JavaScript Reference Guide for CS170

All assignments, quizzes, and exams must use **ONLY** those concepts presented in the reference guides for this course. **Credit will only be given for mastery of the concepts described/listed in the reference guides**.

Variables Names

- Variables names begin with a lower-case letter.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive.
- Camel-case is used for variable names that are more than 1 word.
- Variable names are descriptive.
- Examples:
    - Good variable names: ***taxRate***, ***average***, ***highScore***
    - Bad variable names: ***a, x, n***

Variables Declarations

- To declare a variable the ***var*** statement will be used.
- Example:
    - ***var*** highScore**;**

JavaScript Value Types

Type	**Example**
Number	7
Boolean	true
String	“hello”

Note: Unlike other programming languages, JavaScript does not require the declaration of a type when creating a variable.

Strings

- Strings in JavaScript are surrounded by either single quotation marks, or double quotation marks.
- The first character in a string is in position 0.
- ***length(str)*** is the number of characters in the string, ***str***.
- Strings can be concatenated with other strings using the ‘+’ operator:
    - ***var str1 = “hello”, str2 = “ world”;***
    - ***alert(str1 + str2);***

      hello world will be displayed

User Input

- User input is achieved by using the ***prompt*** function in JavaScript. Be aware that the result returned is a string. If you want the user to enter a ***number,*** you must convert the value returned to a number by using the ***Number*** method.
    - var name = prompt (“Enter your name “);
    - var num1 = Number (prompt (“Enter your age “));

Comments

- Comments may be included by typing // at the beginning of the comment.
- Example:

    // This is a comment

Assignment Statements

- Assignment statements are used to assign a value to a variable. Evaluate the expression on the right of =. Assign the result of the expression to the variable on the left of the =.
- Examples (next examples assume the variables have already been declared):
    - **price = 5;**
    - **str = “apple”;**
    - **count = count + 1;**
    - **result = 2 * 4 + 7 % 3;** // NOTE: *This evaluates to 9.*

Operators

Arithmetic Operators	**x = 4 y = 5**
+	Addition : x + y = 9
-	Subtraction: x – y = -1
*	Multiplication: x * y = 20
/	Division: x / y = 0.8
%	Modulus: y % x = 1

Relational (Comparison) Operators	**x = 4 y = 5**
==	Equal: x == 4
!=	NOT equal: x != 5
<	x < y
>	y > x
<=	x <= y
>=	y >= y

Logical Operators	**JavaScript logical operators**	**x = 4** **y = 5** **z = 7**	**JavaScript examples**
AND	&&	x < y AND y < z *results in **true**;* x < y AND z < y *results in **false***	x < y **&&** y < z (***true***) x < y **&&** z < y ***(false)*** x == 4 **&&** y == 5 ***(true)***
OR	||	x > y OR y > z *results in **false***; x < y OR z < y *results in **true***	x > y **||** y > z ***(false)*** x < y **||** z < y ***(true)*** x = 5 || x = 4 ***(true)***
NOT	!	NOT(x < y) results in **false**	**!**(x < y) ***(false)*** **!**(x == y) ***(true)***

Conditionals

Only the following formats are acceptable in CS170.

Conditional	Acceptable syntax	Example

Unary conditional (no else)	***if Boolean condition*** *is* ***true***
	execute these statements
		if (num < 10)
		alert(“number < 10”);

If-Else	***if Boolean condition*** *is* ***true***
	execute these statements
	else
	execute these statements when Boolean condition is ***false***
		If (num < 10)
		alert(“number < 10”);
		else
		alert(“number >= 10”);

Nested conditionals	***if Boolean condition1*** *is* ***true***
	execute these statements
	else
	if Boolean condition2 *is* ***true***
	execute these statements when Boolean condition1 is false and Boolean condition2 is true
	else
	execute these statements when Boolean conditions 1 and 2 are false
		if (num < 10) **{**
		alert(“less than 10”);
		alert(“number is small”);
		}
		else {
		if (num % 2 == 0)
		alert (“even number >= 10”);
		else
		alert(“odd number >= 10”);
		alert(“number is large”);
		}

If – Else If -Else	***if Boolean condition1*** *is* ***true***
	execute these statements
	else if *Boolean condition2 is true*
	execute these statements when Boolean condition1 is false and Boolean condition2 is true
	else
	execute these statements when Boolean conditions 1 and 2 are false
		if (num > 50)
		alert(“high number”);
		else if (num > 30)
		alert(“medium number”);
		else
		alert(“low number”);

Math Module

	Description	Example
Math.PI	returns value of PI	**Math.PI** returns 3.141592653589793
Math.round(x)	Returns x rounded to its nearest integer	Math.round(6.7) returns 7 Math.round(6.2) returns 6
Math.ceil(x)	Returns x rounded up to its nearest integer	Math.ceil(7.3) returns 8 Math.ceil(7) returns 7
Math.floor(x)	Returns x rounded down to its nearest integer	Math.floor(7.8) returns 7 Math.floor(7)returns 7
Math.random()	returns a random number between 0 (inclusive), and 1 (exclusive):	Always returns a random non-negative value < 1.

Functions

Acceptable Syntax *Note: a function may have 0 or more parameters*	**Example**

function name(***parameters***){
execute some statements
return some value
}
	function hoursToSeconds(numHours) {
	var numSeconds = numHours*60*60;
	return numSeconds;
	}

function name(***parameters***){
execute some statements
}
	function greet(name) {
	alert ("Hello, " + name)
	}

Arrays

- Arrays are used to store multiple items in a single variable.
- ***list1.length*** is the number of elements in the array ***list1.***
- Create an array using square brackets.
  - **var list1 = [3, 8, 9, 4 , 2];**
- Arrays may also be declared as follows (empty array for five elements):
  - **var list1 = new Array(5);**
- Array elements may be accessed individually:
  - **list1[0] = 3**
- The first element in an array, **list1**, is in position ***0***; The last element in this array is in position ***list1.length – 1.***

Loops (CS170 covers only FOR loops)

For loop use	**Example**	**Result that is alerted**

FOR loops	for (j = 0; j < 3; j = j + 1) {
	alert(j);
	}
		0, 1 ,2

Nested **FOR** loops	for (j = 0; j < 3; j = j + 1) {
	for (k = 0; k < 2; k = k + 1) {
	alert (j * k);
	}
	}
		0, 0, 0, 1, 0, 2