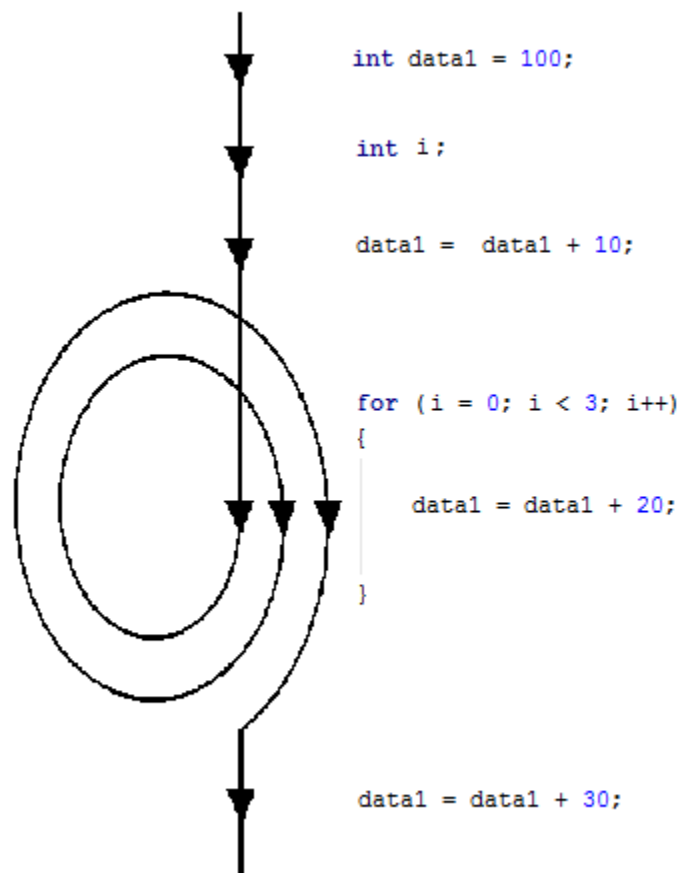# Loops

Loops are everywhere in life and programming. These are logical features that allow a computer to do some task over and over again. This is seen in everyday life from our daily routines to repetition in school and work. In programming, loops are very efficient and make our lives possible and easier. Without loops, we cannot write computer programs that are easy to understand and execute. For instance, we will have to take hours and days longer to write simple programs that would be possible to write in under 5 minutes using loops. With both conditionals and loops, we can do many operations and computations in programs very efficiently and accurately.


```
int data1 = 100;

int i;

data1 =  data1 + 10;

for (i = 0; i < 3; i++)
{
    data1 = data1 + 20;
}

data1 = data1 + 30;
```

Reference: https://en.wikipedia.org/wiki/For_loop

Exercise 1

Objective: Understand what a loop means and why it is important

Assume we have a box that we can fill with chocolates. Consider the following sequence of steps:

1. Check for chocolates in the box.
2. If there are chocolates, eat a chocolate and then return to step 1.
3. Save the box of chocolates

a) At the end of step 3, how many chocolates are there in the box assuming there are currently 12 chocolates in the box before step 1?

Possible answer:

There will be 0 chocolates at the end of step 3. Initially before step 1, there are 12 chocolates in the box. In step 2, we check if there are any chocolates left, in which case we eat a chocolate and then return to step 1. We will again now be at step 1 with 11 chocolates left and we will continue this until we have no chocolates left at the condition check at step 2. This will allow us to go to step 3 and save the box of chocolates.

b) How would it be possible to modify the sequence of steps so that, at the end of step 3, two chocolates in the box (assuming the box is full at the beginning of the process)?

Possible answer:

As shown in part (a), we will have no chocolates left in the box at the end of step 3. This happens because we keep eating a chocolate at step 2 as long as there is at least 1 chocolate left in the box and then we move back to step 1. To make sure that 2 chocolates are left, we can change the condition in step 2 to eat a chocolate and move back to step 1 only and if only there are at least 3 chocolate left in the box. This will ensure that once we eat a chocolate, we are at 2 chocolates and then we will never eat another chocolate and move on to step 3 with at least 2 chocolates in the box.

