Boolean Expressions

Sometimes a programmer would like one statement, or group of statements to execute only if certain conditions are true. There may be a different statement, or group of statements that are to be executed when the condition is false. An expression that can be evaluated to true or false is called a Boolean expression. *(Named after a mathematician named Boole.)*

The Boolean operators

The Boolean operators are a bit different than the ones used in algebra because we have to be able to type them.

- `==` equal
- `<` less than
- `>` greater than
- `>=` greater than or equal to
- `<=` less than or equal to
- `!=` not equal

Boolean Expressions

(`x`, `y`, and `z` are integer variables with the values shown below)

```
x = 0; y=5; z=8;
```

Each expression below is **TRUE**:

- `x < y` 0 is less than 5
- `z > y` 8 is greater than 5
- `y < z` 5 is less than 8
- `y <= z` 5 is less than or equal to 8
- `x != z` 0 is not equal to 8
- `z >= y` 8 is greater than or equal to 5

Continuing with the same values for x, y, and z, each expression below is **FALSE**:

- `x > y` 0 is NOT greater than 5
- `z < y` 8 is NOT less than 5
- `y == z` 5 is NOT equal to 8
- `y <= z` 5 is less than or equal to 8
- `y != y` 8 is not equal to 8 is false
- `x != x` 0 is not equal to 0 is also false

Compound Boolean Expressions

A Boolean expression can also test multiple conditions. Multiple conditions use either AND && or OR ||. The && (and) means that both comparisons must be true. The || (or) means that at least one of them is true.

(**x**, **y**, and **z** are integer variables with the values shown below)

```
x = 0; y=5; z=8;
```

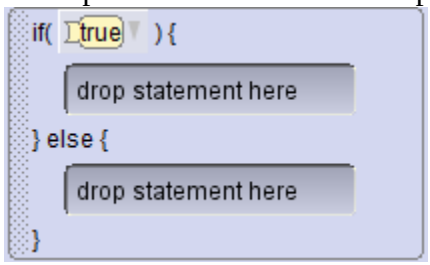
Each expression below is **TRUE**:

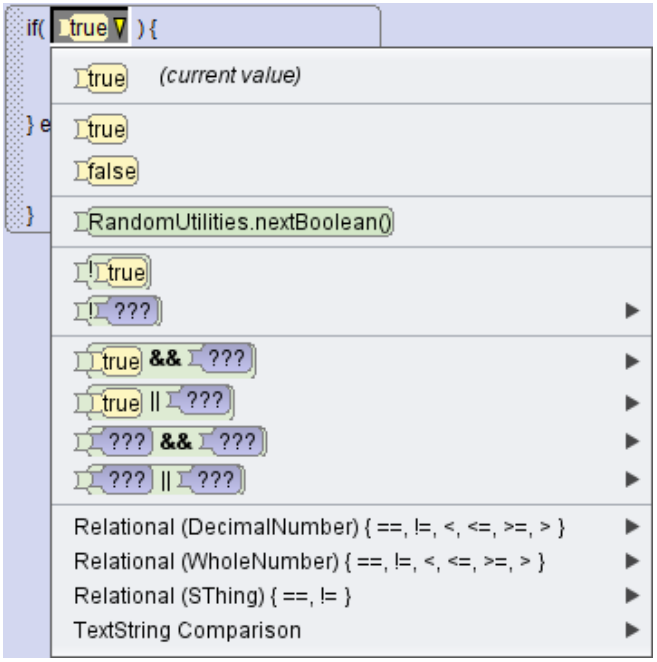
`x < y && y < z`	0 is less than 5 AND y is less than z.
`z < x || y < z`	z is not less than x, but y is less than z
`x < z && y < z`	both the first part AND the second part are true
`y == z || x < y`	the first part is false, but the second part is true
`x != z || y != z`	both parts are true, so this is true.
`z >= x && z >= y`	both parts are true, so this is true.

In Alice, you will use a Boolean expression in **if** and **while** blocks.

If Block

When you drag the if block into your code area, you must select either true or false as the value. This is usually just a place holder. You will replace it with a Boolean expression.



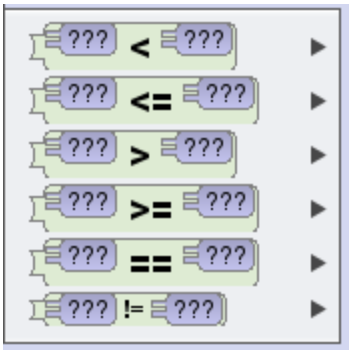


If you click on true, you have a choice of several values.

First you need to decide if this needs to be compound. Do you need to look at 2 comparisons? If there are 2 things (or more) that must be true use ??? && ???

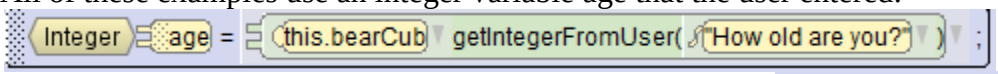
If there are 2 or more things and at least one of them must be true use ??? || ???

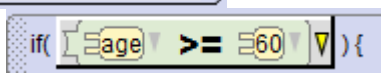
If you just need to do one comparison, then select whether the values you are comparing are decimal numbers (double), whole numbers (integer) or Strings.
After selecting one of these, you will have a chance to make additional choices.
If you select either decimal numbers or whole numbers, then you will be able to select one of the comparisons shown. Follow the arrow to select the first value, then keep following the arrow to select the second value. Usually at least one of the values will be a variable and one will be a custom value. However, you can also select a place holder value for one and then drag in a value with a function or even a property of one of the objects.



