

Unit II: Creating Code

Chapter 10: User Input in Alice 3

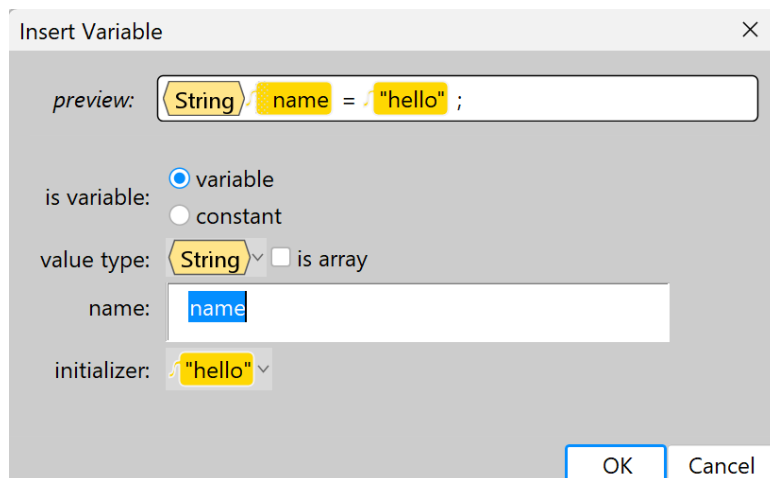
In this chapter, you will learn how to make an Alice program more interactive by getting input from the user and storing that information in variables. You will learn how to ask the user for information, such as a name or age, and save the responses so they can be used later in the program. By combining user input with variables, you can create programs that respond to individual users and produce different actions or results based on the information provided.

Objectives

- Select and use functions to ask the user a question and store the result.
- Use functions to get information from the user.
- Use stored values in the code.
- Use input from users in the code.

Declare a Variable

We are going to ask the user their name, then say hello to them with their name. The first step is to declare a string to hold the name. We must put in an initial value, so just select a default value to start. This is just a placeholder.



Functions

On the left in code view, there are two panels: Procedures and Functions. Procedures perform an action, functions return a value. Select the Functions panel. The functions are divided into categories: editable functions, appearance, size, prompt use, and other. Under prompt use there are 4 functions corresponding to the basic types.

There are four functions to get values from the user corresponding to the primitive types:

- **Double:** `getDoubleFromUser`
- **Integer:** `getIntegerFromUser`
- **Boolean:** `getBooleanFromUser`
- **String:** `getStringFromUser`

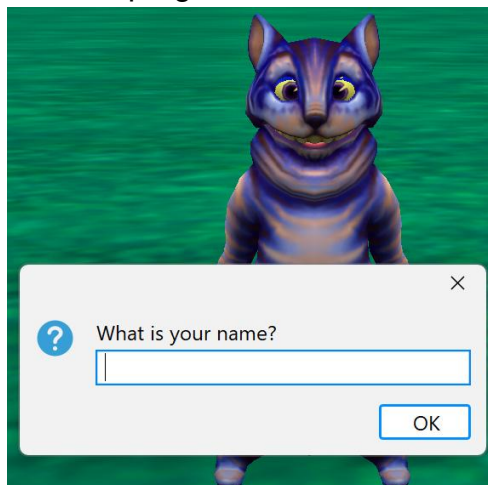
Each of these functions require one argument, the prompt. Because functions return a value, you must use the function as the value for the variable. In the Functions panel, select **getStringFromUser** and drag it and drop it to replace the string “**hello**”.

```
void myFirstMethod( )
do in order
    //User Input for Interactivity
    //Programmer: Janet Joy
    String name = "hello" ;
```

When you drop the function to replace the word “hello” you will be asked to type in a value. This is the question the user will see. Type “What is your name?” as the prompt. Now the code looks like this:

```
void myFirstMethod( )
do in order
    //User Input for Interactivity
    //Programmer: Janet Joy
    String name = this.getStringFromUser( "What is your name?" ) ;
```

Run the program:



