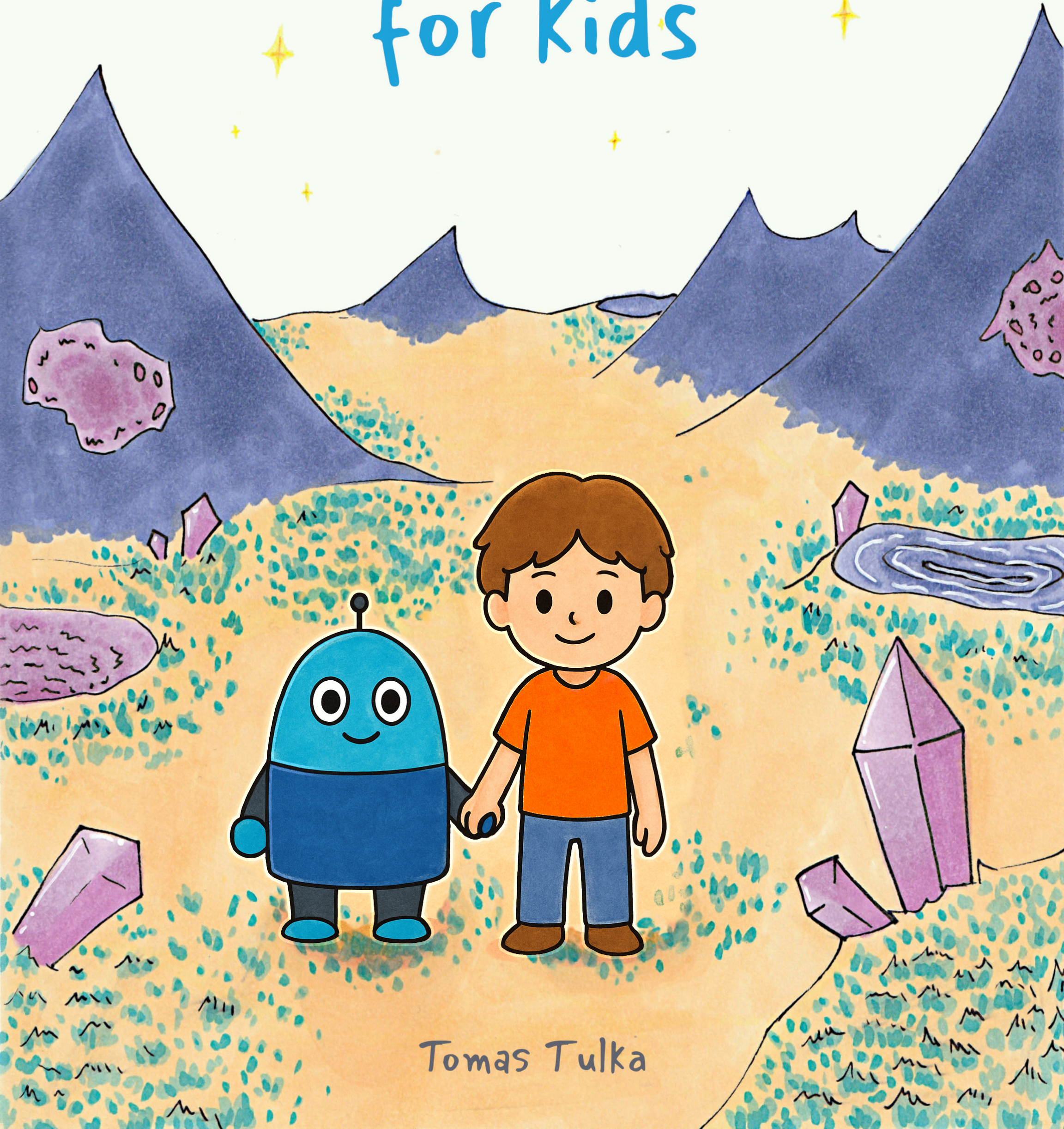
# Computer Science

# for Kids



Tomas Tulka

# Computer Science for Kids

## Discover how computers work and learn the power of programming!

Tomas Tulka

This book is available at
https://leanpub.com/computerscienceforkids

This version was published on 2025-10-31

Leanpub

# Contents

# Introduction

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# A message from space — *The story begins*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# How to talk with robots

