How to Think in Code

Workshop Lead: Larisa Morales Soto

Date: February 2, 2024

McGill initiative in Computational Medicine

Mission statement: deliver quality workshops designed to help biomedical researchers develop the skills they need to succeed.

Location: 740 Dr. Penfield Avenue, Montreal, Quebec

Scan the QR code to sign up for our **mailing list**

Contact: workshop-micm@mcgill.ca

Winter 2024 MiCM Workshops Series

Sign up for our mailing list for updates!

Workshop	Date	Location	Registration
How to think in Code	Feb. 2 9AM-11AM	McIntyre Room 325	Open
Intro to UNIX and HPC	Feb. 7 9AM-1PM	McIntyre Room 325	Open
Git and GitHub	Feb. 9 1PM-5PM	Arts Room 150	Open
Intro to R (Part 1)	Feb. 14 9AM-1PM	Macdonald Engineering Building Room 10	Open
Data Analysis in R (Part 2)	Feb. 16 9AM-1PM	McIntyre Room 325	TBA
Intro to Python (Part 1)	Feb. 21 1PM-5PM	TBA	TBA
Data Analysis in Python (Part 2)	Feb. 23 1PM-5PM	Arts Room 150	TBA
Transcriptomics	TBA	TBA	TBA
Metagenomics	Mar. 13 10AM-12PM	TBA	TBA
WGS Data and Variant Calling	TBA	TBA	TBA
GWAS and PRS	TBA	TBA	TBA

https://www.mcgill.ca/micm/training/workshops-series

What to expect today

✓ Practice structured thinking

✓ Become familiar with programming logic

✓ Learn about basic programming components (variables, algorithms, data structures)

We will NOT:

- Learn a specific programming language
- Perform data analysis
- Learn about machine learning
- Write programs that generate visualizations

Contents

- Part 1 - Programs
  - Instructions
  - Recognizing patterns
  - Pseudocode – control structures
  - Algorithms
- Part 2 – Coding fundamentals
  - Boolean logic
  - Conditional statements
  - Variables
  - Data structures
  - Programming languages

Exercises:
Create an account

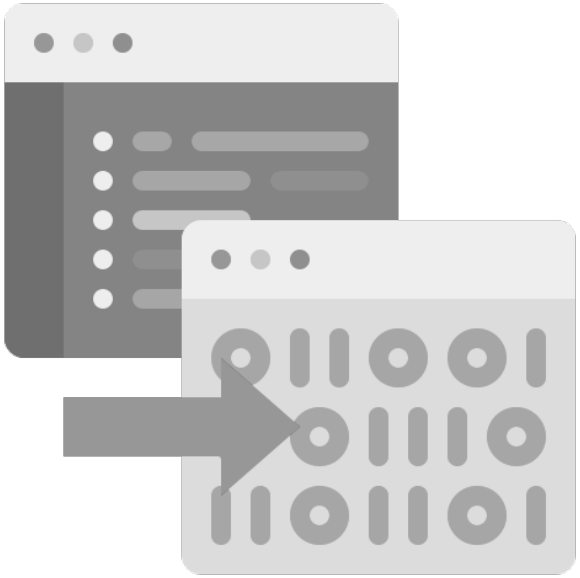
BRILLIANT

https://brilliant.org

Part 1
Programs

What is a computer program?

