Algorithms

◎ Objectives

At the end of this lesson, you should be able to:

- **define** what an algorithm is
- **translate** a simple problem into an algorithm
- **describe** the actions of an algorithm

1. Algorithm: a first definition

If you look on the internet, an algorithm is "a procedure for solving a logical problem in a finite number of steps that frequently involves recursive operations." This is a *long* and *complicated* definition.

Idea

A simpler way to look at what an algorithm is is to think of a cooking **recipe** or a furniture assembly **instruction manual**. Just as for recipes, algorithms are a way to represent the different steps that go into solving a given problem.

Example: a simple recipe to cook pasta

1. put water in the pan
2. put the pan on the stove
3. heat the pan
4. wait for the water to boil
5. put pasta in the water
6. wait for pasta to be *al-dente*
7. wring out the pasta
8. enjoy

Exercise

Think about a different problem (not necessarily food-related) and represent the steps related to solving it.

2. Flowcharts to represent algorithms

Algorithms are basically recipes, or lists of steps to follow. This works fine when dealing with *simple* problems, as the lists stay manageable. However, imagine we have to deal with long or complex problems. Writing the steps to solve them as lists would be a tedious task.

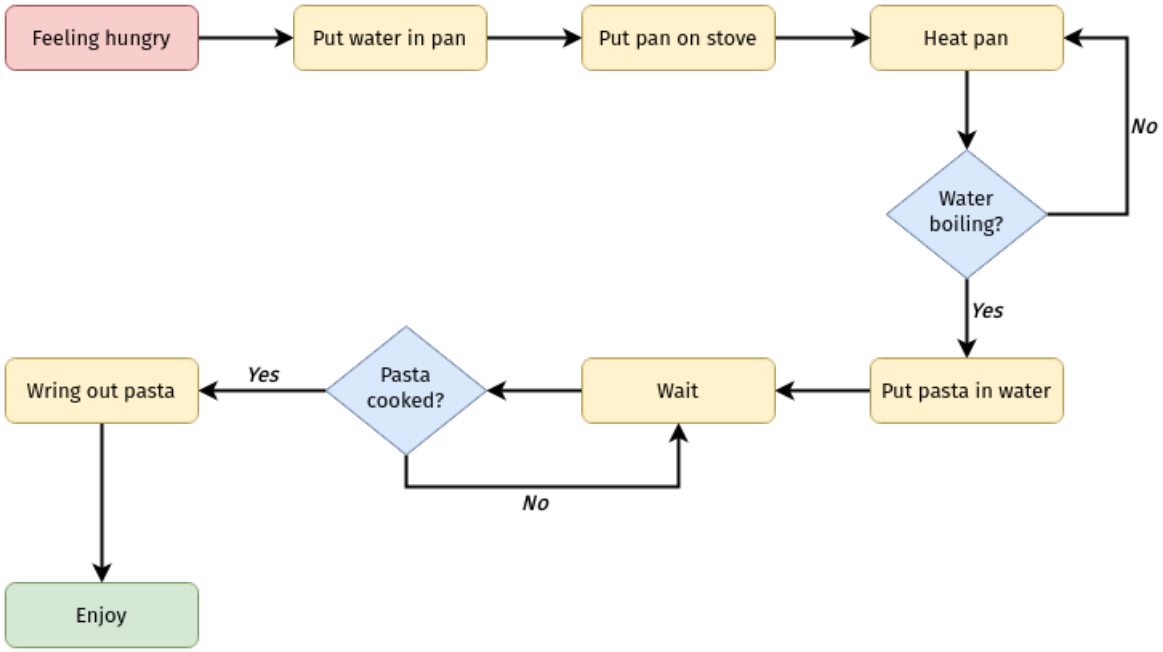
Idea

We need to find a better way than lists to represent the steps taken to solve a given problem. This new method would allow for harder problems to be represented simply as well as simplify the reading of the steps.

Flowcharts allow for this better representation. They use simple drawings to link the different actions. Flows can be single or multiple, and most importantly: **labelled**.

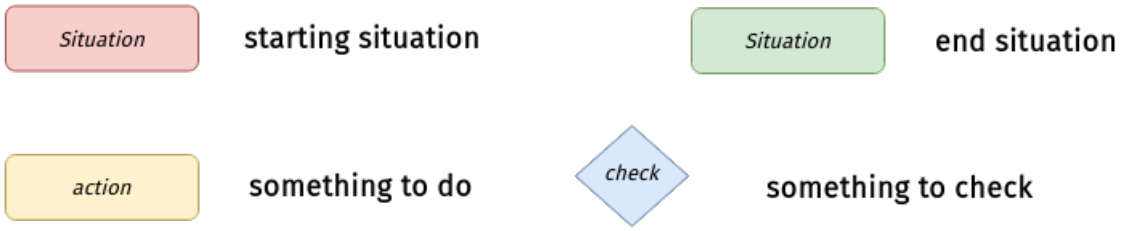
The idea is to make the reading of an algorithm **simple**.

Example: the same recipe, again!



As you can see in the example above, flowcharts rely on simple blocks to represent an algorithm.

Information



Exercise

Draw a flowchart for the problem you found earlier. Does it seem simpler to you now?

Exercise

Can you imagine an algorithm that describes the behaviour of a clock?