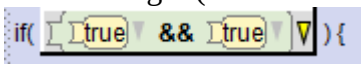
Examples:

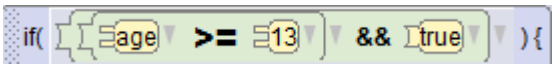
All of these examples use an integer variable age that the user entered:



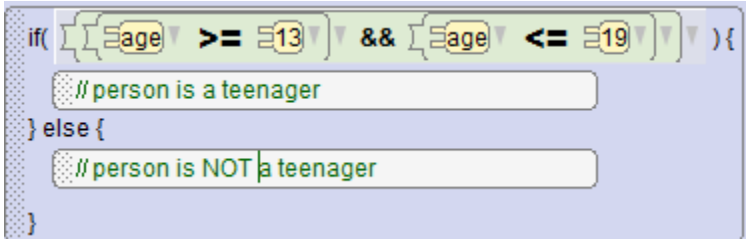
We want to know if the person is a senior (age 60 or older): 

We want to know if the person is a teenager (between the ages of 13 and 19 inclusive.

First we select true && true: 

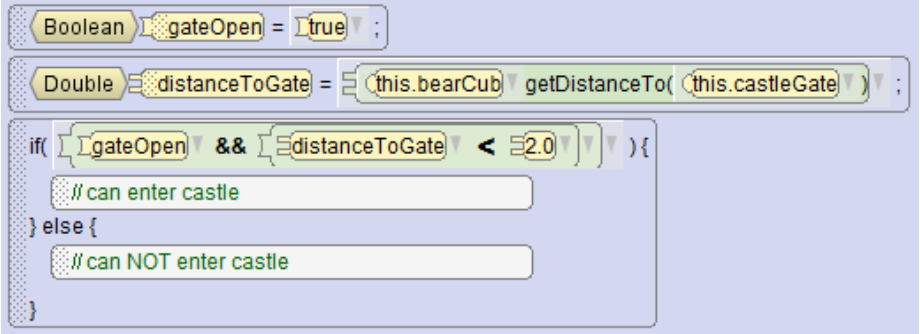
Next we replace the first true with age >=13 

Then we replace the second true with age <=19.



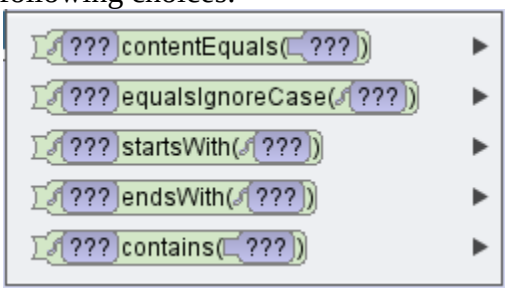
Here we have used comments to show the result, but we could of course put in any commands at this point.

It is important to note that the comparisons using the relational operators must be of the same type. However, && (AND) and || (OR) can be different types. For instance we might want to know if the gate is open and the actor is near the gate. We would need to have a Boolean value such as gateOpen and a decimal value distanceToGate. We would also have to decide what distance to gate is considered "near."

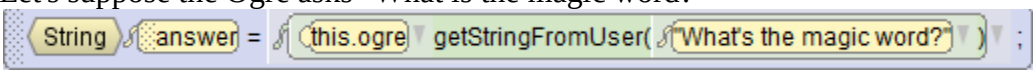


Comparing Strings

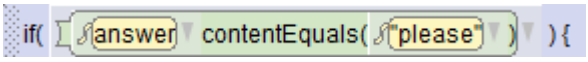
If we select want to compare two strings we select TextString Comparison and have the following choices:



Let's suppose the Ogre asks "What is the magic word?"



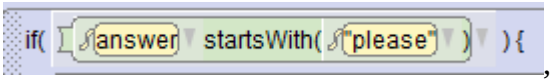
(The answer of course is "please".)

If we use the expression 

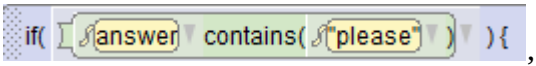
only the exact answer "please" will be true.

If we use the expression 

, then "please" "PLEASE","pLeAsE" or any other case of "please" will be accepted.

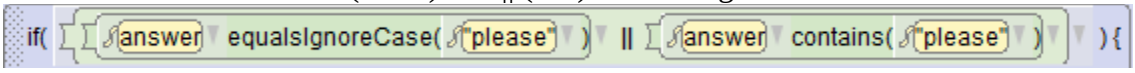
If we use the expression 

, then "please" "please let me in" or any other answer that starts with "please" will be accepted. However, "PLEASE" will not be accepted. endsWith is similar but looks at the ending instead of the start.

If we use the expression 

, then "please", "pretty please", "pleased" or any other answer that contains with "please" will be accepted. However, it will not be accepted if "please" is not all lower case.

You can of course use && (AND) and || (OR) with strings:



Although Alice does not have the full range of string handling functions that other languages have, you should be able to find a way to check for the string you want. You can also tell the user to use lower case, or to use a one word answer.