Methods (functions)
Methods

Can think of methods as packaging multiple commands into one.
Methods

We have used methods before, such as sqrt(), pow() or possibly round()

You can also create your own (similar to creating variables) by:

(1) Give it a name (like variables)
(2) Put it outside of main(), but inside the class

(See: SayHi.java)
public class SayHi {

    public static void main(String[] args) {
        sayHi();
    }

    public static void sayHi() {
        System.out.println("Hello!");
    }
}
Methods

Methods, like variables, have types (int, double, char, etc.)

We call them the return value, as it is what the method will become after being finished

For example: Math.sqrt(4) will become 2.0 (double) when it is finished

(See: Addition.cpp)
Methods

The return statement value must be the same as the return type (or convertible) (See Addition2.java)
Methods

You can actually have multiple methods with the same name, as long as the arguments are different either by:
- a different amount of arguments
- different types of arguments

This is called overloading a method

(See Overloading.java)
Methods

You can make methods return type void, but not variables (an empty variable? ehh...)

This means nothing is returned, so you will get an error if you say:
void x();
... then ...
int y = x(); // x not an int! or anything!

A good use of void methods is to print out
Methods

(See Maze.java)
Methods

It is important to note that the code will resume after the method call where it was used.

For example, sqrt(4) will return the value 2.0 where it was used and the rest of your code will continue.

Where does the maze code return to?
Methods

Multiple methods uses/calls create a “stack” much like pancakes: every time you use a method, it will add another pancake.

When you return, the top pancake is removed.

main() is the bottom pancake.