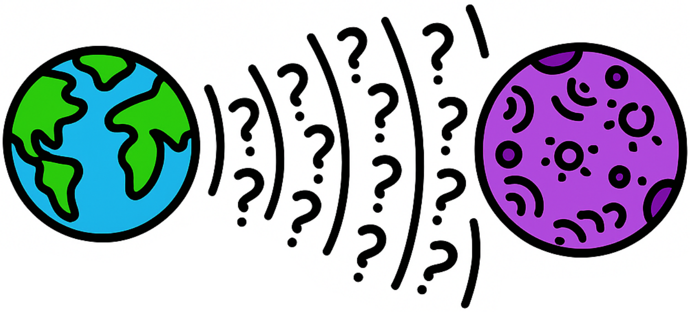
This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Electronic words — *Digital communication*

A rocket as white as moonlight carrying a small robot departs from the home planet. The earth at first looks like a deep ocean with flat islands through its round windows, then like a blueberry pie with whipped cream, and finally like a bluish smudge the size of a walnut. The stars are as bright as cat's eyes. The rocket wobbles as it passes through the cloud of stardust, but the robot remains calm. The goal is already within reach.

Comet Kohoutek II is approaching the solar system at supersonic speeds. It begins to warm up in the bright rays of the sun, and its tail made of escaping vapors glows beautifully.

Comets have long orbits. They fly through distant and unexplored parts of space before returning to the same place. It can take hundreds or even thousands of years. Choosing to stay on a comet is akin to boarding an interstellar express train. Staying on it for too long may mean missing the final stop for the return. You must act quickly.



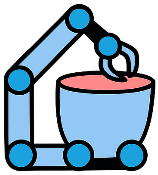
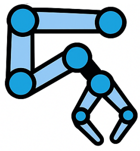
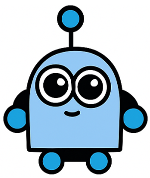
You turn the green lever on the control panel to the “Start” position. Something inside hums, flashes, and falls silent again. Even on the second try, nothing happens. The panel is clearly damaged. It may have broken up in flight through the atmosphere or on impact with the ground. Anyway, it is not working.

What now? The control panel was the only way to speak to the lost robot. If you want to fix it, you must first understand how such communication works.

## Universal robots

Who has not seen a kitchen robot kneading dough? You have probably also seen robots robots vacuuming the floor or even the large robots that assemble cars in a factory. A robot is simply a collection of components needed for its work, such as a kneader, wheels, a vacuum cleaner, or a mechanical arm.

> **Universal robots** (such as those used in research missions) differ from kitchen robots in that they are not specialized for a specific type of work (such as kneading dough), but instead can solve a wide variety of tasks ranging from kneading dough to calculating your math homework.

The mind of a universal robot is a powerful computer that controls everything else. Because this computer is a form of robot brain, robots are in fact just big computers. The robot can function without arms and legs, but not without its central computer. Without it, it is just a bunch of useless spare parts.

**Challenge #1**

A similar universally handy creature already exists in nature — can you guess what it is?

(You can find the solution to this and all other challenges at the end of the book.)

# Signals

How do you get your hand to scratch your ear? You just think about it and the thought is then transferred from the brain to the hand in the form of nerve signals.

Robots are not very different from humans in this way. When a robot wants to move its hand, its brain, the central computer, sends an electronic signal to the mechanical hand, which then rubs its ear (if robots had ears).

> **Signals** are the basic elements of communication, and that's why they must be very simple: *a signal is either present or it is not* - nothing could be simpler!

Think of signals as just a series of simple events. For example, a light bulb flickering or tapping on the wall. Do you know Morse code? It also uses such signals. The ancient Indians sent smoke signals by covering and uncovering a fire. A flash, a puff of smoke, or tapping are all signals.

### For eager beavers

Signals are carried either by cables and wires or through the air. Cables usually use some form of electrical current or light (yes, some cables actually use light - these are known as fiber-optic cables).

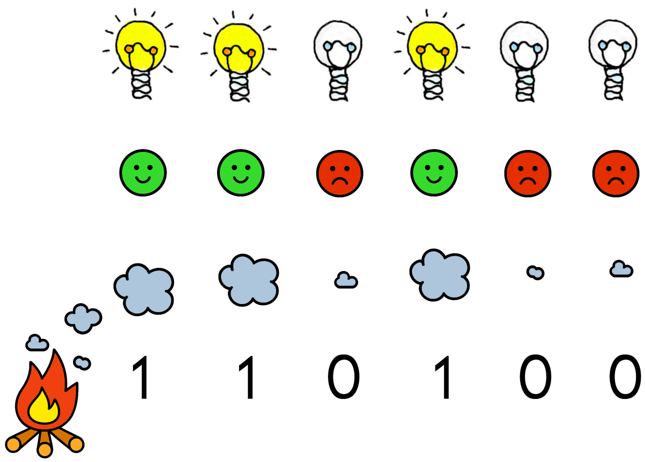
Signals are sent through the air using electromagnetic waves, which you can think of as a type of light that cannot be seen with the naked eye. Antennas and flat satellites on house roofs receive electromagnetic waves.

