

[Lesson Map](#) | [Next > Types of Variables](#)

## Variables

In [Passing Notes](#) you learned that variables hold information, like an envelope holds a note. You can change the notes in a variable, but the variable itself, with its name, stays the same. That means you can write a program for handling variables, and the program will do its job on the information in the variables, no matter how the information changes. So, if your program wants to print the variable called FirstName\$, it will print a name regardless of what the FirstName\$ variable currently holds. Tom, Dick, Harry, Jane, Alyce, or Mary will each get printed if they have been put into FirstName\$.

Computer programs handle different types of information differently. When you add numbers, you get a new number, a total. When you add words together, you get combined phrase, with all the words still there, but now hooked together in a longer string. (That's why we call a piece of text a "string"!)

To help the program know whether to add up the contents of variables for a total, or instead paste them together one after another into a longer string, you tell the program what type of contents a particular variable can hold. Different types of variables hold different types of contents. A number-holding variable holds only numbers, and a string-holding variable holds only text.

Variable types have rules about how they are named. The rules require certain types of variables to have specific symbols in their names, so the program can tell what type of information a variable can hold just by seeing how the variable is named. These naming conventions also help you identify how variables are used in your programs, making it easier for you to come back six months after writing a program and still see what it does just by looking at the code.

The next lesson covers variable types, and how they are identified by their names.

[Lesson Map](#) | [Next > Types of Variables](#)