# Variables:

## How an application remembers things

You use variables to store information temporarily, while your application is running. In this tutorial, learn how to use variables, put them together with text, do calculations, and name them effectively.

### Key topics covered in this tutorial

- Variables are temporary storage space
- Variables are lost when an application stops running
- A variable can contain a number or some text
- How to manipulate text with variables
- How to do calculations with variables
- Recommendations for naming variables

**See also:** Documentation: Containers, variables and sources of value

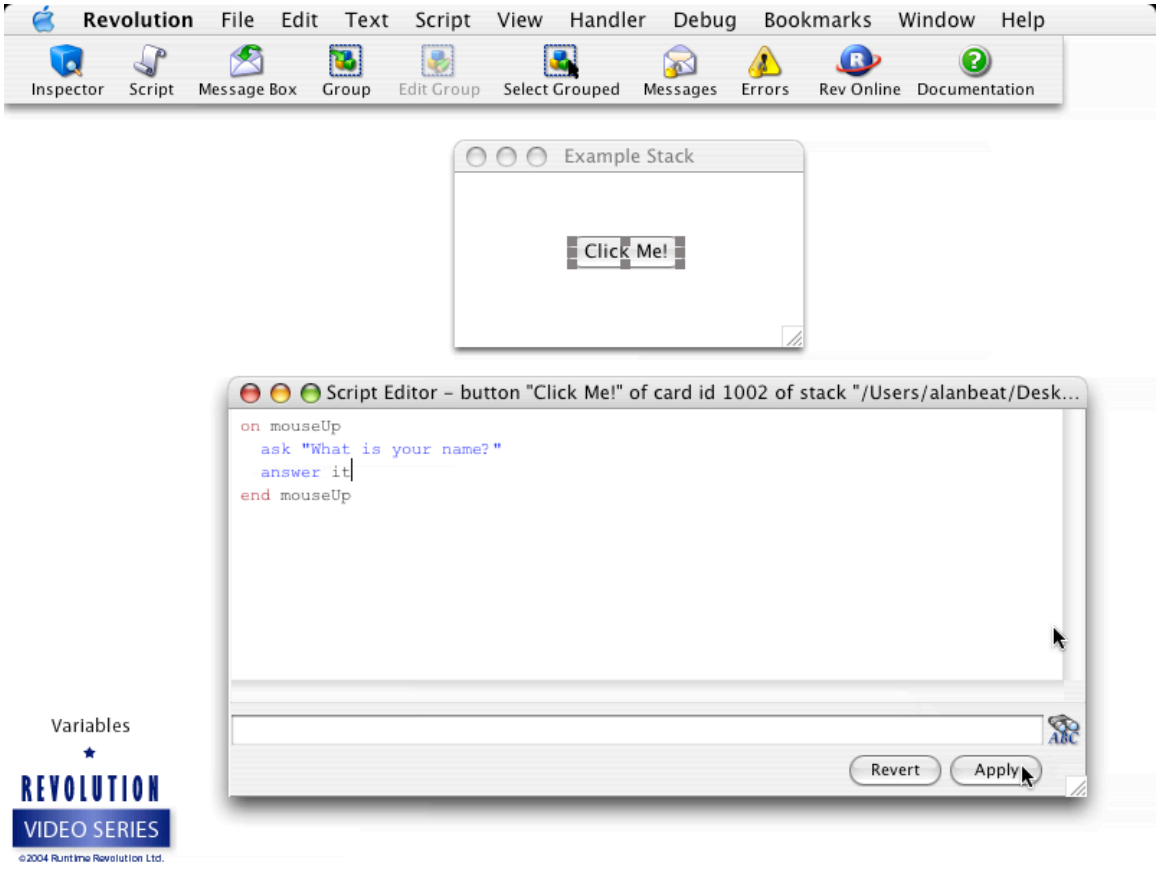
Computer applications frequently ask users to enter information they retrieve from somewhere, whether that is a file, a database or on the Internet. In order to be able to work with this information, you need to be able to store it. Variables are one of the main ways of storing this information. Variables are stored in Random Access Memory (RAM), so they are very fast to access, and also temporary. Some types of variable are stored for the entire time your application is running, others only exist for a short time when a specific script is running – in other words, when a script finishes, their content ceases to exist.

