

## Variables

The goal of this lesson is to learn to use variables and constants. Variables allow you to store values in memory and reuse those values. You will be able to create (declare) variables, including how they must be named and selecting the correct data type. You will give a variable a new value using the assignment statement. A constant can be used almost like a variable, but you cannot assign a constant a new value.

### What is a variable?

A variable is a named location in memory. Variables allow you to store values in memory and reuse them later in the program.

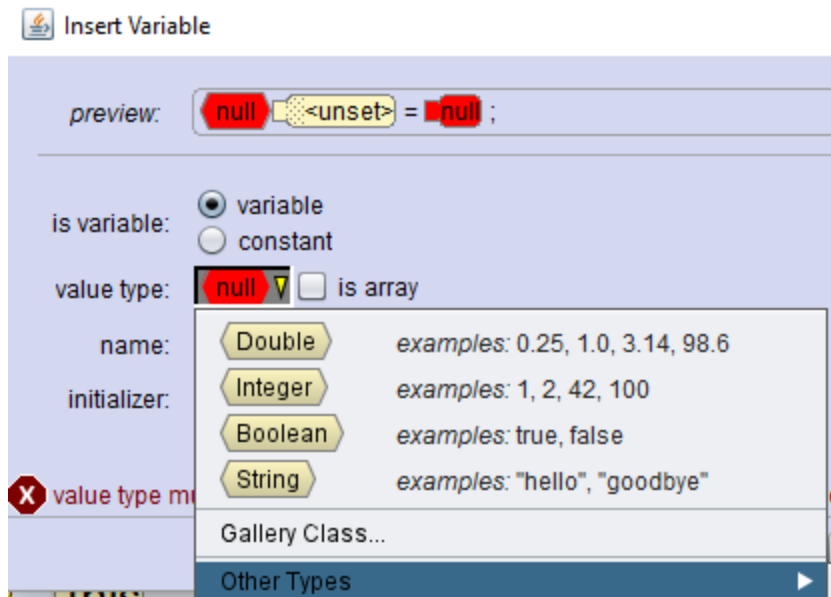
### View code as Java or Alice format

In an earlier lesson we demonstrated how to set up the preferences to view the code using the Java format. If you haven't already done so, please select from the menu: Window, Preferences, Programming language, Java. This is important so that your code will match the code shown in the lessons.

### Variable types

When you drag the variable tile to the code window, a pop up window will ask you to select the value or data type, a name and an initial value.

Alice has 4 primitive data types: Double (decimal numbers), Integer (whole numbers with no decimal places), Boolean (true or false values) and String (sentences and words). *(In later lessons you will learn how to use the other choices Gallery Class and other types.)*



You will select a type based on how you want to use the variable.



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Name of variables

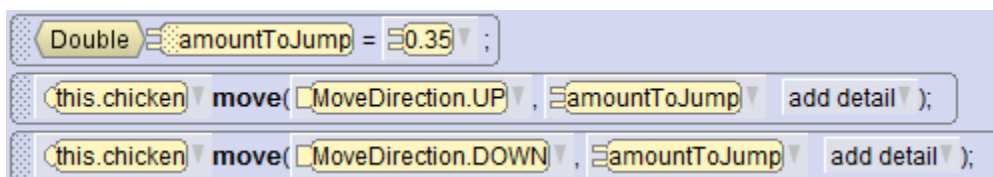
After deciding on the type, the next thing a programmer must do is decide on a variable name.

- A variable name must start with a letter of the alphabet or an underscore.
- After the first letter, there can be additional letters, digits and underscores.
- There cannot be any spaces in a variable name.
- Words that have a special meaning in Alice such as **this** should not be used to name variables.
- The name of the variable should indicate its purpose. Variable names like a, b and c are not very good names for variables unless you are using them for the three sides of a triangle.
- If two words are joined together (*you cannot have any spaces*), an upper case letter for the second word can improve the clarity. This style is often called "camel case." Example: **amountToTurn**, **distance**, and **distanceToTree** are all good variable names.
- It will be easier to remember the names if you develop a particular style and stick to it.

Adding a variable to the code, giving it a name and selecting the type and initial value is called **declaring** the variable.

