# Working with Strings and Text in Alice

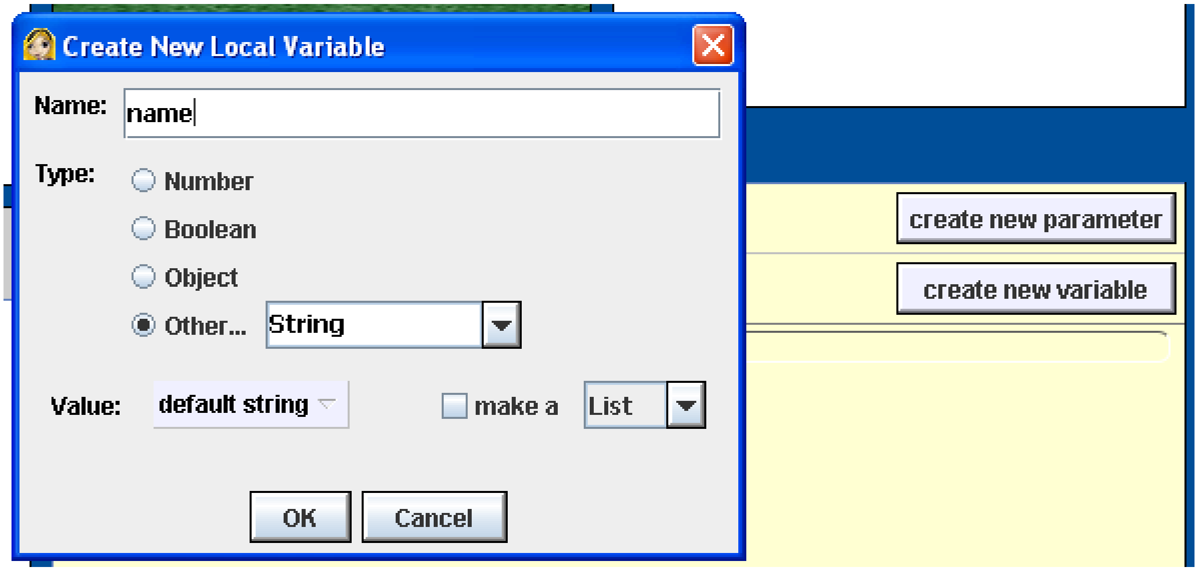
A string is a **sequence of characters**, while a character is (in general) any of the letters, numbers, or symbols represented on your keyboard. In computer programs, strings are used to represent very common and useful information such as names, addresses, messages, and many others. Alice provides various features for dealing with strings.

In this tutorial, we will consider the following:

1. Variables to hold strings
2. Asking the user to enter a string
3. Joining two or more strings together
4. Converting a non-string to a string

## 1. Variables to Hold Strings

So far, we have been creating **world variables**, and will continue this practise as we learn about strings. To create a world variable, use the "create new variable" button at the top of the **world.my first method** program window, as shown.