# Coding

Signals can also be written on paper using symbols. A blank space is written where the flash, puff or tap are missing. Morse code, for example, uses dots to represent a signal and dashes to represent empty spaces where there is no signal.

Robots use **zeros and ones**. Nothing, however, prevents you from writing down signals, such as using two different smileys. Sadly, robots do not have much of a sense of humor.

You will also soon learn that their favorite ones and zeros have one major advantage: they can be *computed* with. And since robots are computers, computing is their favorite hobby.




The process of writing signals using zeros and ones is called **coding** or **digitization**, and the written signal is known as a *digital* or *numerical code*. The code `110100`, for example, is the digitization of the following tapping on the wall: *tap*, *tap*, *pause*, *tap*, *pause*, *pause*.

### Challenge #2

Try tapping this digital code on your table: `10101101`.

Ask someone from your family or friends to try to write down your sound signals again as a digital code (using ones and zeros).

Digitization is the process of converting signals from the *imperfect physical world of humans* (light, sound, electricity) to the *perfect world of computers* and numbers (ones and zeros).

Such a process is not at all easy, because a bulb can flash stronger at times, and weaker at others. The same is true for tapping and electricity. As a result, the digitized signal is free of inaccuracies and much easier to work with than the original imperfect physical signal.

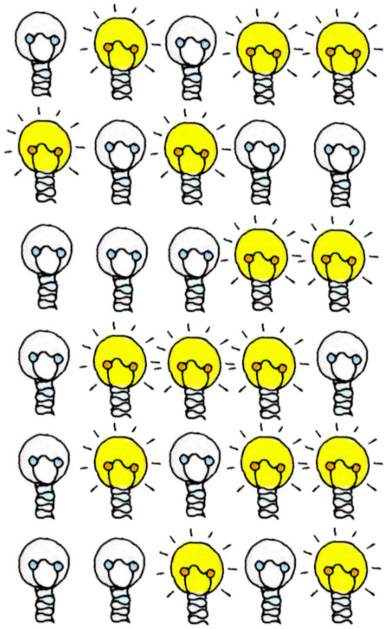
### Challenge #3

Compare the following electrical signal and its digital code for yourself. Note that the ones and zeros in the physical signal differ slightly from each other, whereas they are exactly the same in the digital code:


Once you know how electronic communication works, it is easy to figure out why the control panel does not work: the light bulb broke on landing and the electrical circuit was interrupted.

You dash to your room, unscrew the bulb from the lamp on the desk and run back. You carefully replace the broken bulb and turn the green lever on the control panel back to the “Start” position.

The panel quietly clicks, and the bulb flashes in the following order of signals:



## Challenge #4

Can you encode this light signal using zeros and ones as a digital code?

You have just received the first message from a distant world. You now know how to talk to the lost robot! But what do all those ones and zeros mean? Unfortunately, the second step of the control manual will not be of much help to you:

```
Step 2. Read the message
```

Luckily, even this is not an impossible task!

### For eager beavers

Coding is not just a special right or privilege of machines. Living organisms use the DNA molecular code, in which the instructions for building cells and tissues are encoded. Everything about you is encoded in your DNA.

## Commands

Signals are like letters that make up words. But how can a word be formed from only two symbols (`0` and `1`) when the human alphabet contains many more?

> A digital code consisting of only two values (`0` and `1`) is called **binary code** (in Latin "*binarius*" means two). Two letters can only be used to form two words. For example, `0` can mean "*off*" and `1` "*on*".

Two words may be enough for some simple robots, but definitely not for a universal robot. So how do you express more than two words in binary code? The solution is simple: *longer code*.

Like in human speech, the more you want to say, the longer your story must be. Using a code of length three allows you to create more words than using a code of length one.

For example:

| **code** | **command** |
|---|---|
| `000` | turn off |
| `001` | turn on |
| `010` | sing |
| `011` | dance |
| `100` | make a sandwich |
| `101` | take out trash |
| `110` | clean a room |
| `111` | calculate math homework |

As you can see, only *three digits* of binary code are used to generate eight different commands. And it does not take anything more than zeros and ones!

