Conditions

Life is full of conditions: if you complete your homework, your mum will let me play, if you don't, she will scold you; if you go to Domino's on Wednesday you'll get 50% off on the price, but if you go on any other day, you won't get any discount, etc.

In the real world, you luckily have your brain to check what the condition is, whether it is met or not, and respond accordingly. But a computer isn't that lucky, unfortunately (or fortunately?). Even though it can perform every task at lightning-fast speed, you still need to give instructions to perform those tasks; even checking the conditions.

This is where the conditional statements come in handy. These statements allow the program to check the conditions by testing a variable against a value and act accordingly. A program that has conditional statements is called a Conditional Program, and the process is known as Conditional Programming.

Conditional Blocks

Conditions are expressions that evaluate to either true or false. They are used to define the criteria upon which certain actions or decisions within a program depend. Conditions often involve the comparison of variables, values, or the result of other logical operations.Conditional statements, also known as control structures or flow control statements, are programming constructs that dictate the execution of different code blocks based on the evaluation of specified conditions

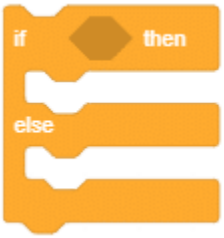
if () then Block

The **if () then** block will check whether the specified condition is true or not. If it is true, the blocks inside it will run, and then the script involved will continue. If

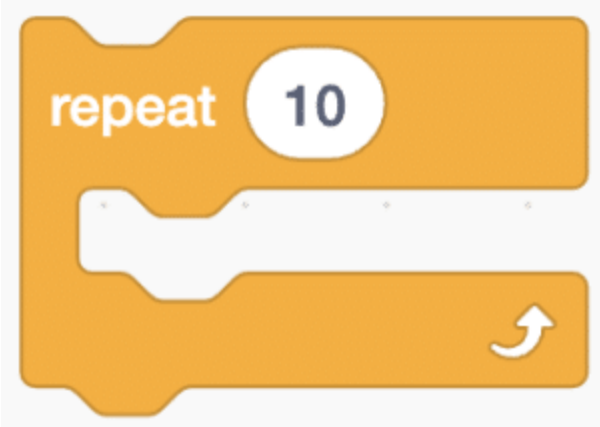
the condition is false, the code inside the block will be ignored and the script will move on. The condition is checked only once.

if () then else Block

The **if () then else** block will check whether the specified condition is true or not. If the condition is true, the code held inside the first C (below the **if** arm) will run. Then, the script will continue; if the condition is false, the code inside the second C (below the **else** arm) will run. (unlike the if () then block).



The **repeat ()**blocks held inside this block will loop a given amount of times, before allowing the script to continue. If a decimal is put in, the number is rounded up.



In the **forever** block**,** the Blocks held inside will be in a loop — just like the Repeat () block and the Repeat Until () block, except that the loop never ends (unless the stop sign is clicked, the Stop All block is activated, or the stop script block is activated within the loop). Due to this infinite loop, the block has no bump at the bottom; having a bump would be pointless, as the blocks below it would never be activated.

The block **wait() seconds** pauses its script for the specified amount of seconds — the wait can also be a decimal number.