

## Sequencing and Algorithms

### Textbook

## Sequencing and Algorithms



Have you ever built something with LEGO bricks? The box and instructions give you a picture of what you want to build, but if you don't follow the steps in the right order, the castle you build might turn out looking like a giant tower instead! Just like building with LEGO, algorithms are all about following steps in the right order to make sure everything fits together correctly. Let's explore how sequencing and algorithms help us put things together, whether it's a LEGO castle or a computer program!



**Sequencing** is the order in which things happen. It's the way we organize steps to make sure something works. For example, think about making chocolate chip cookies:

1. **Buy the ingredients**
2. **Mix the ingredients** to make cookie dough
3. **Put small balls of dough on a cookie sheet**
4. **Put the cookie sheet in the oven**
5. **Let the cookies cool** and then enjoy!

These steps must be followed in the right order. If you mix them up, you won't get the delicious cookies you were hoping for! For example, what if you put the cookie sheet in the oven before mixing the dough? You'd have an empty sheet in the oven instead of cookies!

## Correct Sequence and Algorithms

An **algorithm** is like a recipe or a set of instructions that tells you how to do something in the correct order. When you follow the steps to make cookies, you're following an algorithm. If you follow the right sequence, you end up with tasty cookies!

In computer programming, algorithms are super important too. A computer follows algorithms to do things, like when you press "play" on a video. The computer follows an algorithm to know when to start the video. Everything a computer does is based on algorithms—step-by-step instructions that tell it what to do!

## Algorithms and Computer Programming



In programming, you're building algorithms that tell the computer how to do something. For example, when playing a puzzle game, you choose the right steps to move your character to the goal. Each command you give the computer is part of an algorithm, and if you get the sequence right, the computer will do exactly what you want.

# Programming Environment

We've seen how algorithms are like recipes or building instructions for computers. They tell the computer exactly what to do, step by step. But before a computer can even *start* following those steps, it needs to get ready!

Think back to building with LEGO bricks:

- You need **the right place to build** – maybe a big, clear table, not a messy floor!
- You need **all your LEGO bricks** sorted and ready to go.
- You need **the instruction booklet** handy.

Imagine trying to build your amazing LEGO castle in a dark closet, or without any bricks, or without the instructions! It just wouldn't work, right? You need a special "setup" or "environment" to build successfully.

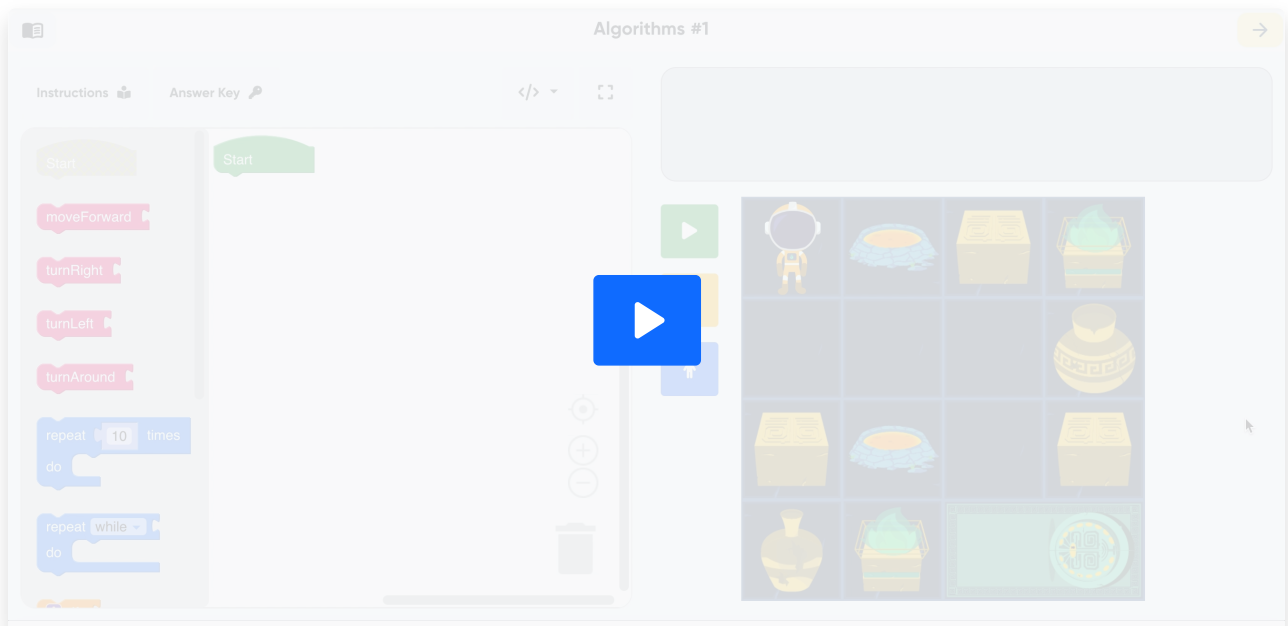
It's the same for computers! When we write algorithms for a computer, we don't just type commands anywhere. The computer needs a special starting place where it can understand our instructions. This special place is called a **program environment**.

