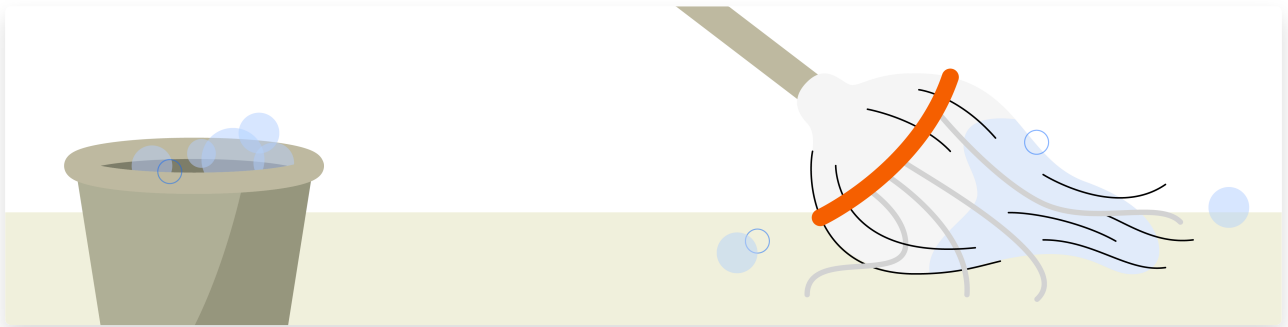


Loops

Textbook

Loops



Chores. No one *wants* to do them, but they need to get done. Sometimes we think "I wish this would just happen on its own"? There are some robots that can do some chores. They repeat the same task over and over without stopping! In programming, loops are like that robot—they take care of repetitive tasks so you don't have to do them yourself.

What is a Loop? A loop in programming is a set of instructions that repeats itself until a specific condition is met. Loops are like the energizer bunny of code—they keep going until you tell them to stop!

Types of Loops There are different kinds of loops, but they all share the same purpose: to save time by automating repetitive tasks.

1. For Loops

- A for loop runs a specific number of times. For example, if you want to print "Hello!" five times, a for loop will handle it perfectly.