Suppose you want to store the information that a user has typed into a dialog. You use the 'ask' command to ask the user a question.

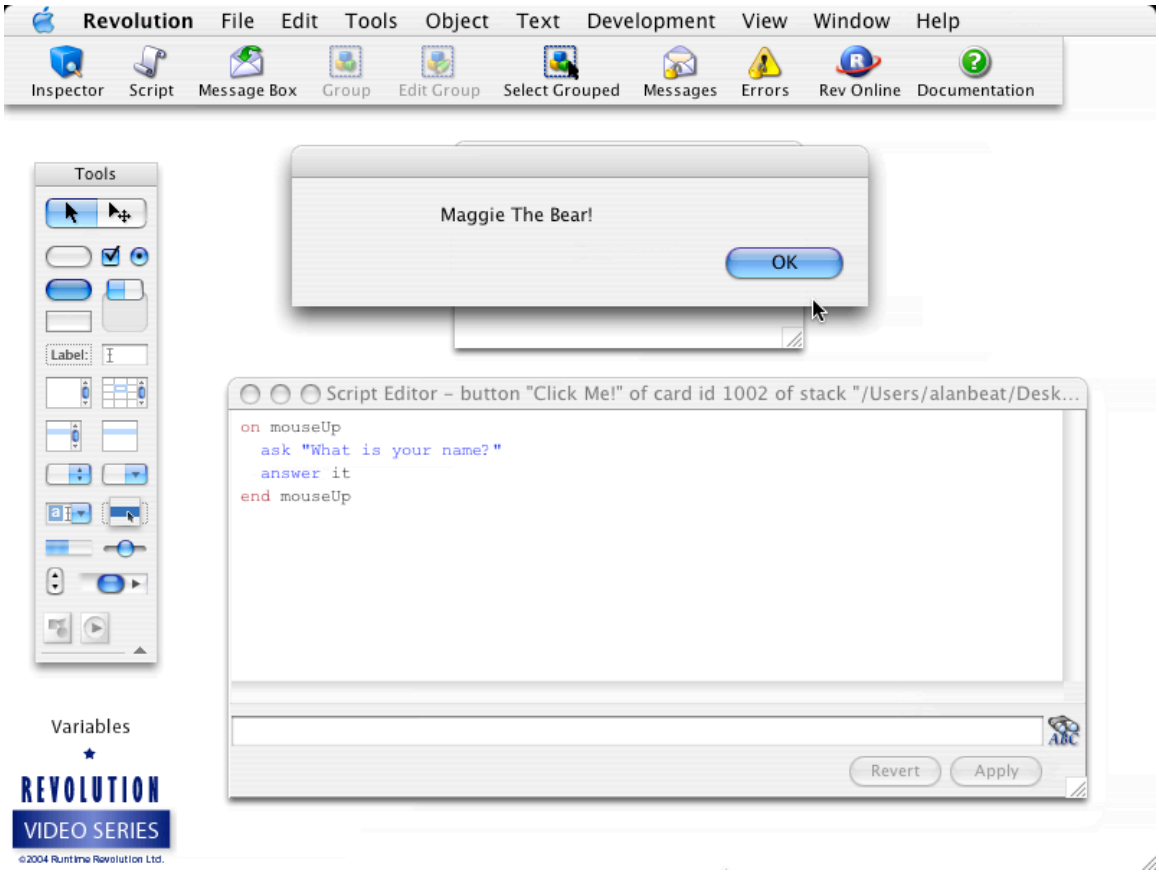
```
ask "What is your name?"
```

When the user enters a response to this question, the text they enter is stored in a variable called 'it'. Let's display the information they entered in a dialog.

```
answer it
```



And now we have a dialog showing the name we just entered.




The variable 'it' is a temporary variable; it doesn't stay around long. If you want to preserve the value to be used later in the script, you can use the 'put' command to place the value of 'it' into another variable.

```
put it into tUserName
```

Variables names have to be one word long, and that word can't be used for any other purpose in Transcript already. To ensure that you use a unique word that you can remember, we

recommend you put ‘t’ in front of the word you use, and use a descriptive name that you’ll remember later in the script. The ‘t’ stands for ‘the’ or ‘temporary’.

Now that we’ve safely stored the user name in a variable, we’re free to ask another question and store the answer to it. Let’s ask "What is your age?"

