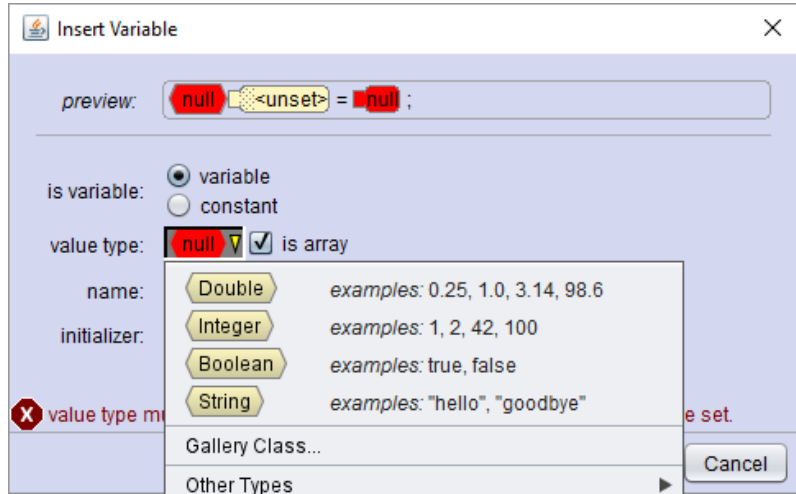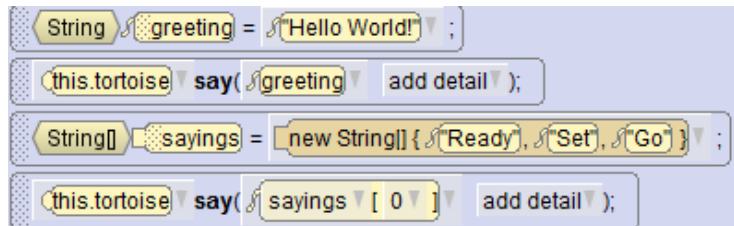# Arrays

An array is a list. When you declare an array in Alice, you can make it an array by selecting the check box next to the type. You can make an array of any types shown, or select Gallery Class, and then select one of the types of classes such as biped, prop, or swimmer.


Compare the two variables in this illustration. The variable **greeting** is not an array, it is a scalar. It has a single value, "Hello World". When the tortoise says greeting, he says "Hello World"

The variable **sayings** is an array. The items in an array are numbered starting with 0. The number is called the index or subscript. When you use an array you need to use a subscript to specify which item, or element, in the list is to be used. In this example, the tortoise says sayings[0], of "Ready".

	greeting
	Hello World

	sayings
0	Ready
1	Set
2	Go

An array can also use variables as the subscript:
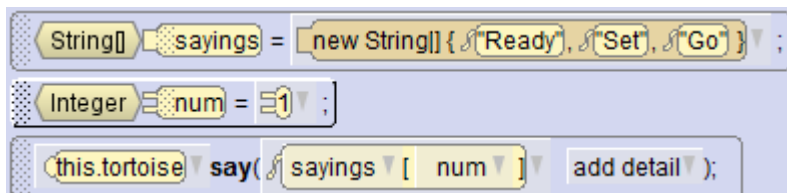
In this example, num is 1, so when the tortoise says **sayings[num]** he says "Set".

It is very useful to be able to use a variable as the subscript because we can use a loop such as the one shown below: The loop changes the value of the variable i. The values in the loop are 0, 1, and 2. When the tortoise says sayings[i] he says "Ready", "Set", then "Go".

This may not seem to be much of a savings, as we could have used just 3 statements and had the tortoise say "Ready" "Set", "Go". In the next example the tortoise says the names of the months.

First, we add the variable months. We select array, then click add to add the months one at a time.

Notice that "January" has an index, or subscript, of 0. "December" will be 11.

To say the months we can use a count loop like above or use "For each" as shown below:

Notice that the **for each** control structure uses a scalar variable month. This variable is assigned each value in the array months as the loop iterates.
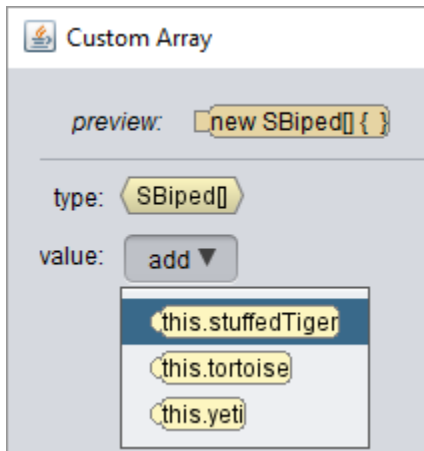
Arrays of Objects