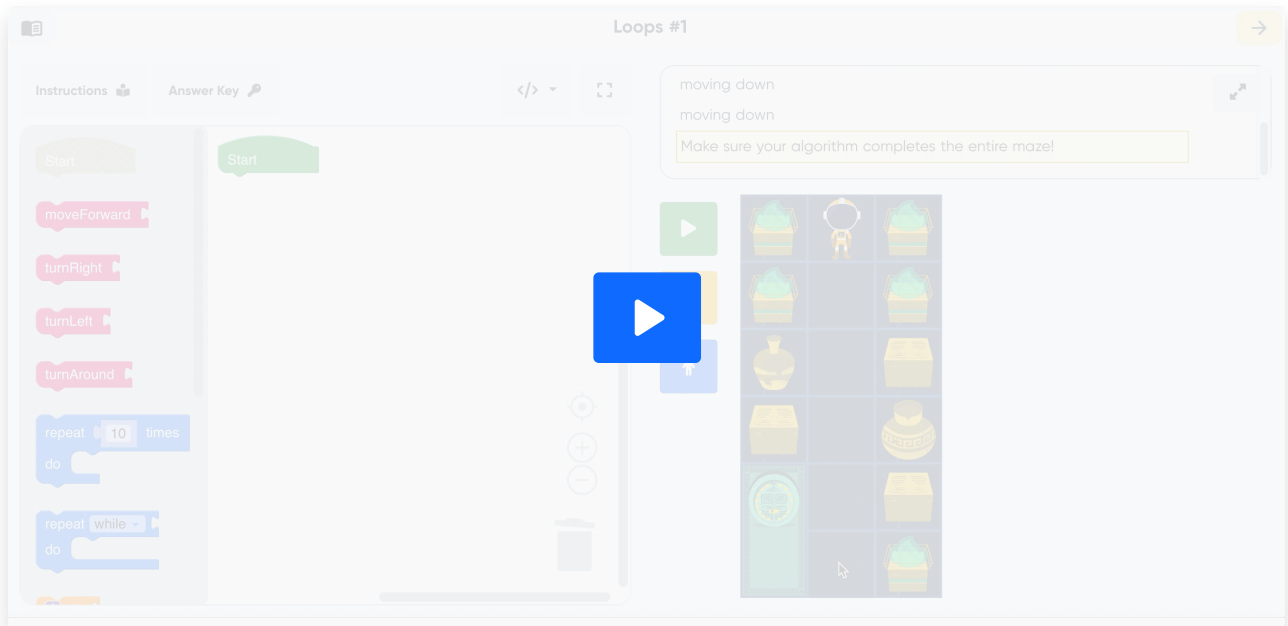
2. While Loops

- A while loop keeps running as long as a condition is true. It's great when you're not sure how many times something needs to repeat, like waiting for a player to finish a game level.

Why Are Loops Useful? Loops can save us time and reduce errors. Instead of writing the same code over and over, you write it once, and the loop takes care of the rest. For example:

- Drawing a row of stars in a video game.
- Checking for updates on your computer every hour.

Loops Video



There are two ways to code a loop in [JavaScript](#):

1. Using a [while loop](#)
2. Using a repeat loop

Coding Loops in Block Code



Coding Loops using a While Block

You can program a loop using a while block.



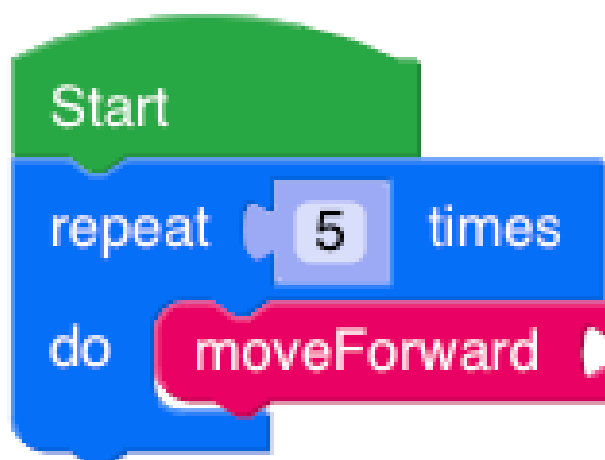
Drag and connect a condition next to where it says "while." For example, repeat while `isClearAhead`. Then, drag and connect the action you want your character to follow while it is clear ahead. For example, `moveForward`. The while loop tells the computer that while it is clear ahead, the character will continue to move forward.

Coding Loops using a Repeat Block

The other way to program a loop in the puzzles is by using a repeat block.



Choose how many times you would like the code to repeat, then connect the action(s) you want your character to follow in the space next to the word "do." For example, if I want my character to move forward 5 times it would look like this:



Coding a While Loop in JavaScript

Coding a while loop using [syntax](#) looks like this:

```
1 while () {
```

```
2  
3 }
```

Your condition goes between the `()` parenthesis.

What you want your character to do goes between the `{ }` braces.

Your character will follow the code over and over again as long as the condition between the `()` parenthesis is true. **WHILE** the condition between the `()` is true, the command between the `{ }` braces will run.

So if you want your character to move forward **WHILE** it is clear ahead, it would look like this:

```
1 while (isClearAhead()) {  
2   moveForward();  
3 }
```

The character will move forward over and over again as long as the path is clear ahead.

Coding a Repeat Loop in JavaScript

Coding a repeat loop in syntax is a little more advanced. Because of that, we recommend copying this code from the instructions.

```
1 for (var count = 0; count < ENTER NUMBER; count++) {  
2   ENTER DIRECTION  
3 }
```

Copy

Delete where it says **ENTER NUMBER** and type the number of spaces you want your character to move. Then delete where it says **ENTER DIRECTION** and type the action you want your character to follow. For example, if I want my character to move forward 5 times it would look like this:

```
1 for (var count = 0; count < 5; count++) {  
2   moveForward();  
3 }
```

Critical Thinking Questions

1. Can you think of a task in your daily life that would be easier if you could "loop" it? How would it help?
2. Why do you think programmers use loops instead of writing the same instructions multiple times?