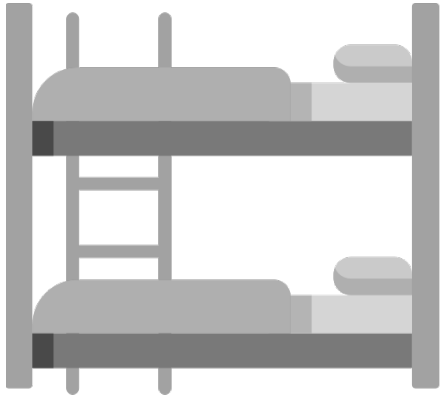
Source code
Human readable



Executable
Machine instructions

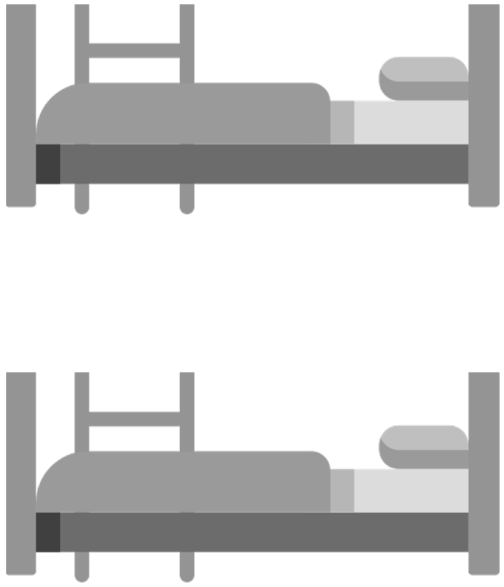
- Sequence of instructions for the computer to execute
- Translation of our thoughts into operations that the computer understands

Sequence of instructions



- Clear
- Concise
- Efficient

Repeating instructions



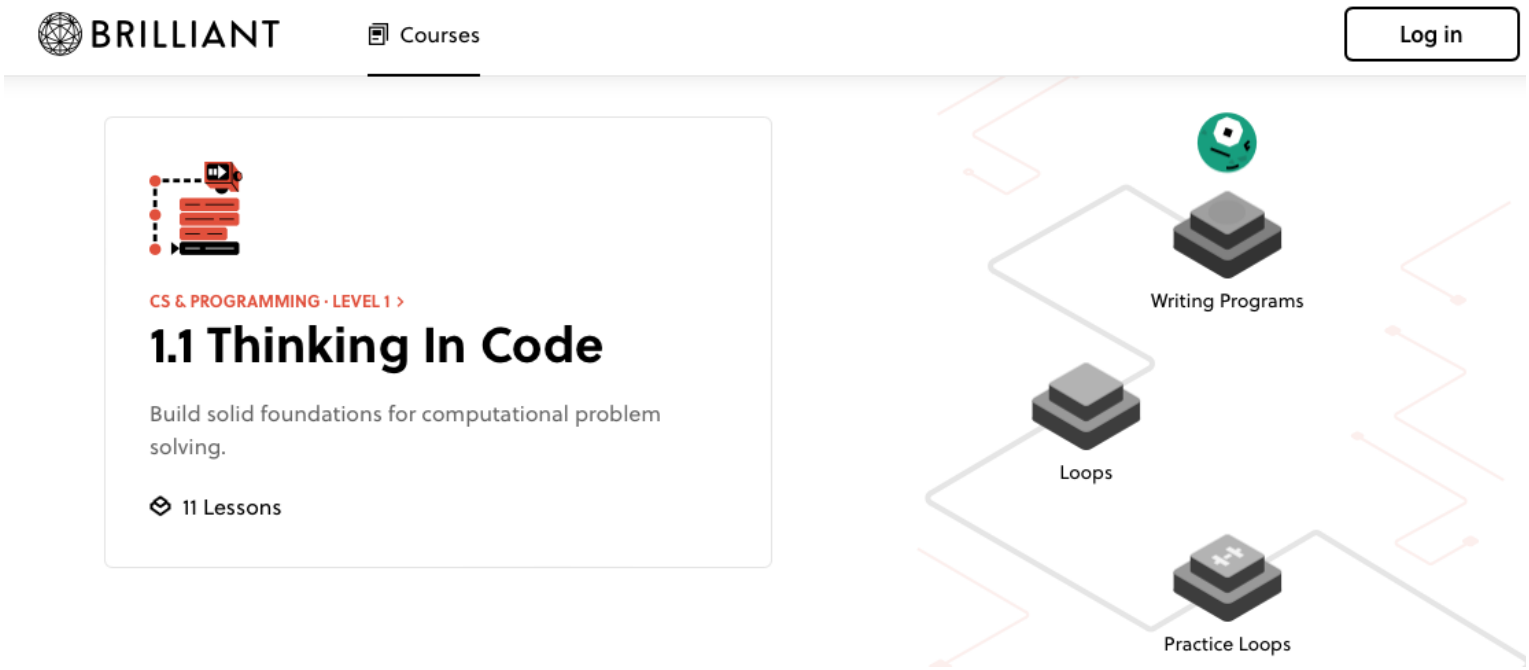
- Learn how to do it once
- Find the pattern
- Repeat it

