# Subject: Repeat Loops

## Overview

In our new lesson, we will explain to our users how to write repeating commands briefly and to the point. Well comprehension of this subject will form the base of many subjects which we will learn in the future.

In this lesson, your students will learn both new commands and how to use our commands within repeat.

During the explanation of the subject you can also find a sample from your own life in order to understand the intended purpose of the repeat expressions in the most effective way. Ensure the concepts become more understandable by giving examples from daily life. Our activity below will form an example for this.

## Activity

What do we do respectively when we get up in the morning? *(such as I got up – I had breakfast-- I came to school)*

*"What would you do if I asked you to perform this process for 5 days in weekdays?".*

*While we complete our task, we will use a new command for our repeating expressions in order to specify the situations repeating by the number we provide. Thus we will have written the processes we must perform once, but since they are included in our repeat command, they will repeat by the number we provide. Instead of writing our repeating expressions in the same way more than once, we need to indicate it with **repeat a:** command in the Codementum platform. If we need to form a sample for this activity, it will be as;*

```
repeat 5:
    getting up-having breakfast-coming to school
```

## Commands to be Used

repeat a:,forward(), turn(), forwardUp(), forwardDown(), beaver.forward()

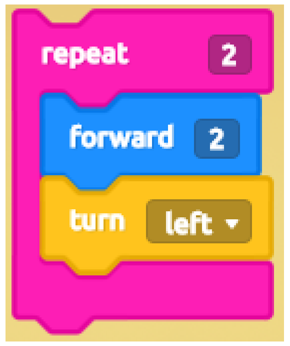
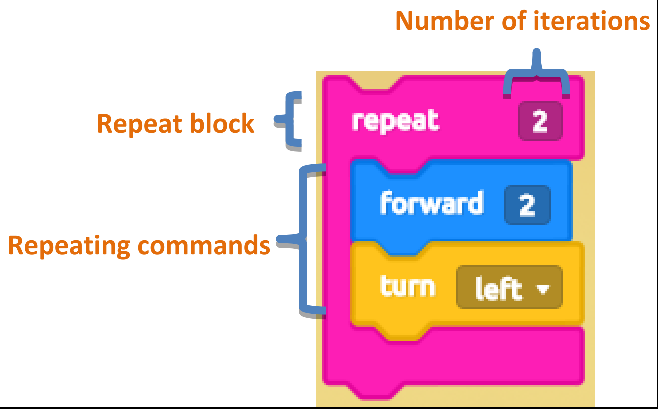
## Blocks and Button icons to be used in the training about this subject

**Block:** You will be able to make block coding by dragging and dropping the related blocks.

**Button:** Buttons will be seen on the screen during coding adventures and when you click, it will write the related code on the screen automatically.

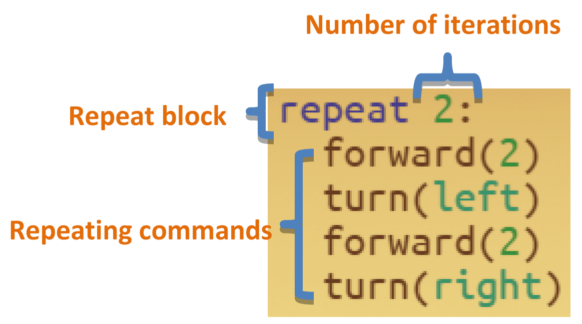
**Explanation:** You can find the basic knowledge you need in the explanation section.

| Block | Explanation |
|---|---|
| | Thanks to the repeat loop, instead of writing our repeating processes more than once, it is enough to write it within the loop and specify the repeat number we want. Thus the number of our command lines decreases and we can reach the most efficient and correct commands, according to three star application.<br><br>*<u>Sample usage:</u>*<br>repeat 3:<br>    forward(5)<br>    turn(left) |
| **Button** | |
| | |

| BLOCK | |
|---|---|
| **Usage** | **Explanation** |
|  |  Number of iterations<br>Repeat block<br>Repeating commands<br>repeat 2<br>forward 2<br>turn left |