Questions (5)

1. What is a loop in programming?

MULTIPLE CHOICE

Choose the correct answer:

- A. A programming loop is a set of instructions that repeats.
- B. An algorithm that includes every variable
- C. A sequence put in the correct order
- D. A code event that only happens once

2. True or False: Loops repeat information.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

3. How can loops make programming easier?

MULTIPLE CHOICE

Choose the correct answer:

- A. By making the computer run slower
- B. By adding more buttons to the screen
- C. By making the computer more colorful
- D. By allowing tasks to be done automatically and repeatedly

4. If you want your character to move forward 3 times in a row, which type of loop would you use?

MULTIPLE CHOICE

Choose the correct answer:

- A. While loop
- B. Forever loop
- C. Repeat loop

5. True or false: A while loop is a type of loop that repeats as long as a certain condition is true.

MULTIPLE CHOICE


Choose the correct answer:

- A. True
- B. False

Games (2)

1. Loops Typing

Full Screen Audio Instructions Restart Pause



0s 100%

Loops in programming re

2. Loops Memory Game

Full Screen

Audio

Instructions

Answer Key

Pause

Flips: 0

1

send 01 loop ob

2

brunoArnuol

3

brunoArnuol

4

brunoArnuol

5

send 01 loop ob

6

brunoArnuol

7

brunoArnuol

8

brunoArnuol

9

brunoArnuol

10




brunoArnuol

11

send 01 loop ob

12

brunoArnuol



Puzzles (15)

1. Loops #1

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

Minimum
Loops
1

2. Loops #2

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not



isClearAhead

isClearLeft

isClearRight

Minimum
Loops
1

3. Loops #3

Instructions  Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead

isClearLeft


isClearRight

Start







Minimum
Loops
1



4. Loops #4

Instructions  Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead

isClearLeft

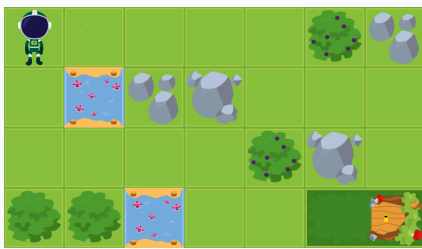
isClearRight

Start













Minimum
Loops
1






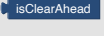

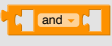

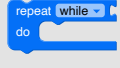





5. Loops #5




Instructions 





Answer Key 

</> 


Start 










Minimum
Loops
1




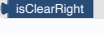

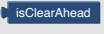



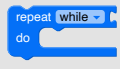

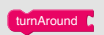



6. Loops #6




Instructions 





Answer Key 

</> 


Start 










Minimum
Loops
1




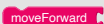
7. Loops #7


Instructions 


Answer Key 


</> 


Start 


moveForward 


turnRight 


turnLeft 


turnAround 


repeat 10 times 


do 


repeat while 


do 


if 


do 





end 

not 


isClearAhead 

isClearLeft 

isClearRight 





Minimum
Loops
1





A 5x5 grid world. The robot is at (5,1). There is a pond at (2,4) and (3,4). There are bushes at (1,2), (2,2), (3,2), (4,2), (4,3), and (5,4). There is a rock at (5,3). There is a tree at (5,5).

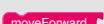
8. Loops #8


Instructions 


Answer Key 


</> 


Start 

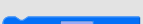
moveForward 


turnRight 

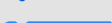
turnLeft 


turnAround 


pickUpKey 


repeat 10 times 


do 


repeat while 


do 


if 





do 

end 

not 


isClearAhead 

isClearLeft 




Variable
EMPTY


Minimum
Loops
1




A 5x5 grid world. The robot is at (5,5). There is a key at (1,1). There is a tree at (2,2). There are bushes at (1,2), (2,3), (3,3), (4,3), and (5,4). There are rocks at (2,1) and (3,1).