Pseudocode

A high-level summary of a sequence of instructions

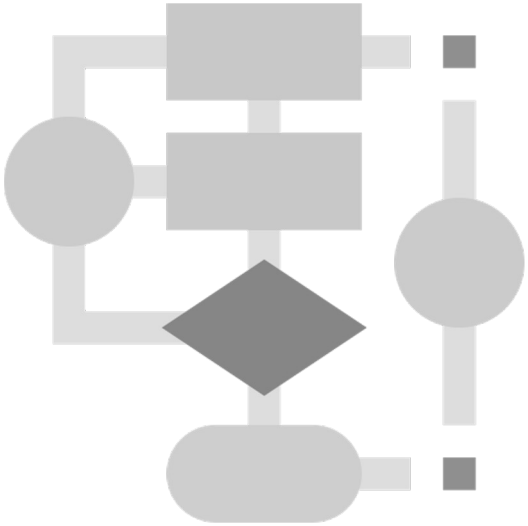
- Useful to visualize the order of steps
- Serves as the backbone of the code

Hands on 1



https://brilliant.org/courses/thinking-in-code/

Algorithms



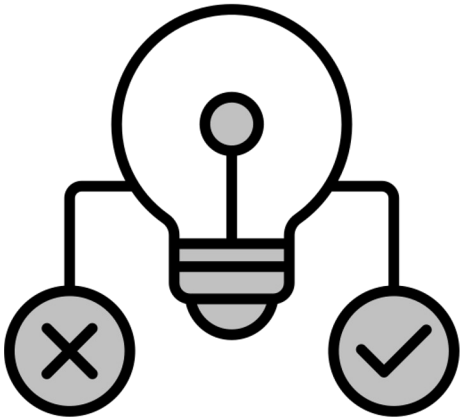
Precise set of instructions used for solving a problem or performing a task.

- Tying your shoes
- Finding a book in the library
- Following a cooking recipe

- Finding the shortest path
- Sorting a list of numbers
- Recommending digital content

Part 2
Coding fundamentals

Boolean logic

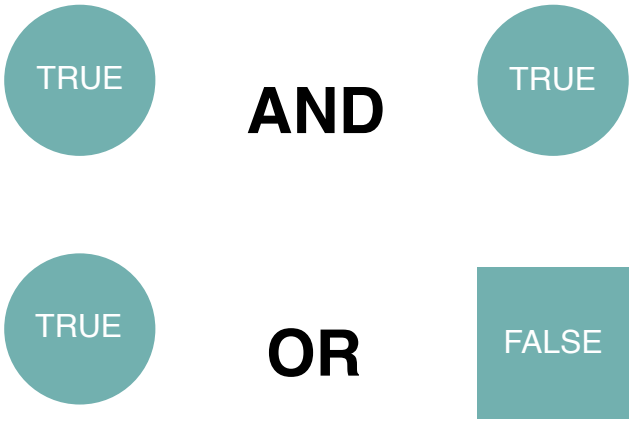


A type of operation that can only have two possible outcomes: **TRUE** or **FALSE**

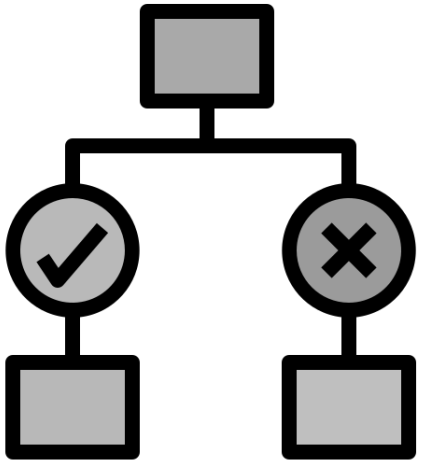
Boolean expression	Result
2+2 == 5	FALSE
“dog”==“human”	FALSE
2*2 > 1	TRUE

Boolean operators

Shape==circle



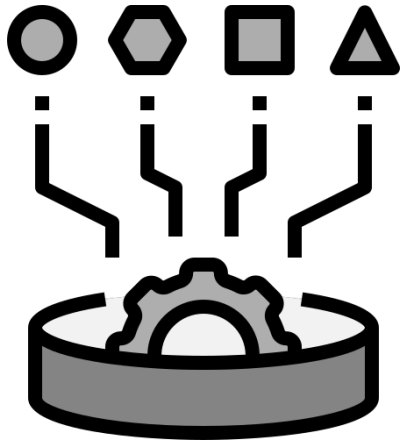
Conditionals



Command used to make a decision and specify what to do in each case

```
If (boolean condition) Then
   (consequent)
Else
   (alternative)
End If
```