These steps can be written as the following:

1. Check for chocolates in the box.
2. If there are at least 3 chocolates in the box, eat a chocolate and then return to step 1.
3. Save the box of chocolates

c) Suppose loops do not exist on a different planet. Think about what the original sequence of steps would look like on this planet and how many steps there would be in total if we could not write "return to step 1" or go back to any previous step. The original sequence of steps currently has only 3 steps. Keeping this in mind and comparing the two approaches with and without loops, what is one benefit of using loops?

Possible answer:

If we did not have loops, we would have the following sequence of steps assuming we have a total of 6 chocolates initially in the box:

1. Check for chocolates in the box. [6 chocolates left]
2. If there are any chocolates left, eat a chocolate [5 chocolates left]
3. Check for chocolates in the box. [5 chocolates left]
4. If there are any chocolates left, eat a chocolate [4 chocolates left]
5. Check for chocolates in the box. [4 chocolates left]
6. If there are any chocolates left, eat a chocolate [3 chocolates left]
7. Check for chocolates in the box. [3 chocolates left]
8. If there are any chocolates left, eat a chocolate [2 chocolates left]
9. Check for chocolates in the box. [2 chocolates left]
10. If there are any chocolates left, eat a chocolate [1 chocolate left]
11. Check for chocolates in the box. [1 chocolate left]
12. If there are any chocolates left, eat a chocolate [0 chocolates left]

The major benefit of using loops is efficiency. The number of steps needed in a computer program is much greater without loops than it is with loops. Without loops, the computer would take more time and energy to do computations and condition checks. Therefore, loops are very beneficial for efficient computations.


Reference: https://www.geeksforgeeks.org/break-statement-cc/
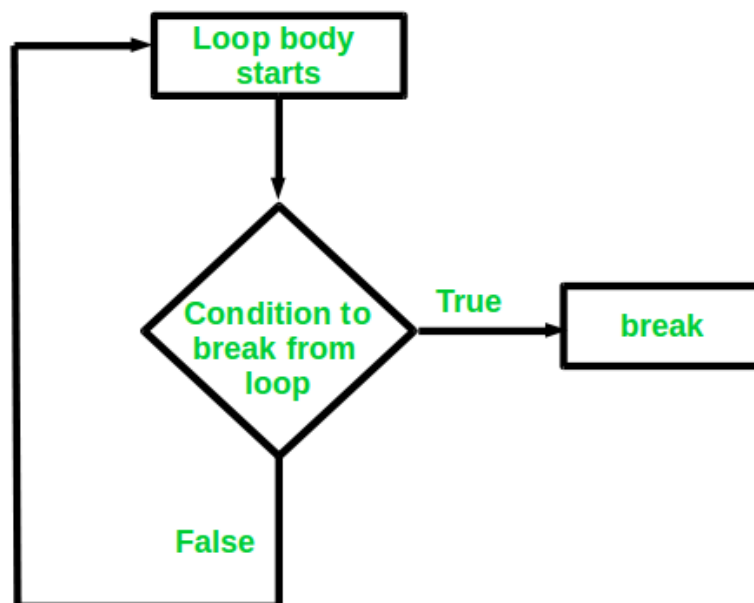
Exercise 2

Objective: Program computers using loops and conditionals

On your computer, go to https://studio.code.org/s/mc/stage/1/puzzle/1. It is a tutorial of programming using Minecraft. Follow the tutorial! After, we will discuss. Remember, loops and conditionals together give you and the computer more power!

Reference: https://studio.code.org/s/mc/stage/1/puzzle/1
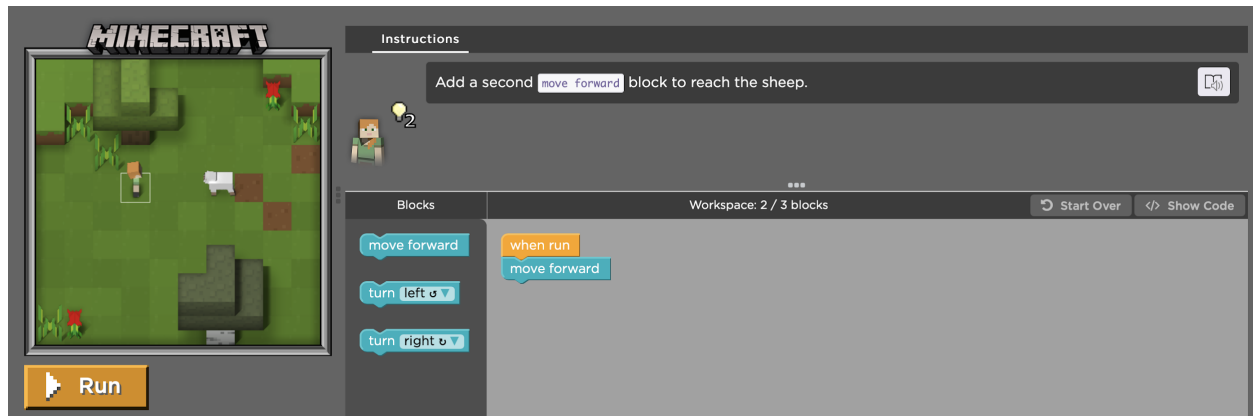
Exercise 3