9. Loops #9

Instructions 

Answer Key 

</>



Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end


not


isClearAhead


isClearLeft





isClearRight

Start












Minimum
Loops
1




10. Loops #10

Instructions 

Answer Key 

</>



Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end


not


isClearAhead


isClearLeft





isClearRight

Start














Minimum
Loops
1









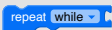






11. Loops #11




Instructions 





Answer Key 

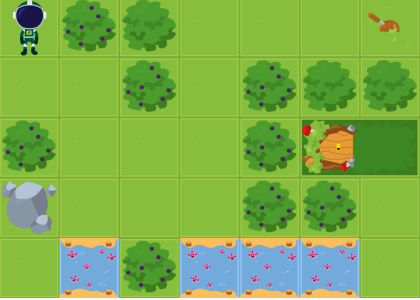
Start 









Minimum
Loops
1






A 10x10 grid map. The robot starts at (1,1). There are trees at (1,2), (1,3), (2,3), (2,4), (2,5), (3,1), (3,2), (3,4), (3,5), (4,4), (4,5), (4,6), (5,2), (5,3), (5,4), (5,5), (5,6), (6,2), (6,3), (6,4), (6,5), (6,6), (7,2), (7,3), (7,4), (7,5), (7,6), (8,2), (8,3), (8,4), (8,5), (8,6), (9,2), (9,3), (9,4), (9,5), (9,6), (10,2), (10,3), (10,4), (10,5), (10,6). A river is at (6,7), (6,8), (6,9), (6,10). A lever is at (4,7). A goal is at (1,10).






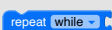






12. Loops #12




Instructions 





Answer Key 

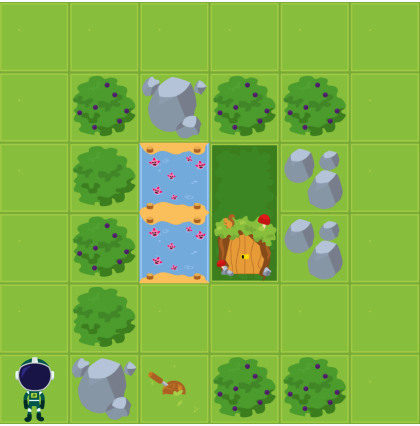
Start 







Minimum
Loops
1



A 10x10 grid map. The robot starts at (10,1). There are trees at (2,2), (2,3), (3,2), (3,3), (3,4), (4,2), (4,3), (4,4), (4,5), (5,2), (5,3), (5,4), (5,5), (5,6), (6,2), (6,3), (6,4), (6,5), (6,6), (7,2), (7,3), (7,4), (7,5), (7,6), (8,2), (8,3), (8,4), (8,5), (8,6), (9,2), (9,3), (9,4), (9,5), (9,6), (10,2), (10,3), (10,4), (10,5), (10,6). A river is at (4,7), (4,8), (4,9), (4,10). A lever is at (5,7). A goal is at (1,10).

13. Loops #13

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

Start

Minimum Loops

1

14. Loops #14

Instructions
Answer Key
</>

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

Start

▶

⬆

🧑

🔊

Minimum
Loops

1

15. Loops #15

Instructions

Answer Key

</>

🔍

Start

moveForward

turnRight

turnLeft

turnAround

pullLever

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

Start

▶

⬆

⬆

⬆

Minimum Loops

1

Answer Keys & Solutions

Questions

1. What is a loop in programming?

MULTIPLE CHOICE

Correct Answer:

- A. A programming loop is a set of instructions that repeats. ✓ Correct
- B. An algorithm that includes every variable ✗ Incorrect
- C. A sequence put in the correct order ✗ Incorrect
- D. A code event that only happens once ✗ Incorrect

Explanation:

Loops save lots of time in coding when things happen again and again

2. True or False: Loops repeat information.

MULTIPLE CHOICE

Correct Answer:

- A. True ✓ Correct
- B. False ✗ Incorrect

Explanation:

Loops are instructions that happen again and again.

3. How can loops make programming easier?

MULTIPLE CHOICE

Correct Answer:

- A. By making the computer run slower ✗ Incorrect
- B. By adding more buttons to the screen ✗ Incorrect
- C. By making the computer more colorful ✗ Incorrect

D. By allowing tasks to be done automatically and repeatedly

✓ Correct

Explanation:

Think about how loops help in automating repetitive tasks on the computer.

4. If you want your character to move forward 3 times in a row, which type of loop would you use?

MULTIPLE CHOICE

Correct Answer:

A. While loop

✗ Incorrect

B. Forever loop

✗ Incorrect

C. Repeat loop

✓ Correct

Explanation:

Consider which loop allows you to specify how many times an action should be repeated.