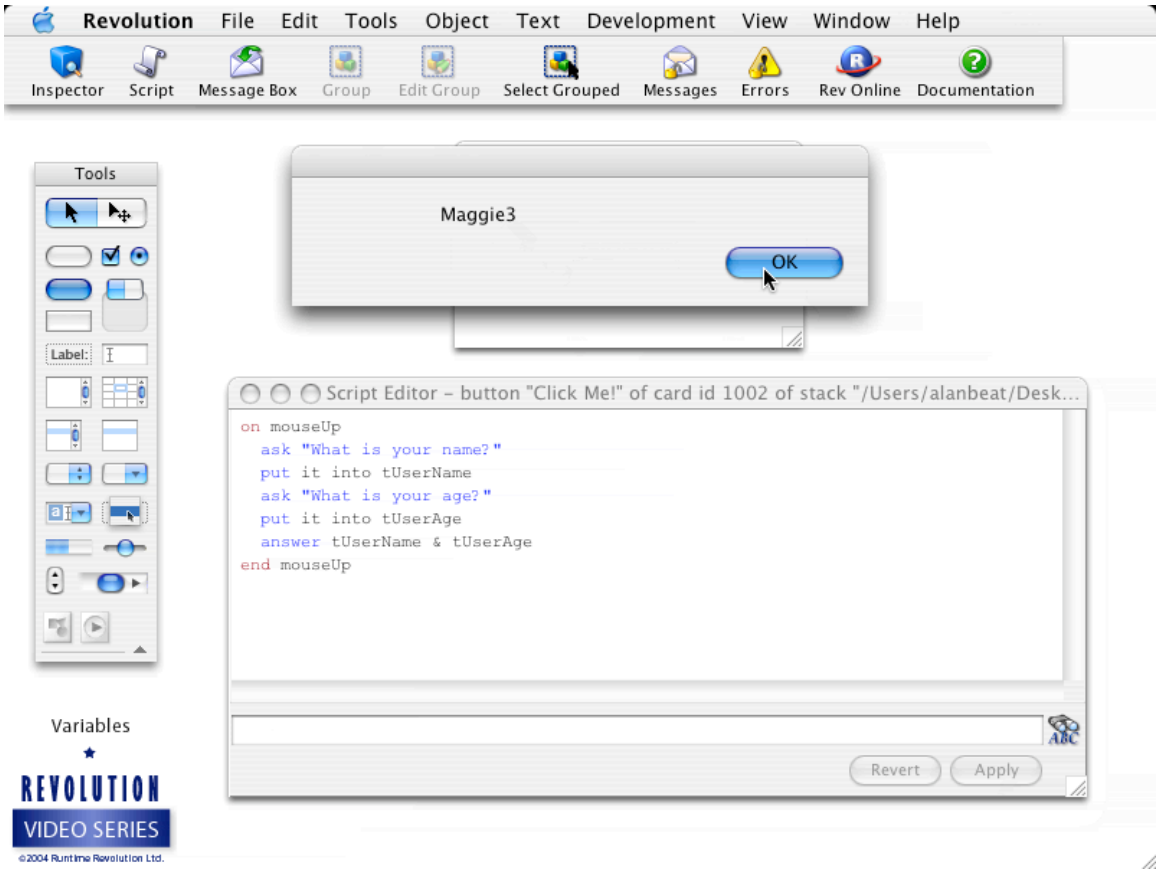
```
ask "What is your age?"
```

We’ll store the answer to that in the variable tUserAge.

```
put it into tUserAge
```

Now that we have the name and the age, let’s bring them both together and display them in a dialog. To put two things together you use the ampersand (&) character.

```
answer tName & tAge
```




And there we have a dialog with the name and age we entered. But wait, there is no space between them. And in fact, the dialog doesn't really tell us much about what it contains. It would be better if there was a complete sentence in the dialog that tells the user what the dialog is displaying in a more user friendly way.