Create New Local Variable

Name: name

Type: ○ Number ○ Boolean ○ Object ● Other... String

Value: default string

□ make a List

# Working with Strings and Text in Alice

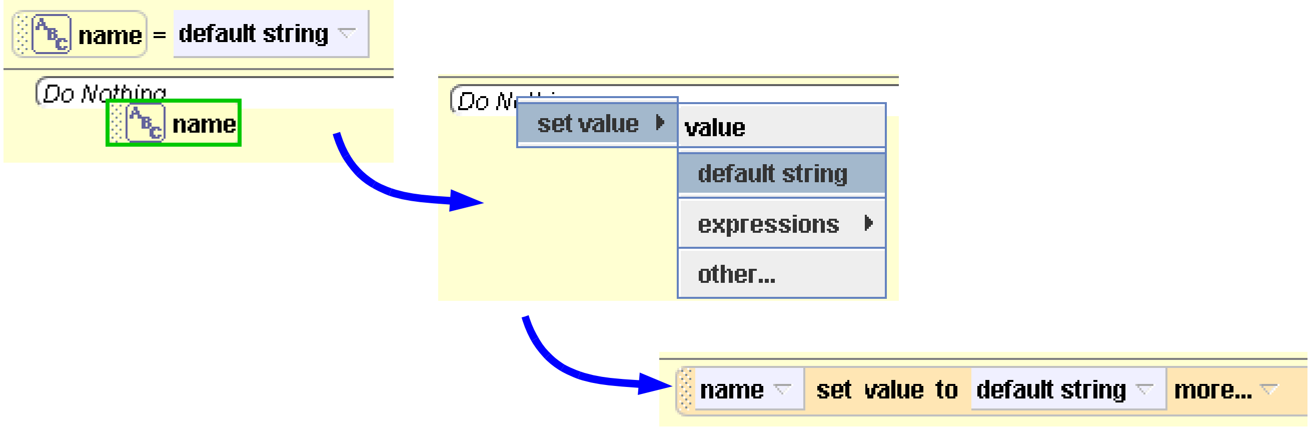
Create a new world using the *grass* template. Add the an instance of the *plato* object from the galleries. Position Plato in the middle of the scene and have him face the camera. Rather than trying to move the camera around, you could **right click on plato** and use the **primitive method** called **plato turn to face camera**.

Create a new variable called "name" of type **string**.

## 2. Asking the User to Enter a String

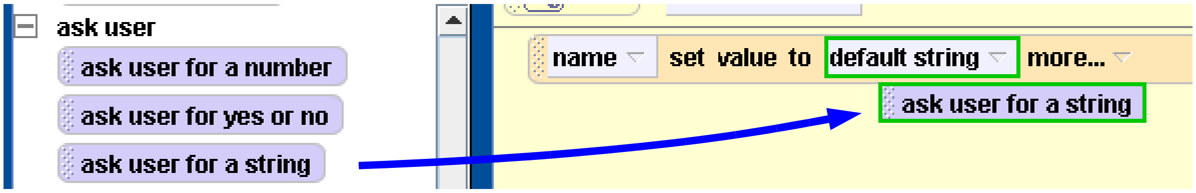
In a previous tutorial, we asked the user to enter a **number**. Now we will ask the user to enter a **string**, which is a very similar process. There is a **world function** called **ask user for a string**, which prompts the user with a question, and the user's response is returned as a string. We will **store this string in the variable called name**.

Drag the variable **name** into the programming window, where we will **set value to** *default string* for now.



```
name set value to default string more...
```

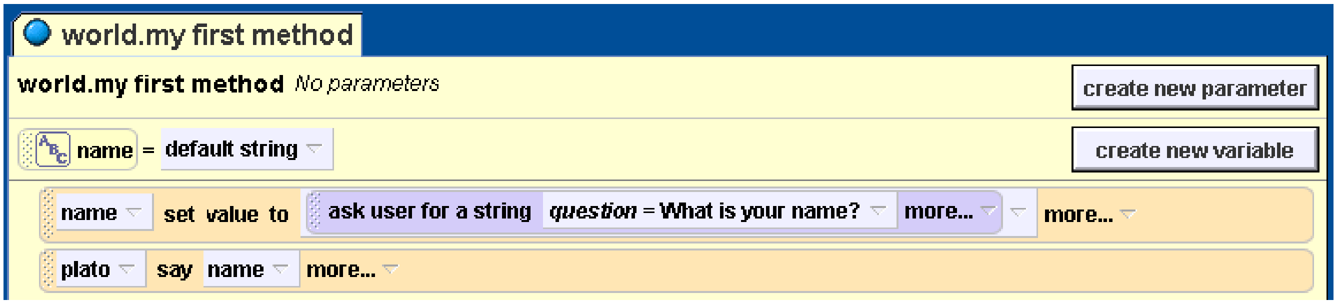
Instead of the *default string*, we want to have **name** set by the user's input. Find the **world function** called **ask user for a string** and drag it into the programming window to replace *default string*. Change the question to "What is your name?".



```
name set value to default string more...
```

Now have Plato repeat the name back to the user using plato's **say method**.

