

[Lesson Map](#) | [Next > Echo the Input](#)

Ask for Input

Your first input-accepting program will ask the user for his or her name, then print that information (echo it) back to the mainwin. You know, from your "Hello, World!" program, that you can show text on a mainwin screen with a simple command:

```
print "Hello, World!"
```

To tell the user what to do for the new program, the command will read:

```
print "What is your name?"
```

The command to receive text is **input**. The **input** command can use text just like the **print** command, to tell the user what to input. That part of the command looks like this:

```
input "What is your name?"
```

Now you need a variable to store the input. You could name the variable just about anything, but for now, just call the variable "name\$". Why the dollar sign in the name? In programming, a line of text is called a **string**, because to the computer, the letters in the text look like a string of pearls, each letter connected to the next. The dollar sign looks like an "S", for "string", and is easy to spot in the program's source code and recognize as the mark for a variable that holds strings. Programmers use tricks like this to make their code readable and understandable.

Put the variable to hold the reply right after the instruction in the **input** command. Separate the instruction from the variable with a semicolon:

```
input "What is your name?"; name$
```

When it sees this command, the program will print "What is your name?" to the mainwin, then stop and wait for the user to type an answer and press the **Enter** key. The answer goes into the **name\$** variable. Now you've got something to work with!

[Lesson Map](#) | [Next > Echo the Input](#)