## A jumping chicken

We will use a variable to make a chicken jump. Since we want the chicken to jump up and down by the same amount, a variable make that easier. The commands move, turn and roll all use a double value (a decimal number) as the argument, we will declare the variable to be type double. Using "camel case" we name the variable **amountToJump**. By substituting the variable amountToJump for the constant in the move commands, we can change the value in just one place instead of two places.



```
Double amountToJump = 0.35 ;  
this.chicken move( MoveDirection.UP , amountToJump add detail );  
this.chicken move( MoveDirection.DOWN , amountToJump add detail );
```

The code shown here makes the chicken jump up by 0.35 and back down by the same amount.

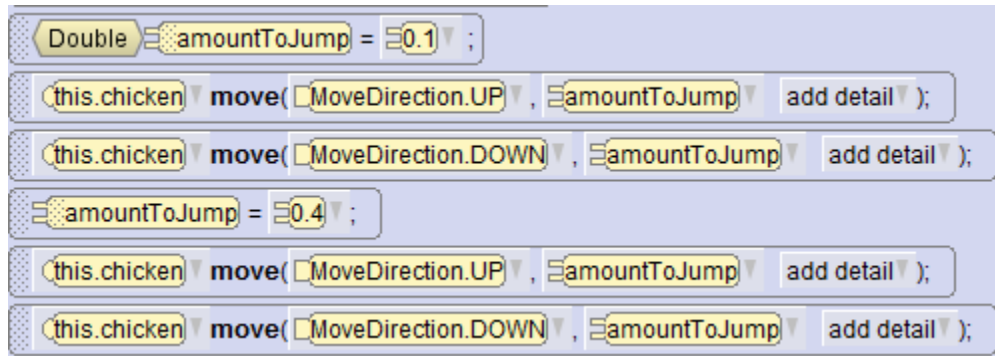


Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/) Creative Commons Attribution-NonCommercial 4.0 International License

## Assign

As the name suggests, a variable can vary, or change. We will use the assign command (statement) to give amountToJump a new value. In the code shown below, amountToJump is given an initial value of 0.1. The chicken jumps up 0.1 and back down by the same amount. Then we use the assign statement (amountToJump = 0.4; ) to give the variable a new value. We copied the two move statements to the clipboard and drag them into the code after the assignment statement:



```

Double amountToJump = 0.1 ;
this.chicken move( MoveDirection.UP , amountToJump add detail );
this.chicken move( MoveDirection.DOWN , amountToJump add detail );
amountToJump = 0.4 ;
this.chicken move( MoveDirection.UP , amountToJump add detail );
this.chicken move( MoveDirection.DOWN , amountToJump add detail );

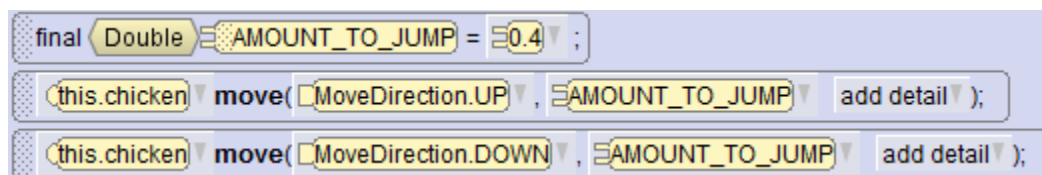
```

When we run the program, the chick jumps up and down a little bit, (0.1) and then up and down a lot (0.4)

## Constants

We could also write the jumping chicken program using a constant. The value of a constant cannot change. It is given an initial value and it will stay the same while the program is running. (Of course, we can change its value in the code editor.)

Constants are usually named using all uppercase letters. Use an underscore to separate two words. A constant will have the word **final** in front of the declaration. This indicates that it is a constant. The initial value is its final value. We cannot use an assign statement to change the value of a constant.



```

final Double AMOUNT_TO_JUMP = 0.4 ;
this.chicken move( MoveDirection.UP , AMOUNT_TO_JUMP add detail );
this.chicken move( MoveDirection.DOWN , AMOUNT_TO_JUMP add detail );

```

When we run the program, the chick jumps up and down by 0.4, the value of the constant **AMOUNT\_TO\_TURN**.



Author: Janet E. Joy; Publisher: Zebra0.com

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/) Creative Commons Attribution-NonCommercial 4.0 International License