

Exercises

The functions tell me which move I need to do. In case I need to repeat the move, that function will be inside { } or repeat plus numbers

Conditionals – If

If the point enters inside the conditional, I need to do the function that is called. If not, I do a break and jump outside.

Some exercises show the loop going to a point that never enters the conditional again.

Exercises 16 never enters the condition for example

Condition – Else

With this condition we have another way. When the condition *if* is not met, we can see the other way (*else*). If both are impossible, the code returns or breaks.

Condition – Else/If

This condition is a mix. *Else if* means we need to meet both rules to enter inside the code. If the rules are not met, the code runs outside.

Condition – While

This condition only enters in case the rules are met, but it never exits until the rules are broken and all code is completed.

Condition – !if

Exactly like *if* but negative — if NOT.

Function calls other functions inside

In this case we create a function with `name() { inside more functions() }`.

But the code only runs this function when it is called.

We can create a function that calls another function, that calls another function, and so on.