

# THE COMPUTATIONAL

# THINKING PROCESS



## 1 | DEFINE QUESTIONS

Think through the scope and details of the problem, defining manageable questions to tackle. Identify the information you have or will need to obtain in order to solve the problem.

## 2 | ABSTRACT TO COMPUTABLE FORM

Transform the question into an abstract precise form, such as code, diagrams or algorithms ready for computation. Choose the concepts and tools to use to derive a solution.

## 3 | COMPUTE ANSWERS

Turn the abstract question into an abstract answer using the power of computation, usually with computers. Identify and resolve operational issues during the computation.

## 4 | INTERPRET RESULTS

Take the abstract answer and interpret the results, recontextualising them in the scope of your original questions and sceptically verifying them. Take another turn to fix or refine.

### Why learn computational thinking?

Because it helps you solve problems—and is increasingly critical for individuals and organisations as computation becomes more effective for decision making across a wide range of activities. As computers become more intelligent, rather than reducing the requirements of human understanding, they have increased them. Unlocking ever-more-insightful answers requires driving the computer with broader computational thinking skills.

Go interactive with the Solution Helix or get this poster at:

[computationalthinking.org/helix](http://computationalthinking.org/helix)



ComputationalThinking.org



computerbasedmath.org