A program environment is like:

- **The computer's special "workshop":** It's where the computer gets ready to build your program, with all its tools and parts.
- **The right "tools":** Just like you need specific LEGO pieces, the program environment has special "tools" (like commands and settings) that the computer needs to understand your algorithm. It helps the computer speak the same "language" as your code.
- **An organized space:** It makes sure everything is neat and in order so the computer doesn't get confused by your instructions.

So, whenever we want a computer to follow our amazing algorithms, we first need to make sure we're in the correct **program environment**. It's like opening the right app on a tablet, or going to a special website designed for coding. This way, the computer knows exactly how to get itself ready and follow your step-by-step instructions!

## Sequencing and Algorithms Video



## Practice Building Algorithms



In the puzzle game, you will have different options to move your character. These are different [JavaScript](#) commands. JavaScript is a programming language.

### Options to Move Your Character

Blocks	Syntax
Start	1 <code>moveForward();</code>
moveForward	2 <code>turnRight();</code>
turnRight	3 <code>turnLeft();</code>
turnLeft	4
	5

Choose from these commands to direct the character to the carpet in front of the door . Make sure to order the commands in the correct sequence to build an algorithm that will work. You can use commands more than once.

If programming with [syntax](#), you will notice there are semicolons ( ; ) after each command. The semicolon tells the computer that is the end of your command.

Remember that these movements are **from the character's point of view**. So `turnRight();` means that according to the character, they will turn right. This will take some thinking to make sure the sequence is in the right order.

### Critical Thinking Questions

1. Why is it important to follow the steps of an algorithm in the correct order?
2. Can you think of an example where sequencing is important in a game or sport?

## Questions (5)

### 1. What is an algorithm?

MULTIPLE CHOICE

Choose the correct answer:

- A. A list of step by step instructions
- B. A programming language
- C. A musical term for the pattern of sound
- D. A save button

### 2. True or False: The order of an algorithm doesn't matter.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

### 3. True or False: An algorithm is a sequence that is in the correct order.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

### 4. What does the semicolon ( ; ) at the end of each command in JavaScript mean?

MULTIPLE CHOICE

Choose the correct answer:

- A. It tells the computer to move the character forward.
- B. It indicates the end of a line of code.
- C. It tells the character which direction to turn.
- D. It tells the starting point of the character.

### 5. True or False: JavaScript is a programming language.

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

## Games (2)

### 1. Sequencing and Algorithms Ordering Game

Put the steps in the right sequence to make chocolate chip cookies.

Full Screen

Audio

Instructions

Answer Key

Pause

Clear All

Check Matches

Attempts: 0

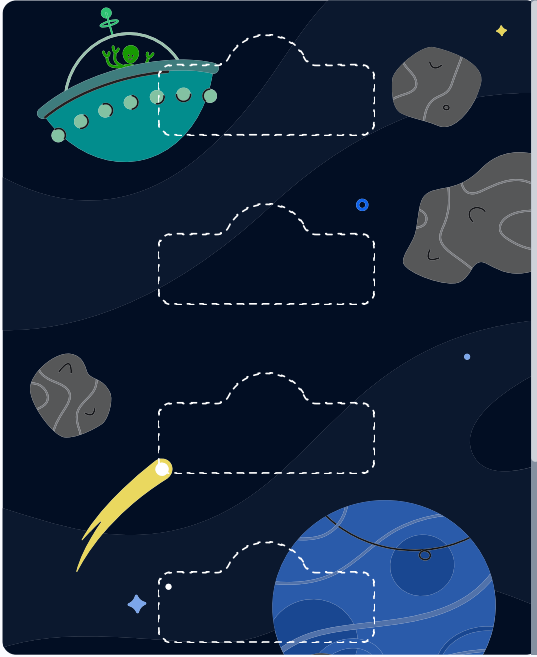
Mix the ingredients to make cookie dough



Buy the ingredients

Put the cookie sheet in the oven

Put small balls of dough on a cookie sheet

When they are finished, let them cool, and enjoy!