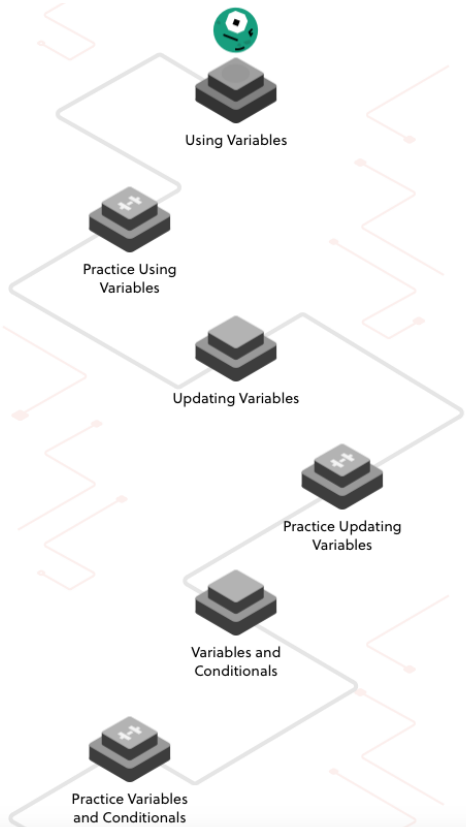
Variables



Container to
store information
needed in the program

shape = “circle”

Hands on 2

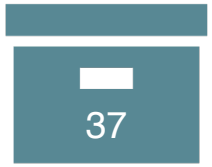


https://brilliant.org/courses/creative-coding/

Data types

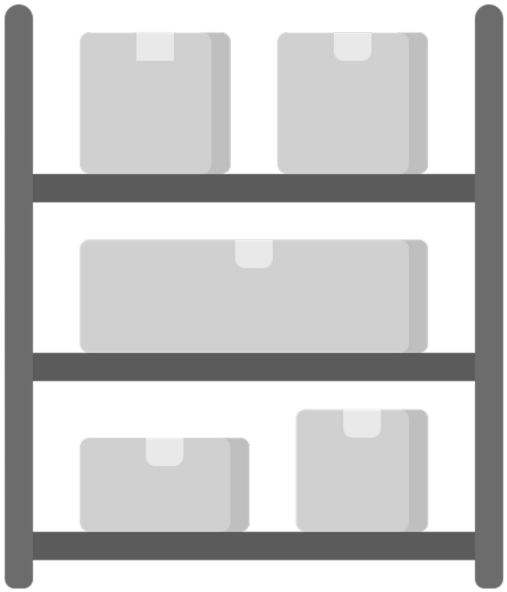
Refers to the kind of information that can be stored in a variable

Data Type	Represents	Examples
integer	whole numbers	`-5` , `0` , `123`
floating point (real)	fractional numbers	`-87.5` , `0.0` , `3.14159`
string	A sequence of characters	`"Hello world!"`
Boolean	logical true or false	`true` , `false`
nothing	no data	`null`



Adapted from https://press.rebus.community/programmingfundamentals/chapter/data-types/

Data structures



Specific storage formats designed for efficient data access and management

In summary

✓ Practice structured thinking

✓ Become familiar with programming logic

✓ Learn about basic programming components

Now you are ready to:

- Learn a specific programming language
- Perform data analysis
- Learn about machine learning
- Write programs that generate visualizations

But before coding…

Understand the problem

Recognize patterns

Think about the solution

Write a pseudocode

Make it work
Make it right
Make it fast

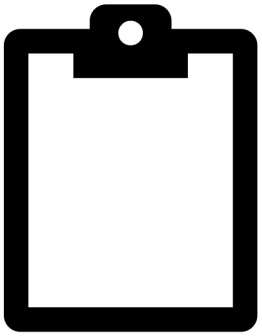
Kent Beck

Thank you for attending!

1

Scan the QR code to confirm you attended today's workshop.

2



Fill out the feedback survey in the next 72h.

3

Get recognition for this workshop on your co-curricular record.