of great authors (as well as tweets, movie reviews, and other text). Much attention has been focused on whether William Shakespeare really wrote the works attributed to him. Some scholars believe there is substantial evidence indicating that Christopher Marlowe actually penned some of these masterpieces. Researchers have started using computers to find similarities in the writings of these two authors. This exercise examines three methods for analyzing texts with a computer. (Put these all together in a program ... shape can be at your discretion ...)
 - a. Write a script that reads several lines of text from the keyboard and prints a table indicating the number of occurrences of each letter of the alphabet in the text. For example, the phrase,

To be, or not to be: that is the question:
Contains one "a" two "b's" no "c's" etc.
 - b. Write a script that reads several lines of text and prints a table indicating the number of one-letter words, two-letter words, three letter words, etc., appearing in the text. For example, the phrase:

Whether 'tis nobler in the mind to suffer
Contains no words with 1 letter, 2 words with 2 letters, etc..

- c. Write a script that reads several lines of text and prints a table indicating the number of occurrences of each of the different words in the text. The first version of your program should include the words in the table in the same order in which they appear in the text. A more interesting (and useful) printout should then be attempted in which the words are sorted alphabetically.
- 5) Computers are frequently employed in check-writing systems such as payroll and accounts payable applications. Many strange stories circulate regarding weekly paychecks being printed (by mistake) for amounts in excess of \$1 million. Incorrect amounts are printed by computerized check-writing systems because of human error and / or machine failure. Systems designers build controls into their systems to prevent erroneous checks from being issued. Another serious problem is the intentional alteration of a check amount by someone who intends to cash a check fraudulently. To present a dollar amount from being altered, most computerized cheque-writing systems employ a technique called check protection. Checks designed for imprinting by computer contain a fixed number of spaces in which the computer may print an amount. Supposed a paycheck contains eight blank spaces in which the computer is supposed to print the amount of a weekly paycheck. If the amount is large, then all eight of those spaces will be filled, for example:

1,230.60 (cheque amount)

12345678 (position numbers)

On the other hand, if the amount is less than \$1,000, then several of the spaces would ordinarily be left blank. For example,

99.87 (cheque amount)

12345678 (position numbers)

Contains three blank spaces. If a check is printed with blank spaces, it is easier for someone to alter the amount of the check. To prevent a check from being altered, many check writing systems insert leading asterisks to protect the amount as follows:

*****99.87 (cheque amount)**

12345678 (position numbers)

- a) Write a script that inputs a dollar amount to be printed on a check, then prints the amount in check-protected format with leading asterisks if necessary. Assume that nine spaces are available for printing the amount. Embed this functionality on a nicely laid out form that enables the user to enter the amount of the cheque. After a "process" button is clicked, make sure that the protected amount (with asterisks) is printed in a new field on the form. Generate an alert that tells the user the cheque has been processed.