

[If-Then Rules]

Let's have a look at some different if-then rules in scratch. To have an if-then rule, then that means we need to have some logical statement that we're going to check the truth of – and this means we either need to input some values, or we need to use some of the other scripts to sense data. Let's look at the latter way first.

First, we'll create another sprite, which is just going to be a straight line that we'll place down the middle so that its x position is 0. Now for our program, we'll start again with a flag from Events so that it runs the script when we press the green flag and our if-then statements are found under the Control category. You'll see here that there's a little hexagon, that's where a true/false statement goes, and then in the gap is where we put the action to do IF the statement is true. So we're just going to create a little algorithm that tells us if the cat is on the right hand side of the line. How does the computer know this? Well the computer can sense the position of the cat, and under Motion, you can see this little block here that says x-position. If we click the check-box then this position will be displayed in the window. Now, we know that the cat is on the 'right' if its x position is greater than zero. So under Operators, we have a long hexagon box which will evaluate whether something is above another. I can put the x position here on the left – so that's a variable that will change depending on where the cat is, and then I want it to be greater than 0. Now for some action to happen if this is true, well let's just make the cat tell us it's on the right. Under Looks, there are some different blocks for the cat doing something. Now when we run this program if the cat is on the left, nothing happens. That's because the if statement read that the condition was false, and so didn't execute the "say I'm on the right command". However if we run the program when the cat is on the right, it speaks to us.

In this case, the input was based on the position of the cat to start off. However we can also ask for some user input. Let's create a

very simple program that multiplies a user input by 3.

We start with the flag again, and now under the Sensing category, we see these blocks that allow us to ask a question, and the answer becomes a variable that we can use in other program blocks. So we'll ask, "What number should I multiply?" and then we'll reply using one of the blocks from Look (like we did before) and we can use the multiply two numbers operator from Operators. In here we'll have the answer block (which will be the answer to the first question) and we can input a 3 here.

So when we run this program, the cat just takes the user input and multiplies it by 3.

How about if we want something a little more tricky, for example if we want to know whether a number is divisible by 3 or not?

Well, here we can use an if-then and else statement. We'll change the question and then for the logical statement we're actually going to use the mod function. For positive numbers, this function essentially tells us the remainder when you divide the first number by the second number. So if $a \bmod b = 0$, then a is divisible by b . (You can see how useful this is if we wanted to make a primes algorithm). Now we can build this block, so if the answer, is divisible by 3, answer mod 3 equals zero, then, we'll say Yes and if not, we'll say no.

We've now looked at a lot of the different kinds of blocks. Have a play around and see what kinds of programs you can create!