

## Using a Repeat Block to Generate Simple Number Patterns

Let's write code in Scratch that uses a **repeat** block to generate simple growing and shrinking patterns.

In Part 1, you'll write code for the basic Scratch application.  
In Part 2, you'll alter the code you've just written to generate a variety of growing and shrinking patterns.

### Part 1: Writing the basic code for a number pattern

The first number pattern you generate with your program will be:  
0, 5, 10, 15, 20, 25

- What is the pattern rule for this set of numbers?

*Pseudocode* is an informal way of describing a computer program—it is somewhere between everyday language and programming code. Computer programmers, or coders, often use pseudocode when planning their programs.

The pseudocode for the application looks like this:

```
clear mypattern[]  
number = 0  
repeat 6 times  
    output number  
    add number to mypattern[]  
    change number by 5  
end repeat
```

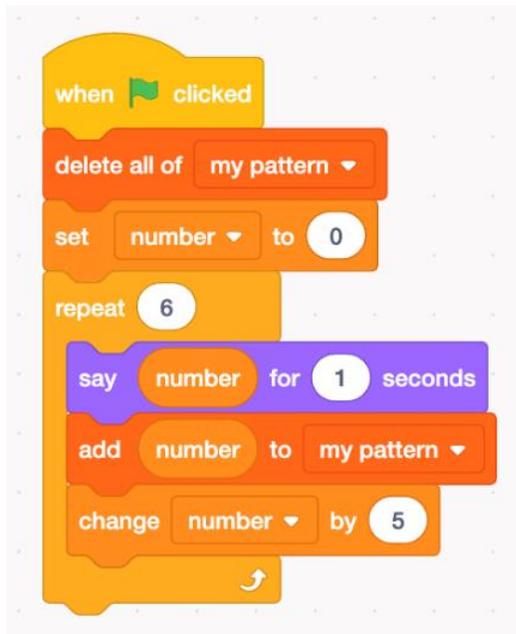
In this code, anything that might have been stored in the list is deleted.

The starting value of our **number** variable is set to 0.

In the repeat, **number** is output, saved in a list called **mypattern**, and then changed by 5.

## Using a Repeat Block to Generate Simple Number Patterns (cont'd)

Since this is repeated 6 times, you will see 6 numbers in your pattern. Let's write this code in Scratch. The completed code is shown below, so if you have experience with Scratch you may proceed to recreate it now and skip to Part B. Otherwise, follow the guided materials below.



**Note:** This code uses both a variable and a list. *Variables* can hold one value at a time, whereas a *list* can hold many values in an organized manner. In this code, **number** is a variable and **mypattern** is a list.

### Guided Materials\*

\*Skip to Part 2 if you have already written the Scratch code based on the image above.

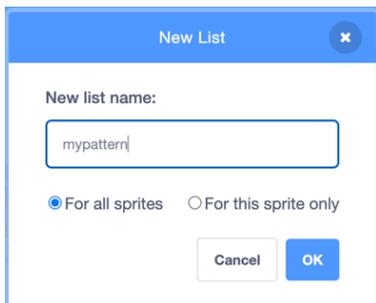
Go to <https://scratch.mit.edu/> and select **Create** to begin.

1. From **Events**, drag the when **green flag is clicked** block to the code editing area.

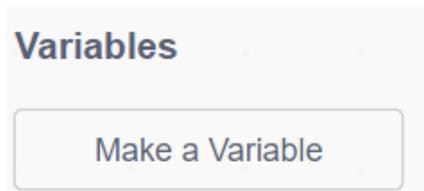


## Using a Repeat Block to Generate Simple Number Patterns (cont'd)

2. Create a list called **mypattern**. To do this, from **Variables**, click on **Make a List**. Enter the name **mypattern** and click **OK**. To make sure the list starts out empty each time you run the code, add the **delete all of mypattern** block to your code.



3. Create a variable called **number**. To do this, from **Variables**, select **Make a Variable** and call it **number**. Add the **set number to 0** block to your code.



4. From **Control**, add the **repeat** block to your code and change the repeat value to **6**.

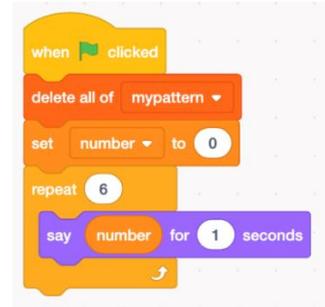


## Using a Repeat Block to Generate Simple Number Patterns (cont'd)

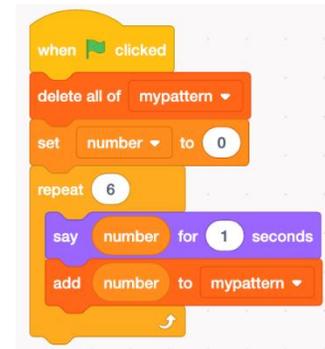
For Steps 5 to 7, you will be adding blocks inside the repeat block.