## Working with Strings and Text in Alice



```
world.my first method No parameters
name = default string
name set value to ask user for a string question = What is your name? more... more...
plato say name more...
```

### 3. Joining Two Strings Together

It is often useful to join strings together, which is accomplished using the **world function** called **a joined with b**, where *a* and *b* are the strings to be joined together.

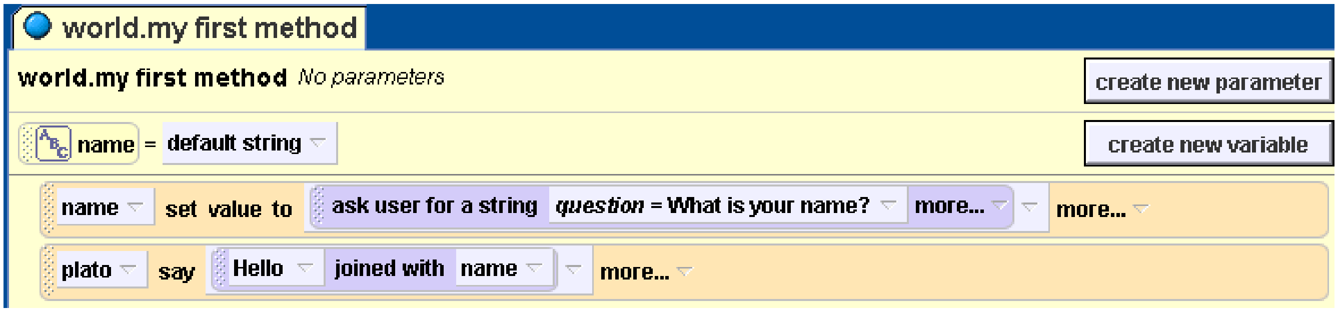
The string output in our current world could use some improvement. After asking the user their name, it simply says the name back to the user. Perhaps a “Hello” followed by the name might be a bit more useful.

Drag the **a joined with b** function to replace the **name** variable in the “plato say ...” method.

Set the first string (a) to be “Hello”, and set the second string (b) to be the **name** variable. Run the world.

You may have noticed that it doesn't look quite right. Most programming languages do not make assumptions about things like punctuation or spacing with strings. If you want them, you need to provide them yourself.

Modify the first string (a) to be “Hello “, making sure you include a space at the end. Now run the program and it should look more like what you expect.



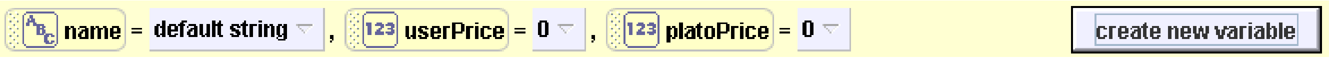
```
world.my first method No parameters
name = default string
name set value to ask user for a string question = What is your name? more... more...
plato say Hello joined with name more...
```

## 4. Converting a Non-String to a String

It is often necessary to display information that is not a string (e.g., numbers). Although many languages allow this automatically, Alice requires all displayed information to be a string. There are two **primitive methods** for displaying information: **think** and **say**. Both of these methods only accept **string arguments** as their **parameters**. To convert any other type of variable to a string, use the **<u>what</u> as a string** world function.

Let us modify our world so that plato also performs a (number) calculation, and then tells us the answer. We will need to convert his numeric answer into a string.

