How Tall Are You?
Introducing Functions

By Jenna Hayes
under the direction of Professor Susan Rodger
Duke University
July 2008
Updates made June 2014 by Yossra Hamid

In this tutorial you will be learning to use functions to ask how tall a character is. Using this information two characters will compare their height and give a specific response depending on who is the tallest.

A function in Alice is basically a question about information in your Alice world that Alice answers. You can ask how tall or wide an object is, how far away it is from another object, and many other questions.

Step 1: Getting Started

Click on the green Add Objects button. Go into your Local Gallery and find the People folder. Then, add the RandomGuy2 object to your world. Now go back up to the Local Gallery, and find the Animals folder. Add a penguin to your world.

Using your object moving buttons, move the person to the left of your viewing screen, and move the penguin to the right of your viewing screen. Your screen should look like this when you are finished.

Update: RandomGuy2 is no longer an object in newer versions of Alice. Use any other object in the People folder (for example, “Josh”).
### Step 1: Getting Started

Right click on `penguin` in the object tree, and go to methods. Then select `penguin resize`, and then `other...`. On the calculator that pops up, type `1.75`. Your penguin will get larger.

Now, right click on `randomGuy2` in the object tree, and select methods. Then find his `turn to face` method. Click on that, and then select `penguin`, and the entire `penguin`. Then, right click on `penguin` in the object tree, and select methods. Go to `penguin turn to face`, and then select `randomGuy2`, and the entire `randomGuy2`. Your characters should now be facing each other.

### Step 2: Understanding Functions

Click on the `penguin` object in the object tree. Then click on the `functions` tab. You will see a LONG list of functions. Scroll down and look at the functions under `size`.

Each of these functions asks a question about the penguin, and then keeps the answer so that you can use it in your Alice world.

- How wide is the penguin?
- Is the penguin larger than a certain object?
- Is the penguin taller than a certain object?

These functions can be very useful in Alice. What if, for example, you want to make something move up and stand on top of the penguin’s head? You don’t know how exactly how tall the penguin is. But Alice does!

### Step 3: Using the True or False Functions

Some functions in Alice are statements to which the answer is either `true` or `false`, like the one on the previous page that says, “penguin is taller than.” We want to know whether the penguin is taller than the `randomGuy2`. It’s almost impossible to tell just by looking at them, because their heights are so close together. So we will use a function to know for sure.

We are going to add commands so that when the Alice world starts, if the penguin is taller, it will say “Hah! I’m taller!”, and do a victory spin, but if the `randomGuy2` is taller, he will say, “Hah! I’m taller!”, and do a victory spin.
Since we only want the penguin to say “Hah! I’m taller” if he is taller, we need to use something called an If Else statement. You will find it located below your method editor:

First drag a Do in order into your method editor. Then, inside that, drag in an If Else statement. Select true when you drop it. We will replace this later.

So let’s construct our question.

Click on penguin in the object tree and then click on functions. Find the penguin is taller than function, and drag it over the word true in your If Else statement. When the little menu pops up, click on randomGuy2.

Here is where you put the question that is either true or false. For us, that will be penguin is taller than randomGuy2. Since it is currently set at true, that means this If Else says, “If this statement right here is true, do whatever commands are right under it.”

This is where you put whatever you want to happen if the answer to your question is true. This is where we’ll tell the penguin to say “Hah! I’m taller!”

This is where you put whatever you want to happen if the answer to your question is NOT true. If it is not true, it will skip everything above the Else, and go straight to whatever is here. This is where we’ll tell the person to say “Hah! I’m taller!” because if our statement is false, and the penguin is NOT taller, that means the person is taller!

Get a Do together from the bottom of your method editor and drag and drop it into the If/Else right below If. Then, click on penguin in the object tree, and go to his methods list. Drag and drop penguin turn into the Do together. When you drop it, select right, and then 1 revolution. Your code will look like this.
Then, find **penguin** say, and drag it into the **Do together**. When the menu pops up, click on **other**, and type in “Hah! I’m taller!”. Drag and drop another **Do together** into the **If/Else**, this time right under the **Else**. Click on **randomGuy2** in the object tree and look at his methods list. Find **randomGuy2 turn**, and then drag and drop that into the newest **Do together**. When you drop it, select **right**, and then 1 revolution. Your code will look like this.

Next, find **randomGuy2** say and put it in the **Do together**. Tell him to say “Hah! I’m taller!”

It seems that the penguin is taller. To test and make sure your function is working correctly, go to the Add Objects screen and use your object buttons to resize the **randomGuy2** and make him clearly taller than the penguin. Now, play your world again. This time, the guy will say that he is taller! You can change him back to his normal size after you test this out.

Now play your world and see who is taller!
Step 4: Using Number Functions

Now we’re going to use one of the functions that is a question whose answer is a number. We’re going to make the penguin move right up to the man and give him a hug. The only problem is, we don’t know how far to tell him to move! That’s why we’ll use a function!

First we’ll tell the man to say something. He’s tired of competing with the penguin about their height, and he wants to be friends.

Click on `randomGuy2` in the object tree, then click on methods. Find `randomGuy2 say`, and drag it under your If Else statement.

Make him say “Let’s be friends. Give me a hug, penguin!”

Step 4: Number Functions Continued...

Now we’re going to use a function to tell the penguin how far forward to move. Click on `penguin` in your object tree, and then click on the functions tab. Find the function called `distance to`.

Drag that function over the 1 meter on your `penguin move forward` command and drop it there. On the menu that pops up, select `randomGuy2`.

Your code for this line will then look like this:
The last thing you need to do is make the penguin hug the guy.

Click on `penguin` in the object tree, and then click on the `methods` tab. Click on `create new method`, and call it `hug`. In this method, experiment until you find code that makes the penguin hug the guy. My code for `penguin.hug` looks like this.

```plaintext
penguin.hug No parameters
No variables
```

```
Do together
penguin.rightWing = turn = backward = 0.25 revolutions = more...
penguin.leftWing = turn = backward = 0.25 revolutions = more...
```

Now click on the `world.my first method` tab of your method editor. Make sure you have clicked on `penguin` in the object tree, and then look at the penguin’s `methods` list. Find `penguin.hug`, and drag it into the bottom of your method editor. Your final code will look like this.

```
Do in order
penguin = in taller than = randomGuy2 = more...

Do together
penguin = turn = right = 1 resolution = more...
penguin = say = Hello! I’m taller! = more...

Else
randomGuy2 = turn = right = 1 resolution = more...
randomGuy2 = say = Hello! I’m taller! = more...

randomGuy2 = say = Let’s be friends. Give me a hug, penguin! = more...
penguin = move = forward = { penguin = distance to = randomGuy2 = 0.5 } = more...
penguin = hug
```

```
Step 5: Finishing Up Continued...
```

Play your world, and see the plot unfold! Now you know the basics of using functions. We only used two functions in this tutorial, but there are MANY functions in the Alice world that can be useful. Try some more functions on your own and explore how they work!