



## Variables in Flash

### Naming rules

Variable names can be no longer than 64 characters and must start with a letter. Beyond the first character it is possible to use numerals and underscore. Interestingly, it is possible to dynamically generate variable names which break these rules and appear to work just fine.

Note that variable names in Flash are not case sensitive so that Number4 is the same as NUMBER4 or number4. Remember that the stored values are case sensitive (the condition "NUMBER4" eq "number4" would return false).

### Scoping

All variables in Flash are associated with a specific timeline. There is no such thing as a global variable in Flash. It is perfectly fine to have a frogs variable in the parent timeline and another frogs variable within a movie clip. Each is treated as a completely different entity. To reference a variable in a different timeline it is necessary to include a target (e.g./instance\_name:frogs). You can also reference a variable in another level (e.g. \_level5/instance\_name:frogs).

### Variable types

Flash does not use different types of variables. Essentially they are all treated as strings and converted into numeric values if they are dealt with in that context. Note that there are different operators for each context. To compare the equality of two numbers use "=", to compare two strings use "eq". What happens if you use the condition "hello" = "goodbye"? In this case, Flash treats each string as 0 and the statement would return true.

The trick is to remember to use the right set of operators depending on whether you are dealing with numbers or strings.

### Capacity

Flash numbers can hold up to 15 significant figures. The length of strings appears to be a practical one. It is possible to have strings of over 20,000 characters if you are willing to live with the big performance hit.

## Cleaning up

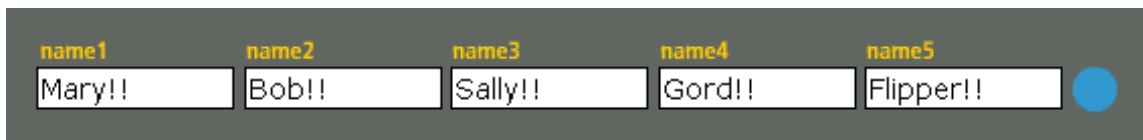
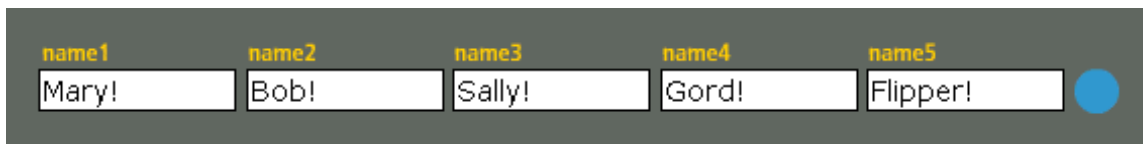
It is worth noting that while you can set a variable to an empty string there is no way to completely remove all traces of a variable once it is created. This can be an issue when you are submitting variables to a server for processing.

## Arrays in Flash

One of the very powerful features of actionscript is dynamic variable naming.

Variable names can be generated on the fly based on an expression. This trick is exploited to build arrays in flash. Arrays are essentially lists of similar data. An example would be a list of names. When you collect together a set of values in an array it becomes very easy to perform an action on the whole set. One, two and multi-dimensional arrays can be created using actionscript.

Here is an example using a one dimensional array. The button on the right of the text fields appends a "!" at the end of each of the values stored in name1, name2, etc. It does this by looping through all the values and changing them one at a time.



Here is the code for the button:

```
On (Release)
  Loop While (i < 5)
    Set Variable: "i" = i+1
    Set Variable: "name"&i = eval("name"&i)&"!"
  End Loop
  Set Variable: "i" = "0"
End On
```

You'll see that we use an expression "name"&i for the variable name. So the first time through the loop it is changing "name1", the second "name2", and so forth.

You can use the same idea any time you have a set of variables or instance names which you want to act on as a group.