Using Functions in Alice
Step 1: Understanding Functions

1. Download the starting world that goes along with this tutorial. We will be using functions. A function in Alice is basically a question about information in your Alice world that Alice answers. 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.

2. 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 2: 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 normalGuy. 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!”, but if the normalGuy is taller, he will say, “Hah! I’m taller!”.
Step 2: cont.

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. It is located below your method editor:

First drag a **Do in order** into your method editor. Then, inside that, drag in an **If Else** statement.
Here is where you put the statement that is either true or false. For us, that will be penguin is taller than normalGuy. 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 your statement 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 your statement 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!
Step 2: cont.

So let’s construct our command. 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 normalGuy. Your code will look like this:
Step 2: cont.

Now click on `penguin` in the object tree, and then click on `methods`. Find `penguin say`, and drag it into the method editor under `If`. When the menu pops up, click on `other`, and type in “Hah! I’m taller!”. Your code will look like this:
Step 2:

Now click on `normalGuy` in the object tree, and click on methods. Find `normalGuy say` and put under the `Else` part of your If Else statement. Tell him to say “Hah! I’m taller!” Your code will look like this:

```
<table>
<thead>
<tr>
<th>If</th>
<th>penguin \ is taller than \ normalGuy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>penguin \ say \ Hah! I’m taller! \ more...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Else</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>normalGuy \ say \ Hah! I’m taller! \ more...</td>
</tr>
</tbody>
</table>
```

Now play your world. Who is taller, the penguin or the person?
Step 3: 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 normalGuy in the object tree, then click on methods. Find normalGuy say, and drag it under your If Else statement. Make him say “Let’s be friends. Give me a hug, penguin!” Your code will look like this:
Step 3: cont.

Now we want the penguin to move right up to the guy to hug him. Click on **penguin** in your object tree and then click on **methods**. Find the **penguin move** and drag it into your method editor under everything else. We don’t know exactly how far yet, so just put 1 meter for now. Your code will look like this:

![Code snippet](image)

Play your world just to see what it looks like.
Step 3: cont.

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 in front of.

Drag that function over the 1 meter on your penguin move forward command and drop it there. On the menu that pops up, select normalGuy. Your code will look like this:
Step 4: Finishing Up

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. Find the hug method and drag it into your method editor under everything else. Your final code will look like this:

Now play your world and watch the magic!
Now you know the basic of using functions. We only used two functions in this tutorial, but there are MANY functions in the Alice world that can be useful. Feel free to explore the uses of other functions.