Let's try putting some text together with these variables. We've already entered text to be displayed before when we did 'Hello World' – text in a script is always enclosed in quotes to show that it is to be taken literally as it is written in the script; in other words that it is not a variable. We'll change

```
answer tUserName & tUserAge
```

to

```
answer "My name is " & tUserName & ".  My age is " & tUserAge
```

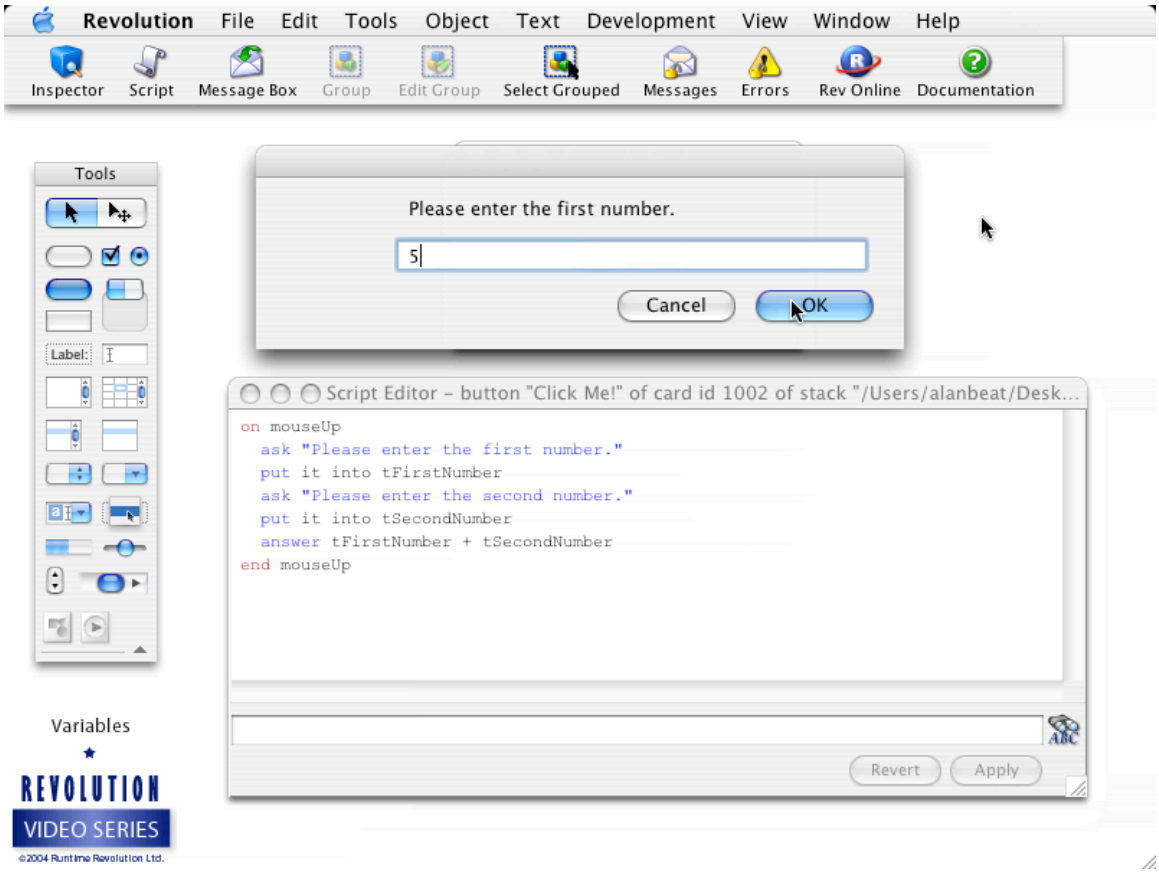
The script looks quite complex, but it is fairly straightforward. The text enclosed in quotes is to be displayed exactly as we have written it, and then we have the ampersand which connects the text to the next piece of information to display, in this case a variable, tUserName. After the variable we have another ampersand which connects the next piece of literal text, and finally an ampersand which connects the second variable containing age. This simple feature is incredibly powerful and can be used to put text together or to perform formatting operations. We'll look at this in more detail in the tutorials on working with text.