5. True or false: A while loop is a type of loop that repeats as long as a certain condition is true.

MULTIPLE CHOICE

Correct Answer:

A. True

✓ Correct

B. False

✗ Incorrect

Explanation:

Think about the structure of the while loop where an action is repeated while a condition remains true.













Games

1. Loops Typing

Typing game - no answer key needed. Students practice typing the provided content.

2. Loops Memory Game

Memory Game Pairs:

1.  A blue code block that says repeat while...do ↔  A blue code block that says repeat while...do
2.  A blue code block that says repeat 10 times...do ↔  A blue code block that says repeat 10 times...do
3.  A pink code block that says moveForward ↔  A pink code block that says moveForward
4.  A pink code block that says turnRight ↔  A pink code block that says turnRight
5.  A pink code block that says turnLeft ↔  A pink code block that says turnLeft
6.  A pink code block that says turnAround ↔  A pink code block that says turnAround

Students must find all matching pairs by flipping cards and remembering their positions.

Puzzles

1. Loops #1

Solution:

```
1 while (isClearAhead()) {  
2   moveForward();  
3 }  
4 turnLeft();  
5 moveForward();  
6 moveForward();  
7 moveForward();
```

2. Loops #2

Solution:

```
1 turnRight();  
2 while (isClearAhead()) {  
3   moveForward();  
4 }  
5 turnLeft();  
6 while (isClearAhead()) {  
7   moveForward();  
8 }  
9 turnLeft();  
10 while (isClearAhead()) {  
11   moveForward();  
12 }  
13 turnLeft();  
14 moveForward();  
15 moveForward();
```

3. Loops #3

Solution:

```
1 turnLeft();
2 while (isClearAhead()) {
3     moveForward();
4 }
5 turnRight();
6 moveForward();
7 moveForward();
8 turnRight();
9 while (isClearAhead()) {
10    moveForward();
11 }
12 turnLeft();
13 moveForward();
14 moveForward();
15 turnLeft();
16 while (isClearAhead()) {
17     moveForward();
18 }
```

4. Loops #4

Solution:

```
1 while (isClearAhead()) {
2     moveForward();
3 }
4 turnLeft();
5 while (isClearAhead()) {
6     moveForward();
7 }
8 turnRight();
9 moveForward();
10 turnLeft();
11 moveForward();
12 moveForward();
```

5. Loops #5

Solution:

```
1 turnAround();
2 moveForward();
3 turnRight();
4 moveForward();
5 moveForward();
6 turnRight();
```

```
7 while (isClearAhead()) {
8     moveForward();
9 }
10 turnRight();
11 while (isClearAhead()) {
12     moveForward();
13 }
14 turnRight();
15 while (isClearAhead()) {
16     moveForward();
17 }
18 turnRight();
19 while (isClearAhead()) {
20     moveForward();
21 }
```

6. Loops #6

Solution:

```
1 turnAround();
2 moveForward();
3 turnLeft();
4 while (isClearAhead()) {
5     moveForward();
6 }
7 turnLeft();
8 moveForward();
```

7. Loops #7

Solution:

```
1 turnAround();
2 while (isClearAhead()) {
3     moveForward();
4 }
5 turnRight();
6 while (isClearAhead()) {
7     moveForward();
8 }
```

8. Loops #8

Solution:

```
1 turnAround();
2 while (isClearAhead()) {
3     moveForward();
4 }
5 turnLeft();
```

```

6 while (isClearAhead()) {
7     moveForward();
8 }
9 turnLeft();
10 for (var count = 0; count < 2; count++) {
11     moveForward();
12 }
13 turnLeft();
14 moveForward();
15 turnRight();
16 for (var count2 = 0; count2 < 2; count2++) {
17     moveForward();
18 }
19 turnRight();
20 moveForward();
21 pickUpKey()
22 turnAround();
23 moveForward();
24 turnLeft();
25 while (isClearAhead()) {
26     moveForward();
27 }
28 turnRight();
29 moveForward();

