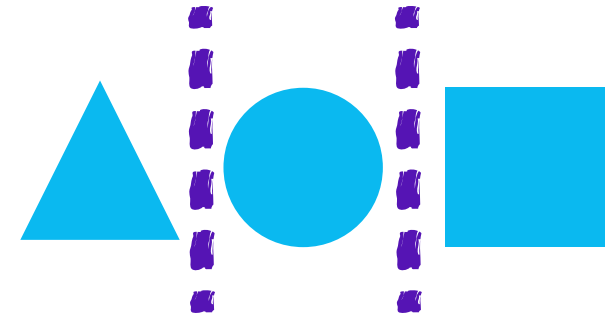


# Decomposition

Breaking down into parts

Decomposition is breaking a problem or system down into its parts.

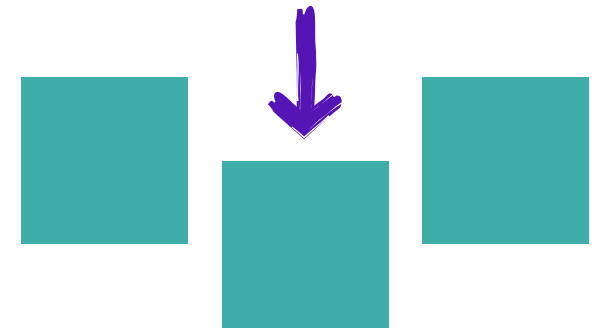


**Barefoot**  
Computing at School

# Tinkering

Experimenting and playing

Tinkering means changing things to see what happens.



**Barefoot**  
Computing at School

# Abstraction

Removing unnecessary detail

Abstraction is identifying what is important and leaving out detail we do not need.

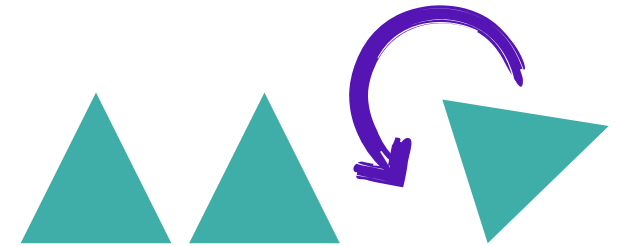


**Barefoot**  
Computing at School

# Debugging

Finding and fixing errors

Debugging is about finding out what is wrong in an algorithm or program and fixing it.

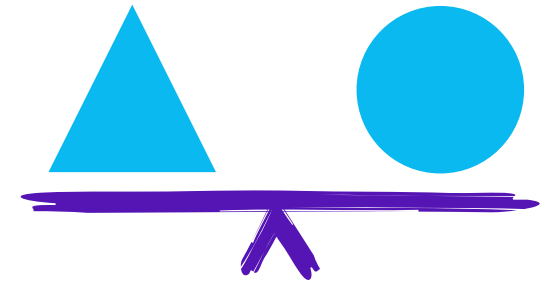


**Barefoot**  
Computing at School

# Evaluation

## Making judgements

We use evaluation when we make judgements based on different factors, such as design criteria and user needs.

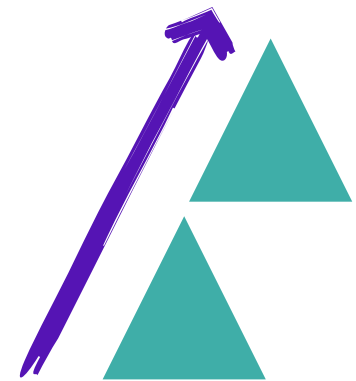


**Barefoot**  
Computing at School

# Persevering

## Keeping going

Persevering is never giving up, being determined, resilient and tenacious.

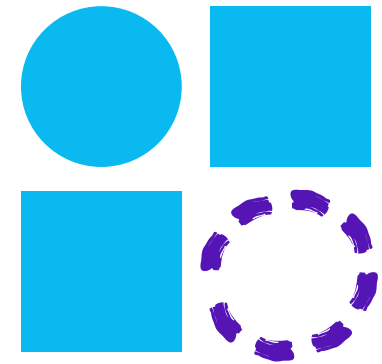


**Barefoot**  
Computing at School

# Patterns

## Spotting and using similarities

By spotting patterns we can make predictions, create rules and solve other problems.

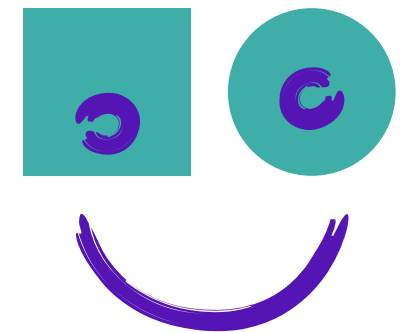


**Barefoot**  
Computing at School

# Creating

## Designing and making

Creating is about planning, making and evaluating things; e.g. animations, games or robots.



**Barefoot**  
Computing at School

# Logic

## Predicting and analysing

Logic helps us to establish and check facts, and make predictions.

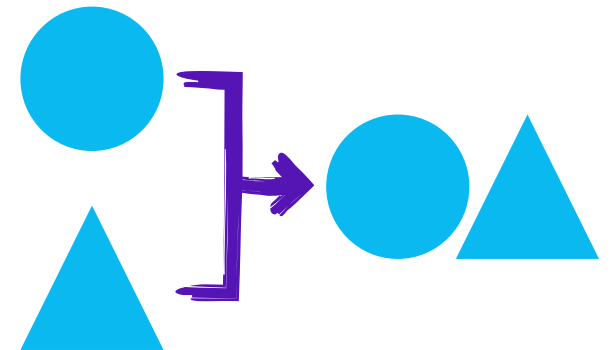


**Barefoot**  
Computing at School

# Algorithms

## Making steps and rules

An algorithm is a precise sequence of instructions, or set of rules, for performing a task.



**Barefoot**  
Computing at School

# Collaborating

Working together

Collaborating means working with others to ensure the best result.



**Barefoot**  
Computing at School

# We're all computational thinkers here!

[barefootcomputing.org](http://barefootcomputing.org)