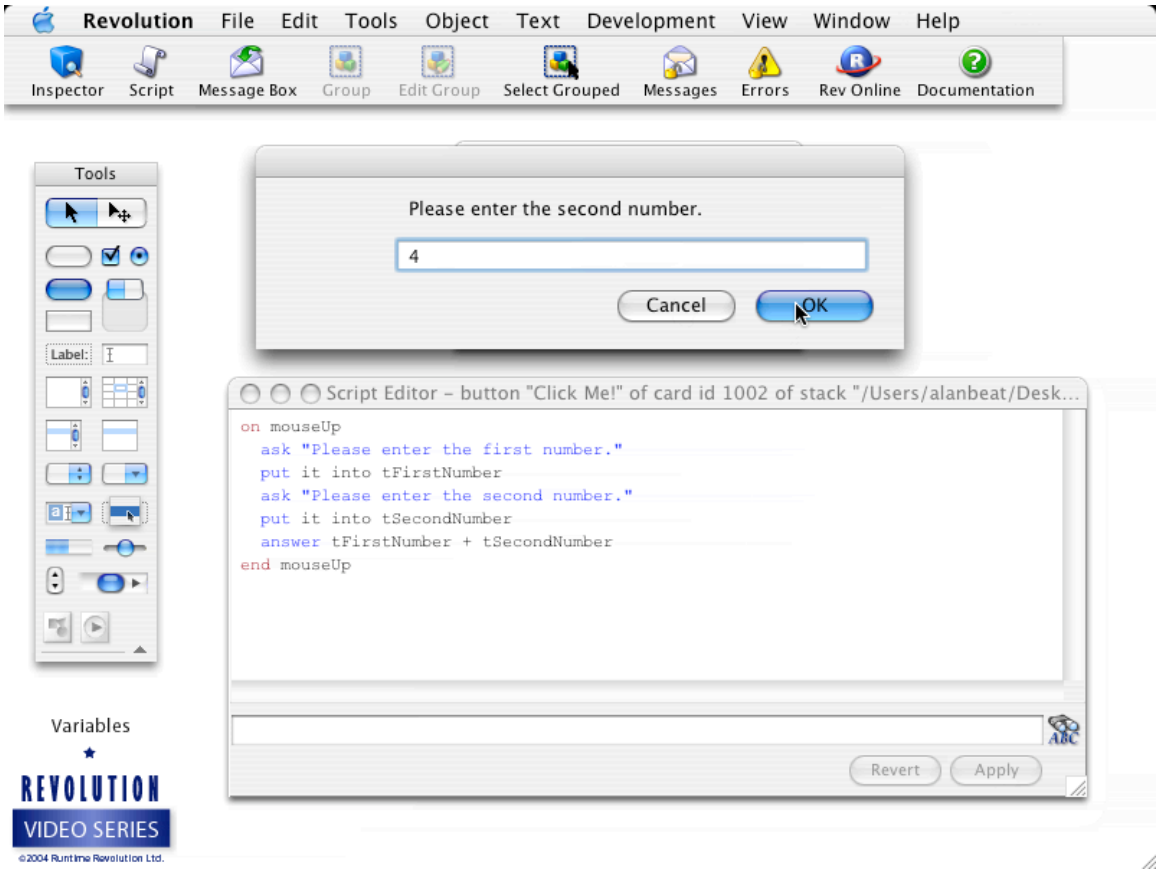
Finally, let's take a look at how we use variables to do calculations. We'll create a simple application that asks the user to enter two numbers and then adds them together.

```
ask "Please enter the first number."
put it into tFirstNumber
ask "Please enter the second number."
put it into tSecondNumber
```