5. From Looks, add the say hello for 2 seconds block to your code. Change the value from 2 to 1 seconds.

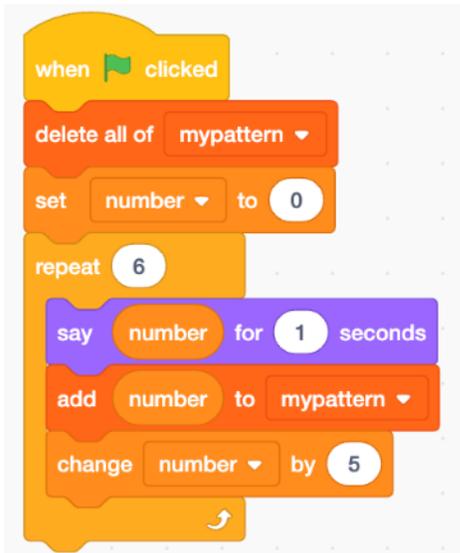
From **Variables**, drag the **number** block to inside of the **say** block to replace **Hello!**.



6. From **Variables**, add the **add thing to mypattern** block to your code. Drag the **number** block to inside of the **add** block to replace **thing**.



7. Finally, from **Variables**, add the **change number by 1** block to the code, and adjust the value to **5**.



If you have a Scratch login, save the project in your Scratch account by selecting **Save now** from the **File** menu.

A login is not required to work with the code, but you will not be able to save your application without it.

## Using a Repeat Block to Generate Simple Number Patterns (cont'd)

8. Click on the green flag above the stage to execute the code.

The sprite should output the following numbers, one at a time, in the speech bubble:

0, 5, 10, 15, 20, 25.

After each number appears in the bubble, it should be added to the **mypattern** list.

If the application is not working as expected, look through the code carefully and debug by checking whether each part of the code matches the blocks shown in Steps 1 to 7.



| mypattern    |    |
|--------------|----|
| 1            | 0  |
| 2            | 5  |
| 3            | 10 |
| 4            | 15 |
| 5            | 20 |
| 6            | 25 |
| + length 6 = |    |

### Part 2: Altering the code to generate other growing and shrinking patterns

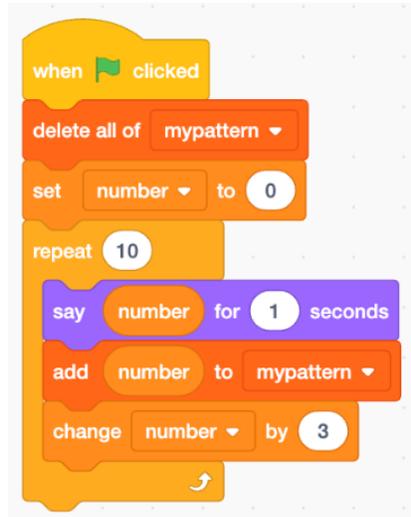
1. Alter the code to generate a pattern of numbers that increases by 3 each time. Output 10 numbers in the pattern, beginning at 0. The numbers output should be: 0, 3, 6, 9, 12, 15, 18, 21, 24, 27.

Here's the pseudocode to generate this new pattern of numbers:

```
clear mypattern[]
number = 0
repeat 10 times
  output number
  add number to mypattern[]
  change number by 3
end repeat
```

## Using a Repeat Block to Generate Simple Number Patterns (cont'd)

The Scratch code will look like this:



2. Alter this pattern slightly to begin at **9**, rather than **0**.  
The numbers generated should be:  
9, 12, 15, 18, 21, 24, 27, 30, 33, 36.
3. Alter the code to generate each pattern of numbers.  
What is different about the patterns in parts d) and e) compared to the patterns in parts a), b), and c)?
  - a) 0, 4, 8
  - b) 10, 12, 14, 16, 18, 20, 22
  - c) 100, 200, 300, 400, 500
  - d) 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 0
  - e) 0, -6, -12, -18

If you get stuck, check with a classmate and/or your teacher for tips.

### Additional Challenge

Generate your own patterns and challenge your classmates to write the corresponding code.