## 2. Sequencing and Algorithms Typing Game

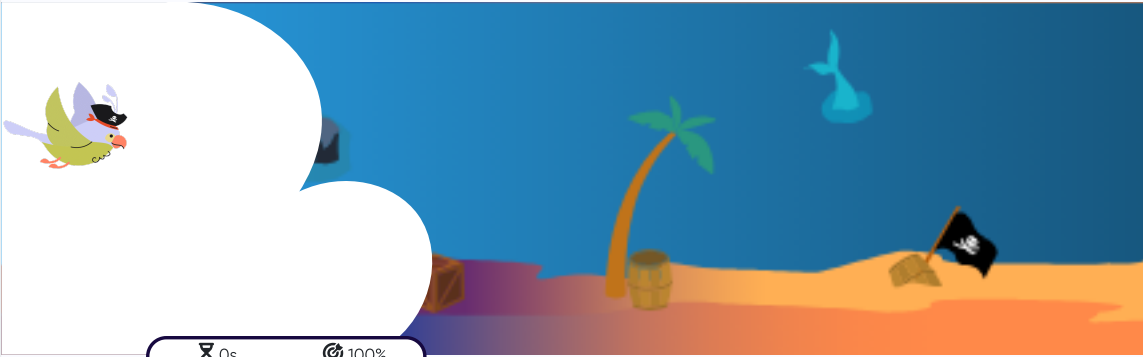
Full Screen

Audio

Instructions

Restart

Pause



0s100%

Almost everything your co

# Puzzles (10)

## 1. Algorithms #1

Instructions

Answer Key

</>


## 2. Algorithms #2



Instructions

Answer Key

</>




### 3. Algorithms #3

Instructions  Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead

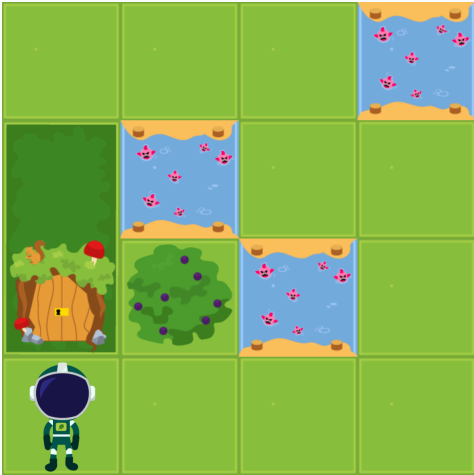
isClearLeft

isClearRight



Start



### 4. Algorithms #4

Instructions  Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead

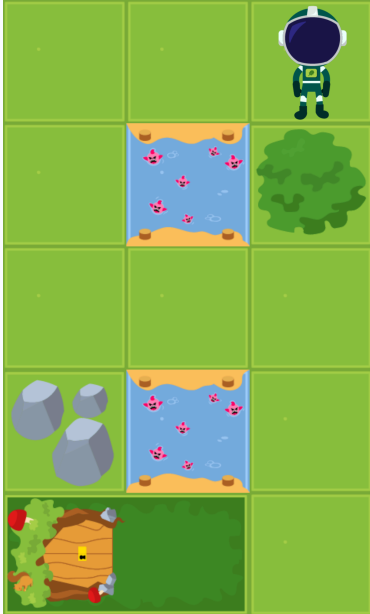
isClearLeft

isClearRight

Start



## 5. Algorithms #5

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

Start

▶

↗

↑

⏻

## 6. Algorithms #6

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

Start



▶

↗

↑

⚙


## 7. Algorithms #7

Instructions  Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

end




not





isClearAhead










isClearLeft

isClearRight



Start

## 8. Algorithms #8

Instructions  Answer Key 

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do




end





not














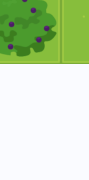
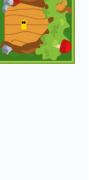
isClearAhead

isClearLeft

isClearRight

## 9. Algorithms #9

Instructions 
Answer Key 
</>

Start
Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

## 10. Algorithms #10

Instructions

Answer Key

</>

Start

moveForward

turnRight

turnLeft

turnAround

repeat 10 times

do

repeat while

do

if

do

and

not

isClearAhead

isClearLeft

isClearRight

Start

▶

↗

↑

⏏

## Answer Keys & Solutions

### Questions

#### 1. What is an algorithm?

MULTIPLE CHOICE

Correct Answer:

- A. A list of step by step instructions ✓ Correct
- B. A programming language ✗ Incorrect
- C. A musical term for the pattern of sound ✗ Incorrect
- D. A save button ✗ Incorrect

#### Explanation:

You must follow these in order.