More than eight commands will simply require slightly longer code, and you can continue like this forever and ever. The longer the code, the more commands you can create from it.

> Individual code characters are called **bits** ("*binary digits*"). So the previous encoding had three bits, and now it will work with eight bits. A group of eight bits is often called one **byte** or one *octet* (in Latin "*octo*" means eight).

Robots usually count *from zero*, that is, 0, 1, 2, 3, and so on.

# Binary alphabet

Coding single commands is fine at the start, but as more and more commands are needed, it becomes hard to keep adding new ones and learning them again and again. Instead, you can use something more flexible—like the human alphabet (A to Z). Every word is made from a mix of letters that never change. So, when you need a new command, you can simply spell it with letters you already have. This means letters are a *universal* way to make codes.

Now, you just need to turn the letters into signals. To keep things in order, you can give each letter a number. Because robots start counting from zero, the letter “A” gets number 0, “B” gets 1, “C” gets 2, “D” gets 3, and so on:

| | | | | | |
|---|---|---|---|---|---|
| A | 0 | J | 9 | S | 18 |
| B | 1 | K | 10 | T | 19 |
| C | 2 | L | 11 | U | 20 |
| D | 3 | M | 12 | V | 21 |
| E | 4 | N | 13 | W | 22 |
| F | 5 | O | 14 | X | 23 |
| G | 6 | P | 15 | Y | 24 |
| H | 7 | Q | 16 | Z | 25 |
| I | 8 | R | 17 | | |

You can now turn any word into numbers:

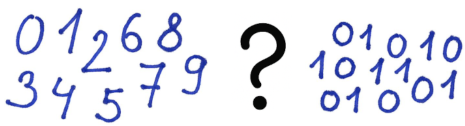
H $\Rightarrow$ 7
E $\Rightarrow$ 4
L $\Rightarrow$ 11
L $\Rightarrow$ 11
O $\Rightarrow$ 14

HELLO $\Rightarrow$ 7 4 11 11 14

### Challenge #5

Write your name as numbers using the table above.

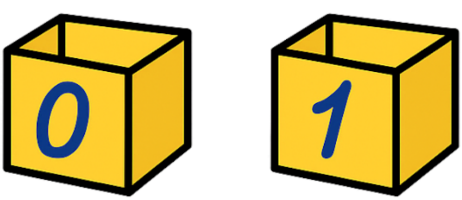
This works great! The next step is to change those numbers into binary code, which uses only zeros and ones.



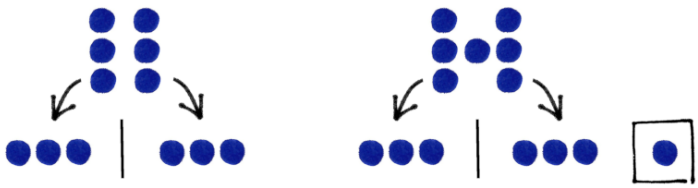
So far, you've used normal numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, and 0. This type of code is known as **decimal** (in Latin "*decem*" means ten). For example, the number one hundred and twenty-three is written as 123. There is nothing special about the decimal code; it was simply chosen so that people could easily count on their ten fingers.

You can count with your fingers only up to ten. After ten, you'd need more fingers! In code, this means adding one more bit. For example, 9 (one bit) plus 3 (one bit) is 12 (two bits for "1" and "2").

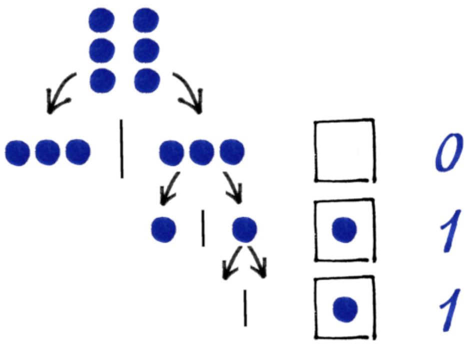
When you want to write a number in binary code, you must squeeze its value into two boxes—a box for zeros and a box for ones:



Luckily, this is not hard! You just divide the number by two and keep track of what's left over. You can imagine this using marbles. If you have six marbles, you can split them into two piles with three marbles each. If you have seven marbles, you can still make two piles of three, but one marble will be left over—that's the **remainder**:

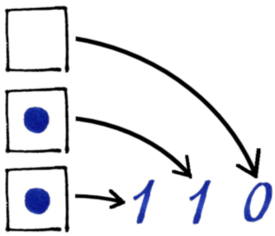


The remainder is always zero or one, because if it were two or more, you could still divide it again. So, the remainder is actually binary! If you keep dividing the piles until you can't anymore, you'll end up with a column of remainders that form the binary code:




At the end, each pile has only one marble left. That last marble is also a remainder, because one divided by two is zero—so make sure to count it too!