codementum.com

## PYTHON

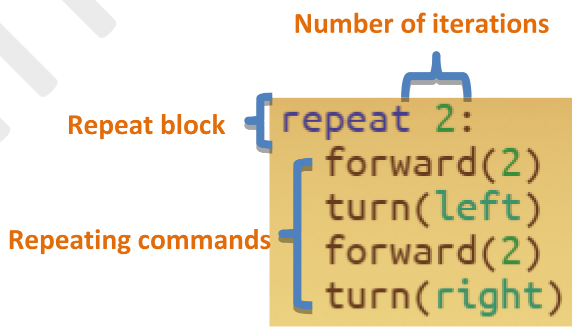
| Usage | Explanation |
|---|---|
| `repeat 2:`<br>`    forward(2)`<br>`    turn(left)`<br>`    forward(2)`<br>`    turn(right)` |  |

```
repeat 2:
    forward(2)
    turn(left)
    forward(2)
    turn(right)
```

### Erroneous writing forms

| Python | | |
|---|---|---|
| `repeat 2:`<br>`forward(2)`<br>`turn(left)` | `repeat2`<br>`    forward(2)`<br>`    turn(left)` | `repeat 2:`<br>`    forward(2)`<br>`    turn(left)` |
| ✗ | ✗ | ✓ |
| **Why is it wrong?** | **Why is it wrong?** | **Correct form:** |
| You should write the commands which you want to repeat within “Repeat” after indenting with the tab key. | Number of repeats should be written following spacing after “Repeat” command and it should be used with a colon “:” | Our code is correct due to paying attention to spacing between expressions and the indenting, which should be done with the tab key for the commands written within Repeat. |

## JAVASCRIPT

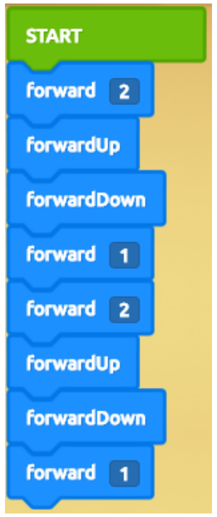
| Usage | Explanation |
|---|---|
| `repeat 2:`<br>`    forward(2)`<br>`    turn(left)`<br>`    forward(2)`<br>`    turn(right)` |  |

```
repeat 2:
    forward(2)
    turn(left)
    forward(2)
    turn(right)
```

### Erroneous writing forms

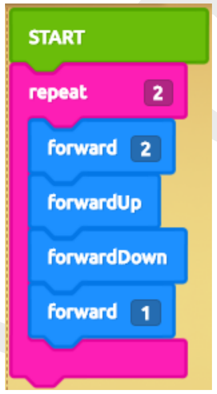
| JavaScript | | |
|---|---|---|
| `repeat(2)`<br>`{forward(2)`<br>`turn(left)`<br>`forward(2)`<br>`turn(right)}` | `repeat(2)`<br>`{`<br>`forward(2)`<br>`turn(left)`<br>`forward(2)`<br>`turn(right)`<br>`}` | `repeat(2){`<br>`    forward(2)`<br>`    turn(left)`<br>`    forward(2)`<br>`    turn(right)`<br>`}` |
| ✗ | ✗ | ✓ |

**Sample Solution:** (Without using repeat)

| Algorithm | Block | Python | Javascript |
|---|---|---|---|
| • Start<br>• Move 2 squares forward<br>• Move up<br>• Move down<br>• Move 1 square forward<br>• Move 2 squares forward<br>• Move up<br>• Move down<br>• Move 1 square forward |  | forward(2)<br>forwardUp()<br>forwardDown()<br>forward(1)<br>forward(2)<br>forwardUp()<br>forwardDown()<br>forward(1) | forward(2)<br>forwardUp()<br>forwardDown()<br>forward(1)<br>forward(2)<br>forwardUp()<br>forwardDown()<br>forward(1) |

```
START
forward 2
forwardUp
forwardDown
forward 1
forward 2
forwardUp
forwardDown
forward 1
```

**Sample Solution:** (By using repeat)

| Algorithm | Block | Python | Javascript |
|---|---|---|---|
| • Start<br>• 2 times within repeat<br>  ▪ Move 2 squares forward<br>  ▪ Move up<br>  ▪ Move down<br>  ▪ Move 1 square forward |  | repeat 2:<br>  forward(2)<br>  forwardUp()<br>  forwardDown()<br>  forward(1) | repeat 2:<br>  forward(2)<br>  forwardUp()<br>  forwardDown()<br>  forward(1) |

```
START
repeat 2
  forward 2
  forwardUp
  forwardDown
  forward 1
```