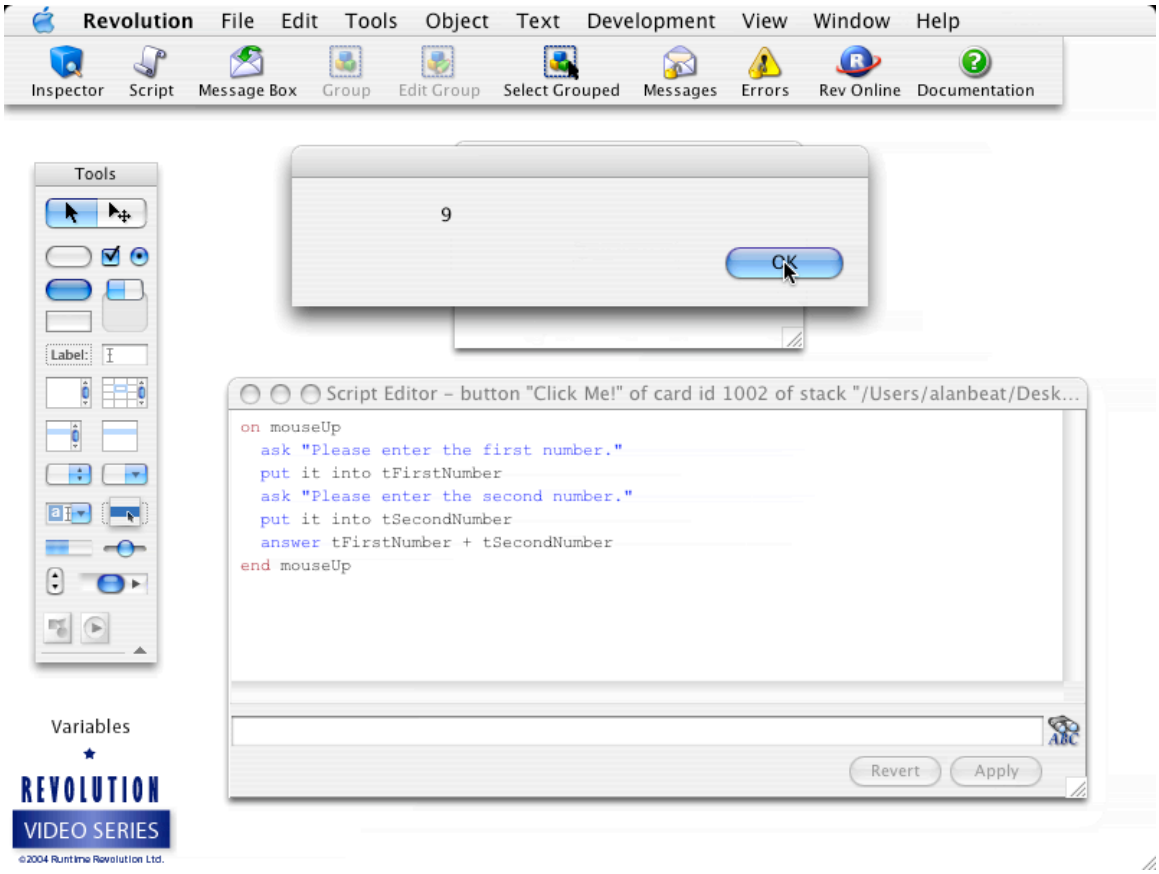
Now, instead of putting the two numbers together side by side with ampersand, we'll use the mathematical plus symbol to add them to each other.

```
answer tFirstNumber + tSecondNumber
```


Revolution File Edit Tools Object Text Development View Window Help
Inspector Script Message Box Group Edit Group Select Grouped Messages Errors Rev Online Documentation
Tools
Label:
Please enter the first number.
5
Cancel
OK
Script Editor – button "Click Me!" of card id 1002 of stack "/Users/alanbeat/Desk...
on mouseUp
ask "Please enter the first number."
put it into tFirstNumber
ask "Please enter the second number."
put it into tSecondNumber
answer tFirstNumber + tSecondNumber
end mouseUp
Revert
Apply
Variables
REVOLUTION
VIDEO SERIES
©2004 Runtime Revolution Ltd.


Revolution File Edit Tools Object Text Development View Window Help
Inspector Script Message Box Group Edit Group Select Grouped Messages Errors Rev Online Documentation
Tools
Label:
Please enter the second number.
4
Cancel
OK
Script Editor – button "Click Me!" of card id 1002 of stack "/Users/alanbeat/Desk...
on mouseUp
ask "Please enter the first number."
put it into tFirstNumber
ask "Please enter the second number."
put it into tSecondNumber
answer tFirstNumber + tSecondNumber
end mouseUp
Revert
Apply
Variables
REVOLUTION
VIDEO SERIES
©2004 Runtime Revolution Ltd.

And we have the answer, 9.



## Appendix: Scripts used in this tutorial

```
on mouseUp
  ask "What is your name?"
  answer it
end mouseUp
```

```
on mouseUp
  ask "What is your name?"
  put it into tUserName
  ask "What is your age?"
  put it into tUserAge
  answer "My name is " & tUserName & ".  My age is " & tUserAge
end mouseUp
```

```
on mouseUp
  ask "Please enter the first number."
  put it into tFirstNumber
  ask "Please enter the second number."
  put it into tSecondNumber
  answer tFirstNumber + tSecondNumber
end mouseUp
```