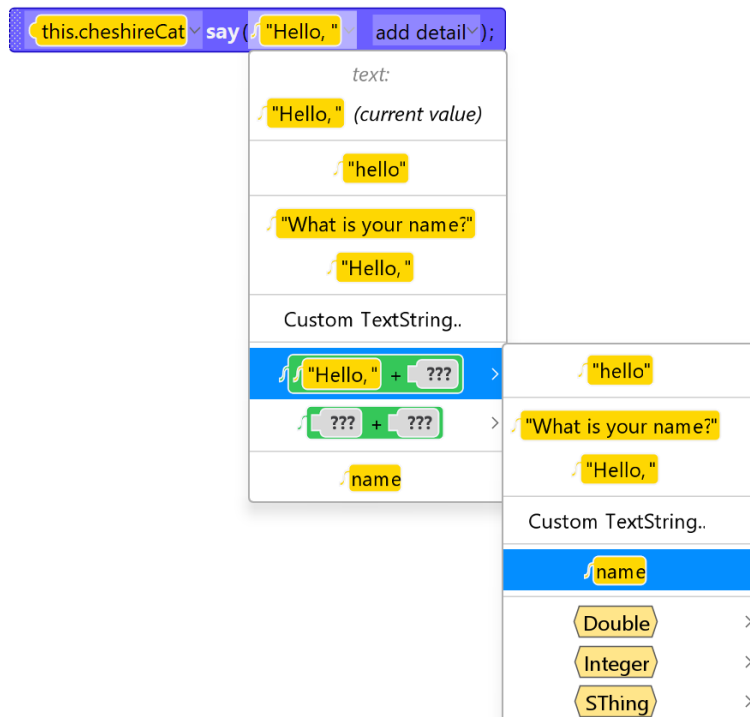
Using the Value

After you type your name and press OK, the program stores whatever you typed as the value for name. We can use these variables the same way we would any other variable. If the user types in “Charlie”, we would like the cat to say “Hello, Charlie.”

Start with the say command (on the Procedures tab) and the first part of the sentence including the space.

`this.cheshireCat say ("Hello, " add detail);`

Next, click the down arrow right after the word. Select “Hello, “+ ???, then select name from the next box.



The code is now:

```
void myFirstMethod ( )
do in order
    //User Input for Interactivity
    //Programmer: Janet Joy
    String name = this.getStringFromUser ("What is your name?");
    this.cheshireCat say ("Hello, " + name add detail);
```

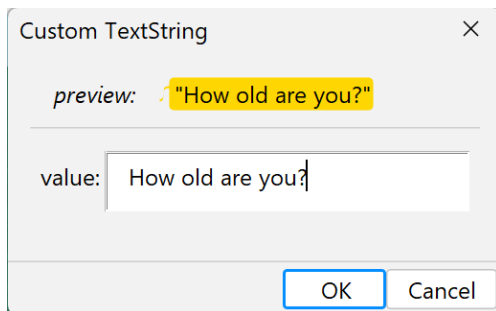
Run the program and enter your name. The cat now says “Hello, “ plus your name. Make sure that there is a space between words.: “Hello, Charlie” not “Hello,Charlie”

Integer

Declare an integer variable **age**, with a placeholder value of 3.

```
Integer age = 3 ;
```

Drag the Function **getIntegerFromUser** to replace the 3. Type in “**How old are you?**” as the prompt.



The code is now:

```
Integer age = this.getIntegerFromUser("How old are you?") ;
```

We can now have the cat say something like “Wow, 23! That’s old.” The sentence now has 3 parts: “**Wow**, “, the variable **age**, and “**! That’s old.**” We put it together the same way.

Step 1: `this.cheshireCat say ("Wow, " add detail);`

Step 2: `this.cheshireCat say ("Wow, " + age add detail);`

Step 3: `this.cheshireCat say ("Wow, " + age + "! That's old!" add detail);`

Make sure all the spacing between words is there, then add any of the other arguments to the say command.

Double

We will ask the user how much the cat should turn, a **Double**.

Step 1: Declare variable with a placeholder value: `Double turnAmount = 1.0 ;`

Step 2: Add the function `getDoubleFromUser` with a prompt:

```
Double turnAmount = this.getDoubleFromUser("How much should I turn?");
```

Step 3: Use the value just as you would any other variable:

```
this.cheshireCat.turn(TurnDirection.RIGHT, turnAmount, add detail);
```

The cat turns the amount the user enters.

Boolean

A **Boolean** can only have values of **true** or **false**. You will learn more about using Boolean in another lesson. For now, we will ask the user if he likes pizza. No matter what answer the user selects, we will reply "Me too!"

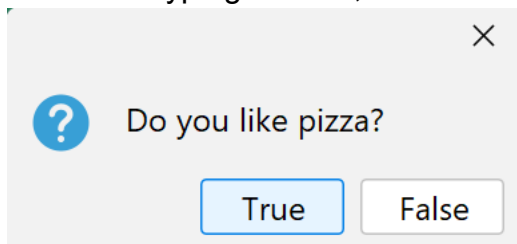
Step 1: Declare the variable with a placeholder value:

```
Boolean pizza = true;
```

Step 2: Drag the **getBooleanFromUser** function to replace the value.

```
Boolean pizza = this.getBooleanFromUser("Do you like pizza?");
```

Instead of typing a value, the user selects **true** or **false**.



The Code

```
public void myFirstMethod() {
    // User Input for Interactivity
    // Programmer : Janet Joy
    String name = this.getStringFromUser("What is your name?");
    this.cheshireCat.say("Hello, " + name);
    Integer age = this.getIntegerFromUser("How old are you?");
    this.cheshireCat.say("Wow, " + age + "! That's old!");
    Double turnAmount = this.getDoubleFromUser("How much should I turn?");
    this.cheshireCat.turn(TurnDirection.RIGHT, turnAmount);
    Boolean pizza = this.getBooleanFromUser("Do you like pizza?");
    this.cheshireCat.say("Me too!");
}
```