Web Client Programming

Assignment 5

Due: Next Section. (*one day before if you submit by email***)**

Notes: This assignment is individual assignment, every student should complete it by himself.

1. True or False?

   a. ECMAScript variables are loosely typed.
   b. In JavaScript, uninitialized variables have no values.
   c. It’s possible to define your own data types in JavaScript.
   d. Typeof function requires parentheses to operate correctly.
   e. The literal undefined value is provided mainly for comparison.
   f. In JavaScript, Boolean type has two values true equal to number 1, and 0 equal to number 0.
   g. The Boolean() function converts the Boolean value true to 1, and the Boolean value false to 0.
   h. It’s recommended to use float numbers arithmetic as a conditions .
   i. Special numeric value NaN is equal to null.
   j. When applying the Number() function to undefined it returns NaN.
   k. When applying the Number() function to empty string it returns NaN.
   l. When applying Number(“5.3ab”) it returns 5.
   m. When applying parseInt(“5.3ab”) it returns 5.
   n. When applying parseFloat(“5.3ab”) it returns 5.
   o. Strings are mutable in ECMAScript, meaning that you can change their values after initialization.
   p. Firefox and Internet Explorer 6 had very slow string concatenation.
   q. When the unary plus is applied to a nonnumeric value, it performs the same conversion as the Number() casting function.
   r. ECMAScript bitwise operations converts values first into a 64-bit binary, and then apply the operation.
   s. If var x=2<<5 then x =64.

t. The logical NOT operator first converts the operand to a Boolean value and then negates it .
u. If Infinity is multiplied by 0, the result is NaN.
v. If Infinity is divided by Infinity, the result is NaN.
w. If var x=5+”5” then x=10.
x. The equality operators convert operands first into a numeric value before checking.
y. ECMAScript If-statement automatically converts the result of an expression into a Boolean by implicitly calling Boolean() function.
z. In ECMAScript you must specify whether the function should return value or not.
aa. An ECMAScript function doesn’t care how many arguments are passed in, nor does it care about the data types of those arguments
bb. Arguments in ECMAScript create a function signature that must be matched later on.
cc. If two functions are defined to have the same name in ECMAScript, it is the last function that becomes the owner of that name
dd. JavaScript permits direct access of memory locations.
ee. With reference values, you can add, change, or delete properties and methods at any time
ff. All function arguments in ECMAScript are passed by value.
gg. In ECMAScript, there are only two primary types of execution contexts, global and function.
hh. JavaScript’s lack of block-level scopes
ii. JavaScript is a garbage-collected language.
jj. JavaScript follow two garbage-collecting strategies, Mark-and-Swap and Reference Counting.
kk. Mark-and-Swap garbage collecting suffers from circular references deadlock.
ll. Give a null value to each reference after usage would increase the performance of your JavaScript code.

2. Visit the following link, and work through sections 6 to 10
https://www.codecademy.com/en/tracks/javascript