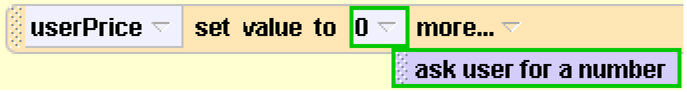
Create two new **world variables**, both of which will be **numbers**. Call the first one "userPrice" and the second one "platoPrice". We are going to use these to do some haggling with Plato. **Initialize** the new variables to zero.



```
name = default string , userPrice = 0 , platoPrice = 0
```

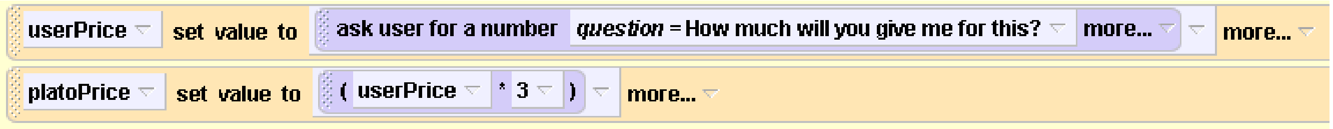
You might have noticed that Plato is holding an object in his hand. Let's assume he wants to sell us this object. He will ask the user their price (that they want to pay), and then he will calculate his own price, which is three times higher, which he will tell the user.

Drag the **userPrice** variable into the programming window and **set value to** the world function **ask user for a number**. Since Plato seems to be holding out the object in his hand, have him ask, "How much will you give me for this?"



```
userPrice set value to 0 more...
```

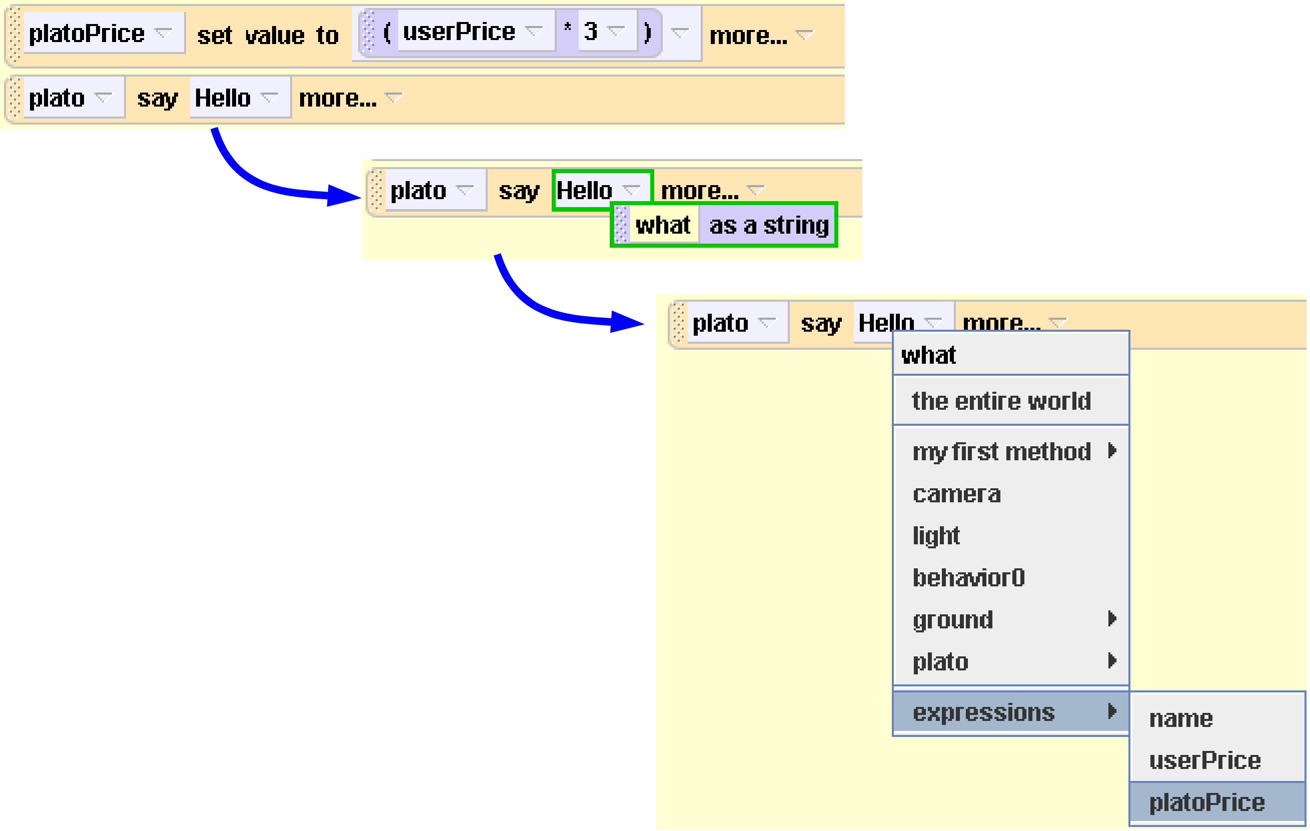
Plato is a bit greedy, so he thinks the object in his hand is worth three times as much as you do. Set the value of the **platoPrice** variable to be three times as much as **userPrice** using a **mathematical expression**.



```
userPrice set value to ask user for a number question = How much will you give me for this? more... more...
platoPrice set value to ( userPrice * 3 ) more...
```

