Variable Scope

Variables have a scope in every javascript file. Scope refers to where a variable is “seen” by the interpreter.

Scope is important because there can be naming conflicts in which the same name is used more than once. This can cause problems and unreliable results.

There are two scopes (Local and Global)

Local

Local variables are defined within a function.

Global

Global variables are defined outside of a function. They are susceptible to naming conflicts.

Even though you may not use a variable name more than once, a script you import may use a name that you have already used.

Example:

In the example below the variables have been classified according to their scope.

**Local – area, w, h**

**Global – wallSize, el1**

Function getArea(w, h) {

var area = w \* h;

return area;

}

var wallSize = getArea(4,7);

var el1 = document.getElementById(“first”);

el1.textContent = wallSize;

Variable storage

A variable references a value that is stored in memory. The same value can be used with more than one variable.