#### 2. True or False: The order of an algorithm doesn't matter.

MULTIPLE CHOICE

Correct Answer:

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

#### Explanation:

Should steps be followed in order?

#### 3. True or False: An algorithm is a sequence that is in the correct order.

MULTIPLE CHOICE

Correct Answer:

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

### Explanation:

Do algorithms need to happen in order?

## 4. What does the semicolon ( ; ) at the end of each command in JavaScript mean?

MULTIPLE CHOICE

### Correct Answer:

- A. It tells the computer to move the character forward. ✗ Incorrect
- B. It indicates the end of a line of code. ✓ Correct
- C. It tells the character which direction to turn. ✗ Incorrect
- D. It tells the starting point of the character. ✗ Incorrect

### Explanation:

Think about the purpose of the semicolon in JavaScript syntax and its role in writing correct code.

## 5. True or False: JavaScript is a programming language.

MULTIPLE CHOICE

### Correct Answer:

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

### Explanation:

Think about what JavaScript is commonly used for when interacting with websites.

## Games

### 1. Sequencing and Algorithms Ordering Game

#### Correct Order:

1. Buy the ingredients
2. Mix the ingredients to make cookie dough
3. Put small balls of dough on a cookie sheet
4. Put the cookie sheet in the oven
5. When they are finished, let them cool, and enjoy!

**Scoring:**

- Gold: 1 attempts or fewer
- Silver: 2 attempts or fewer
- Bronze: 3 attempts or fewer

*Students must arrange items in the correct sequence.*

## 2. Sequencing and Algorithms Typing Game

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

### Puzzles

#### 1. Algorithms #1

**Solution:**

```
1 moveForward();  
2 moveForward();  
3 moveForward();  
4 turnLeft();  
5 moveForward();
```

#### 2. Algorithms #2

**Solution:**

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

#### 3. Algorithms #3

**Solution:**

```
1 turnLeft();  
2 moveForward();  
3 moveForward();  
4 moveForward();  
5 turnLeft();
```

```
6 moveForward();
7 moveForward();
8 turnLeft();
9 moveForward();
10 turnRight();
11 moveForward();
12 turnLeft();
13 moveForward();
14 moveForward();
15 turnLeft();
16 moveForward();
```

## 4. Algorithms #4

**Solution:**

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

## 5. Algorithms #5

**Solution:**

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

## 6. Algorithms #6

**Solution:**

```
1 turnLeft();
2 moveForward();
3 turnLeft();
4 moveForward();
5 moveForward();
```



```
6 turnLeft();
7 moveForward();
8 turnRight();
9 moveForward();
10 moveForward();
11 turnRight();
12 moveForward();
13 moveForward();
14 moveForward();
15 moveForward();
16 turnRight();
17 moveForward();
18 moveForward();
19 moveForward();
```

## 7. Algorithms #7

**Solution:**

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

## 8. Algorithms #8

**Solution:**

```
1 turnLeft();
2 moveForward();
3 moveForward();
4 moveForward();
5 turnRight();
6 moveForward();
7 turnLeft();
8 moveForward();
9 turnRight();
10 moveForward();
11 moveForward();
```

```
12 turnRight();
13 moveForward();
14 moveForward();
15 turnLeft();
16 moveForward();
17 moveForward();
18 turnLeft();
19 moveForward();
20 moveForward();
```

## 9. Algorithms #9

**Solution:**

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

## 10. Algorithms #10

**Solution:**

```
1  turnAround();
2  moveForward();
3  moveForward();
4  moveForward();
5  turnLeft();
6  moveForward();
7  turnRight();
8  moveForward();
9  moveForward();
10 turnLeft();
11 moveForward();
12 moveForward();
13 turnLeft();
14 moveForward();
```

```
15 turnRight();  
16 moveForward();
```