Finally, read all the remainders from bottom to top (like tipping the column over to the right), and you'll get the binary code for your number:




That's it! The number six is `110` in binary code.

Instead of marbles, you can also use the numbers themselves:

6 ÷ 2 = 3 + [**0**]
3 ÷ 2 = 1 + [**1**]
1 ÷ 2 = 0 + [**1**]

### Challenge #6

Write numbers 2, 7, and 10 in binary code.

Now you know how to turn letters into binary code. But you still need to separate the letters in a word so you can tell whether "11" means the number 11 or two numbers, 1 and 1. To do that, you could use something to separate them, like a space or a comma. But spaces and commas are another symbols, and you only have two—`0` and `1`. Hmm, you'll need a new idea.

Because the last letter ("Z") has the order number 25, which in binary is `11001` (five bits), you can use five bits for every letter to tell when one letter ends and another begins. That way, every five bits is one letter.

Adding zeros to the beginning of a number doesn't change it, so 6 is the same as 06 or 00006. For example, the binary number `110` becomes `00110` when written with five bits, and two 1's become `0000100001` (each 1 written as `00001` and joined together):


After a bit of work, you complete the full table for binary coding:

| | | | | | |
|---|---|---|---|---|---|
| A | 00000 | J | 01001 | S | 10010 |
| B | 00001 | K | 01010 | T | 10011 |
| C | 00010 | L | 01011 | U | 10100 |
| D | 00011 | M | 01100 | V | 10101 |
| E | 00100 | N | 01101 | W | 10110 |
| F | 00101 | O | 01110 | X | 10111 |
| G | 00110 | P | 01111 | Y | 11000 |
| H | 00111 | Q | 10000 | Z | 11001 |
| I | 01000 | R | 10001 | | |

Now you can turn any word into binary code—and back again:

H ⟹ `00111`
E ⟹ `00100`
L ⟹ `01011`
L ⟹ `01011`
O ⟹ `01110`

HELLO ⟹ `0011100100010110101101110`

## Challenge #7

Write your name in binary code.

You're now ready to decode the message from the lost explorer!

## Challenge #8

Check the code you got earlier on the control panel. Use the table above to find each letter and write them next to each other.

The result is the name of the robot in trouble.

# Mechanical tongue — *Programming languages*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Compilers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Programs

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Bright shiny eyes — *Display devices*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Pixels

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Colors

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Silicon mind — *What computers are made of*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Computers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Integrated circuits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Input and output

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Memory

# How to program robots

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Strange world — *Getting started with programming*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Programming languages

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## LudolfC

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# A little bit of everything — *Data types*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Numbers

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Text

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Truthy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Composite data types

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# This and that — *Variables*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What are variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to write variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Variable names

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Values of variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Using variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Step by step — *Arrays*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What are arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to write arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Multidimensional arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to work with arrays

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/comput

erscienceforkids.

# Size of an array

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Array variables

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# One way or another — *Conditionals*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What are conditionals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to write conditionals

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What else?

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Else with a condition

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Round and round — *Loops*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What is a loop

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to write loops

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Nested loops

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# On call — *Functions*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What are functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to write functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to call functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Functions without output

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/comput

erscienceforkids.

# Functions without parameters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Using functions

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# The scope of the function

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# I, robot — *Objects*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## What are objects

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to write objects

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## How to work with objects

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Using objects

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/comput

erscienceforkids.

# Everything is an object

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Back home

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# The way out — *Solution*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## First attempt

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Over craters

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## From the beginning

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Energy

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/comput

erscienceforkids.

# A better solution

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# To the rocket

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Exercise

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Goodbye! — *Epilogue*

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Solutions

## Challenge #1

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #4

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #6

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #8

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #9

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #10

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #11

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #12

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #13

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #14

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #16

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #17

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #18

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #19

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #20

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #21

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #22

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #23

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #24

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #25

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #26

## Challenge #27

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #28

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #29

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #30

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #31

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #32

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #34

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #35

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #36

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #37

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #38

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #39

## Challenge #40

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #41

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #42

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

## Challenge #43

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #44

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #45

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #47

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #48

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Challenge #54

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.

# Credits

This content is not available in the sample book. The book can be purchased on Leanpub at https://leanpub.com/computerscienceforkids.