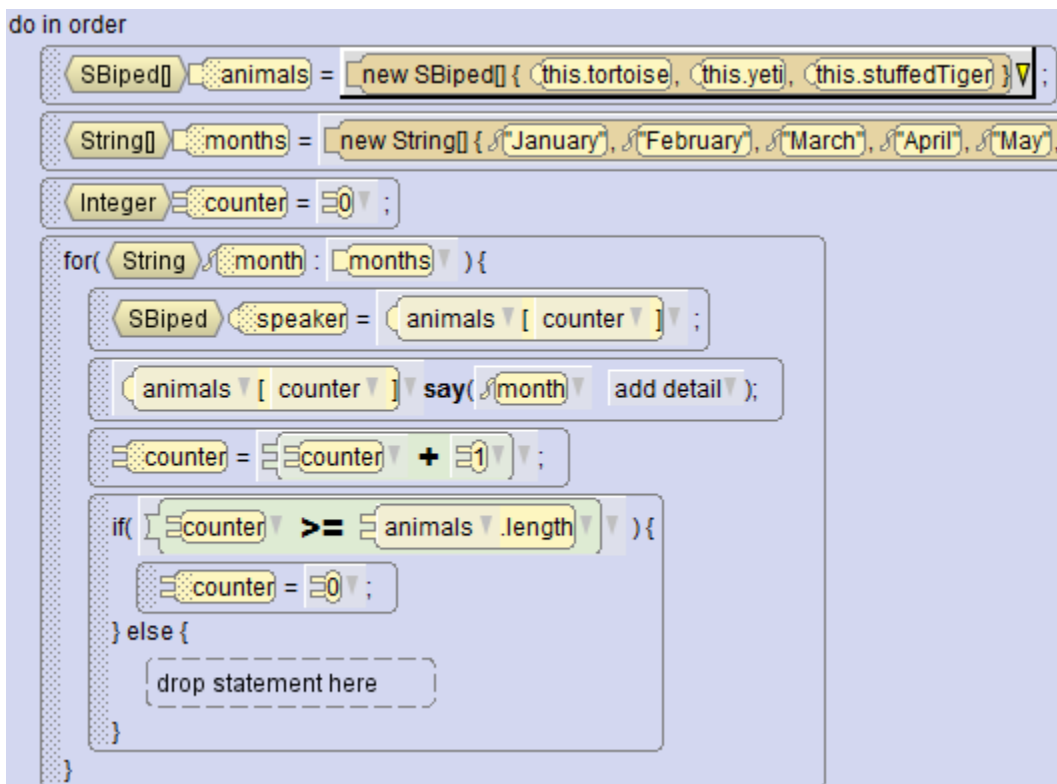In this example, we select Gallery Class and then select Biped. We can add each of the actors in the scene to the array.

You can see that we have created an array of animals, as well as an array of the months.

Can you figure out what this program does? Study the code and see if you can guess before going to the next page. Use scrap paper to write down the values of the variable and trace through the code like the computer will do when you run it.

Here is a trace of the program:

counter   month   output

0   January   tortoise says "January"

1   February   yeti says "February"

2   March   stuffedTiger says "March"

           (when counter becomes 3, it is set back to 0)

0   April   tortoise says "April"

1   May   yeti says "May"

2   June   stuffedTiger says "June"

           (when counter becomes 3, it is set back to 0)

0   July   tortoise says "July"

1   August   yeti says "August"

2   September   stuffedTiger says "September"

           (when counter becomes 3, it is set back to 0)

0   October   tortoise says "October"

1   November   yeti says "November"

2   December   stuffedTiger says "December"

Or in other words, the animals take turns saying the months.

This is a fairly long example, but it gives you an idea how powerful arrays can be.

You can download this movie and watch the videos at http://zebra0.com/alice/arrays/index.php

Parallel Arrays

Sometimes you want to have two or more arrays with matching information. For example, you might have one array with animals and a second array with the sounds the animals make. So if you have a cow in animals[0] then you would have the word "moo" in sounds[0]. These two arrays are called parallel arrays because they are the same size and each item in one array has a corresponding element in the other array.

