

# Variables

A variable is letter or word used to represent a value that can be altered. The main types of variables are bool, string, int, and float. Bool is when the variable can be true or false. String is a letter, word, or phrase. Int is an integer (whole number). Float is a number with a decimal (can also be a whole number).

Examples:

```
int x = 1;
int k = 73;
float r = 5;
float pi = 3.1415;
bool tf = false;
bool w = true;
string h = "h";
string aa = "lava";
```

## Local and Global Variables

Local and global variables are variables that have a certain scope compared to the script in which it is located in. There are certain levels of scope to a script, there is the function's scope, then there is the local scope, then there is the global scope. A function's scope consists of everything within its braces ( { } ). Normal variables as shown above will only work in a function unless carried over to another function.

A local variable's scope is the whole entire script, so if you made a local variable, then you wouldn't have to worry about having to carry it over from function to function. The downside to it is that you can't use the value of the local variable in commands. It is mostly used to check and see if the player has done multiple things within the map to make something greater happen. This is also the case with global variables, but it can be used over many levels in your custom story. Currently, I don't yet know how to properly use global variables, for I never had the need to use it. Here is an example of a local variable integer:

```
void OnStart()
{
    SetLocalVarInt("Var01", );
    AddEntityCollideCallback("Player", "ScriptArea_1", "Func01", true, 1);
    SetEntityPlayerInteractCallback("Object_1", "Func02", true, 1);
}
void Func01(string &in asParent, string &in asChild, int alState)
{
    AddLocalVarInt("Var01", 1);
    Func03();
}
void Func02(string &in asEntity)
```

```
{
    AddLocalVarInt("Var01", 1);
    Func03();
}
void Func03()
{
    if (GetLocalVarInt("Var01") == 2)
    {
        AddDebugMessage("Run!", false);
    }
}
```

This is what it says in the Script Functions page about what GetLocalVarInt, AddLocalVarInt, and SetLocalVarInt are and also how there can be a GetLocalVarFloat, AddLocalVarFloat, and SetLocalVarFloat. There are also GetLocalVarString, AddLocalVarString, and SetLocalVarString.

## Local

Local variables can be used throughout the same script file.

```
void SetLocalVarInt(string& asName, int alVal);
void AddLocalVarInt(string& asName, int alVal);
int GetLocalVarInt(string& asName);
```

```
void SetLocalVarFloat(string& asName, float afVal);
void AddLocalVarFloat(string& asName, float afVal);
float GetLocalVarFloat(string& asName);
```

```
void SetLocalVarString(string& asName, const string& asVal);
void AddLocalVarString(string& asName, string& asVal);
string& GetLocalVarString(string& asName);
```

## Global

Global variables can be used throughout several maps and can be accessed by several script files.

```
void SetGlobalVarInt(string& asName, int alVal);
void AddGlobalVarInt(string& asName, int alVal);
int GetGlobalVarInt(string& asName);
```

```
void SetGlobalVarFloat(string& asName, float afVal);
void AddGlobalVarFloat(string& asName, float afVal);
float GetGlobalVarFloat(string& asName);
```

```
void SetGlobalVarString(string& asName, const string& asVal);
void AddGlobalVarString(string& asName, string& asVal);
```

```
string& GetGlobalVarString(string& asName);
```

The SetVar is used to set the variable's value. The AddVar is used to add the given number or letter(s) to the value of the existing variable. The GetVar is used when checking the value of the variable.

**This wiki entry has been made by Kyle S. If you have any comments or need help with this, send me a private message on the Frictional Games Forum. (My name on there is Kyle)**

From:

<https://wiki.frictionalgames.com/> - Frictional Game Wiki

Permanent link:

<https://wiki.frictionalgames.com/hpl2/tutorials/script/localandglobalvariables?rev=1310587479>

Last update: **2011/07/13 21:04**