Objective: To reflect on loops and conditionals in our daily life

With your partner, discuss when, in our daily lives, we encounter situations that look like conditionals or with loops. As was briefly mentioned in Exercise 1, one example of a conditional is whether I eat or do homework depending on if I already have done my homework or not. We can use loops here if we have more than 1 person, such as in a group. We can go back to the starting step for each person in our group. This is a simple example. Feel free to come up with creative examples of loops and conditionals in your daily life!

Possible answers:

There are many answers that are correct for this exercise. Some examples are briefly detailed below:

1. Checking the weather first to see if I need to bring an umbrella or not bring an umbrella when I go outside.
2. We repeat our favorite songs many times or watch out favorite movies many times.
3. Many people have a very strict daily routine where the same tasks are done over and over again.
4. When I am hungry, I go eat food.
5. When I do not get enough sleep, I am tired
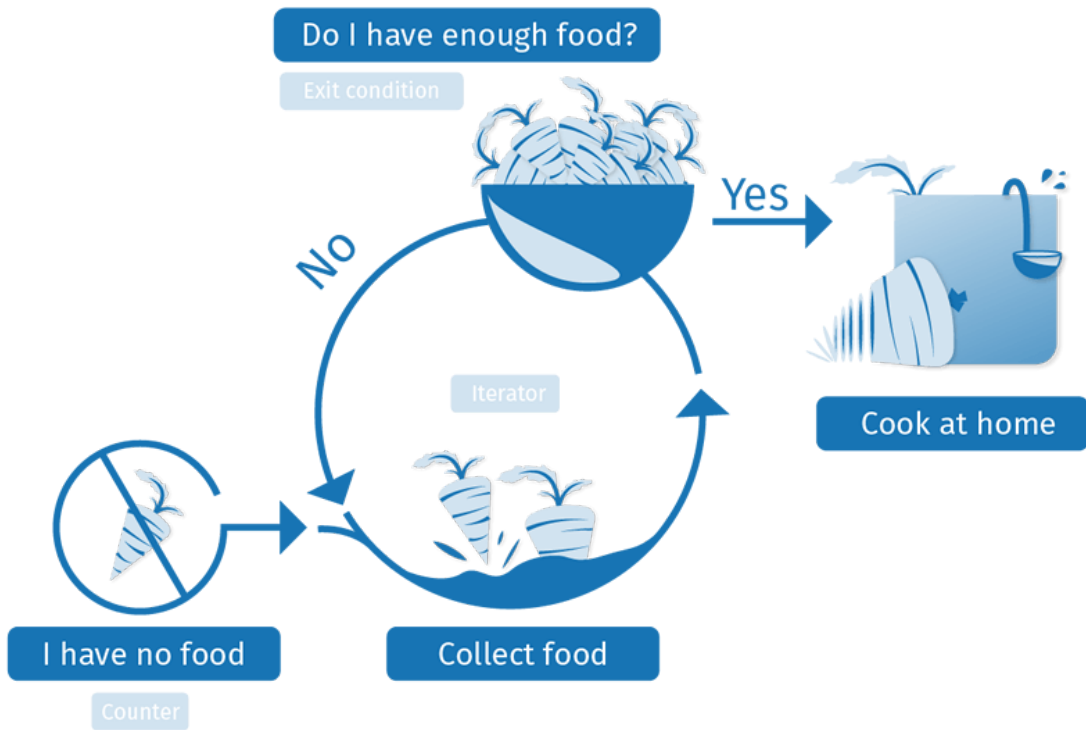
Reference:

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building_blocks/Looping_code