Farm Animals

Here we have an Alice project with several farm animals: a cow, pig, billyGoat and a chicken. The highest class that they are all member of that has a "say" procedure is the SJointedModel. We have created an array of the animals and a parallel array of the things they say. Notice that cow is animals[0] and "moo" is sayings[0].

```
public void myFirstMethod() {
    SJointedModel[] animals = new SJointedModel[] {
        this.cow, this.pig, this.billyGoat, this.chicken
    }
    ;
    String[] sayings = new String[] {
        "moo", "oink", "maa", "cluck"
    }
    ;
    for( Integer indexA = 0; indexA<animals.length; indexA++ ) {
        SJointedModel animal = animals[i];
        animal.say( sayings[i] );
    }
}
```

We use a for loop to have each animal say the corresponding sound.

Who is the tallest?

Sometimes we want to find the largest or smallest value in an array. In this example, we make an arry of the 4 men. We will start by assuming that the fist one is the tallest. Then we compare the height of each person to the height of the tallest. If a person is taller, then he becomes the new value for tallest.
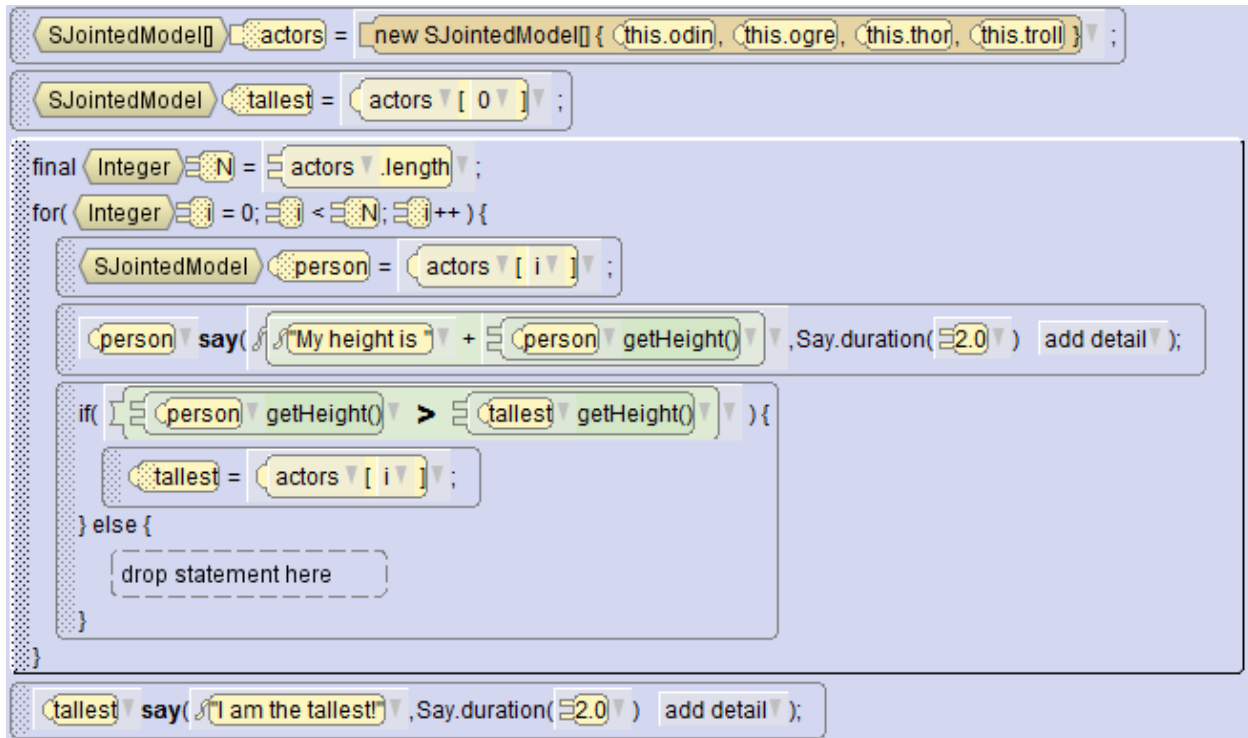
```
public void myFirstMethod() {
    SJointedModel[] actors = new SJointedModel[] {
        this.odin, this.ogre, this.thor, this.troll
    }
    ;
    SJointedModel tallest = actors[0];
    for( Integer indexA = 0; indexA<actors.length; indexA++ ) {
        SJointedModel person = actors[i];
        person.say( "My height is "+person.getHeight(), Say.duration( 2.0 ) )
;
        if( person.getHeight()>tallest.getHeight() ) {
            tallest = actors[i];
        } else {
        }
    }
    tallest.say( "I am the tallest!", Say.duration( 2.0 ) );
}
```

You can download this movie and watch the videos at http://zebra0.com/alice/arrays/index.php

Experiment: Can you also find the shortest person?