# Working with Strings and Text in Alice

Finally, have Plato tell you his price using the plato method **say**. Since you cannot display a number in Alice, you will need to convert it to a string using the **what as a string** world function.

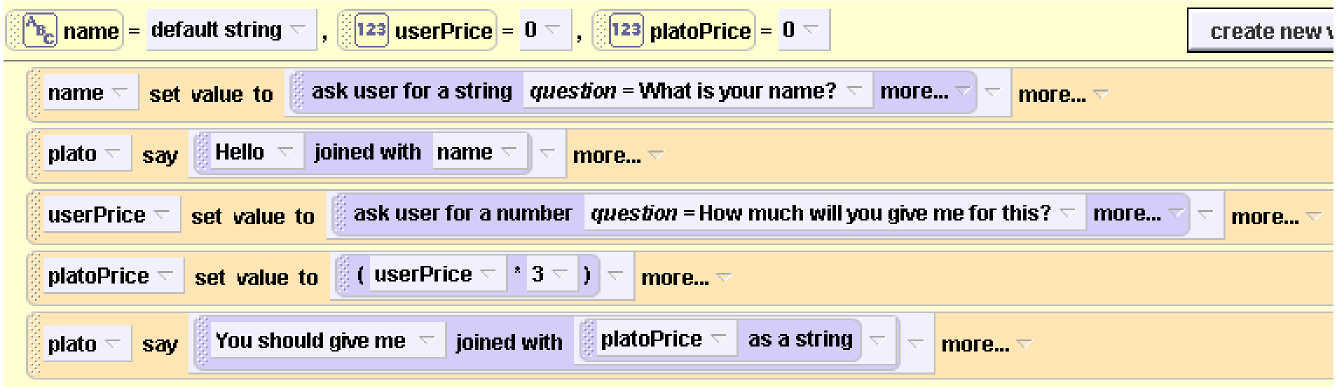


```
platoPrice set value to ( userPrice * 3 ) more...
plato say Hello more...
```

```
plato say Hello more...
          what as a string
```

```
plato say Hello more...
```

Plato isn't communicating this new price very well. See if you can improve this by using the world function called **a joined with b**. You final program should look very similar to this:



```
name = default string , userPrice = 0 , platoPrice = 0          create new v
  name set value to ask user for a string question = What is your name? more... more...
  plato say Hello joined with name more...
  userPrice set value to ask user for a number question = How much will you give me for this? more... more...
  platoPrice set value to ( userPrice * 3 ) more...
  plato say You should give me joined with platoPrice as a string more...
```