```

9. Loops #9

Solution:

```

1 while (isClearAhead()) {
2     moveForward();
3 }
4 turnLeft();
5 while (isClearAhead()) {
6     moveForward();
7 }
8 turnRight();
9 moveForward();
10 turnLeft();
11 while (isClearAhead()) {
12     moveForward();
13 }
14 turnRight();
15 moveForward();

```

10. Loops #10

Solution:

```

1 moveForward();
2 turnRight();
3 while (isClearAhead()) {
4     moveForward();
5 }

```

```

6 turnRight();
7 while (isClearAhead()) {
8   moveForward();
9 }
10 turnLeft();
11 moveForward();
12 turnRight();
13 while (isClearAhead()) {
14   moveForward();
15 }
16 turnRight();
17 moveForward();
18 turnLeft();
19 moveForward();
20 turnRight();
21 while (isClearAhead()) {
22   moveForward();
23 }
24 turnRight();
25 moveForward();
26 turnLeft();
27 moveForward();
28 turnRight();
29 moveForward();
30 turnLeft();
31 while (isClearAhead()) {
32   moveForward();
33 }
34 turnLeft();
35 while (isClearAhead()) {
36   moveForward();
37 }

```

11. Loops #11

Solution:

```

1 moveForward();
2 turnLeft();
3 moveForward();
4 turnRight();
5 moveForward();
6 turnLeft();
7 while (isClearAhead()) {
8   moveForward();
9 }
10 turnLeft();
11 while (isClearAhead()) {
12   moveForward();
13 }
14 turnRight();
15 while (isClearAhead()) {
16   moveForward();
17 }
18 pullLever()
19 turnAround();
20 while (isClearAhead()) {
21   moveForward();

```

```

22 }
23 turnLeft();
24 while (isClearAhead()) {
25     moveForward();
26 }
27 turnLeft();
28 while (isClearAhead()) {
29     moveForward();
30 }
31 turnLeft();
32 while (isClearAhead()) {
33     moveForward();
34 }

```

12. Loops #12

Solution:

```

1  turnAround();
2  while (isClearAhead()) {
3      moveForward();
4  }
5  turnRight();
6  while (isClearAhead()) {
7      moveForward();
8  }
9  turnRight();
10 while (isClearAhead()) {
11     moveForward();
12 }
13 turnAround();
14 moveForward();
15 turnLeft();
16 while (isClearAhead()) {
17     moveForward();
18 }
19 turnLeft();
20 moveForward();
21 pullLever();
22 turnAround();
23 while (isClearAhead()) {
24     moveForward();
25 }
26 turnRight();
27 moveForward();

```

13. Loops #13

Solution:

```

1  turnRight();

```

```

2 moveForward();
3 turnLeft();
4 moveForward();
5 turnRight();
6 while (isClearAhead()) {
7     moveForward();
8 }
9 pullLever();
10 turnAround();
11 moveForward();
12 turnLeft();
13 while (isClearAhead()) {
14     moveForward();
15 }
16 turnRight();
17 while (isClearAhead()) {
18     moveForward();
19 }
20 turnLeft();
21 moveForward();
22 pullLever();
23 turnAround();
24 moveForward();
25 turnRight();
26 while (isClearAhead()) {
27     moveForward();
28 }
29 turnRight();
30 while (isClearAhead()) {
31     moveForward();
32 }
33 turnRight();
34 while (isClearAhead()) {
35     moveForward();
36 }

```

14. Loops #14

Solution:

```

1 turnAround();
2 while (isClearAhead()) {
3     moveForward();
4 }
5 turnRight();
6 moveForward();
7 moveForward();
8 turnRight();
9 while (isClearAhead()) {
10    moveForward();
11 }
12 pullLever();
13 turnLeft();
14 moveForward();
15 moveForward();
16 turnLeft();
17 while (isClearAhead()) {
18    moveForward();
19 }

```



```
20 pullLever();
21 turnRight();
22 moveForward();
23 moveForward();
24 turnRight();
25 while (isClearAhead()) {
26     moveForward();
27 }
```

15. Loops #15

Solution:

```
1 for (var count = 0; count < 4; count++) {
2     moveForward();
3     pullLever()
4     moveForward();
5 }
6 moveForward();
```