

Working with Data - Lists



Lists let you store multiple values in one data structure. These values can be anything from a number to a text, another list (a table is a list of lists), a block...

1	1
2	3
3	5

length: 3 ▼

list 1 3 5 ◀▶

You can access elements of a list with the **item _ of _** block. Drop the list you want to access into the second input.

item 1 ▼ of

You can either **select** one of the options from the dropdown menu or **write the index into the first input slot**.

length ▼ of

length
rank
dimensions
flatten
columns
reverse
lines
csv
json

The **of** block **for lists** lets you find out information about a list (e.g. the length input reports the number of elements in the list).

Additionally you can perform operations on lists, e.g. with the reverse input.

length ▼ of list 1 3 5 ◀▶

3

1	5
2	3
3	1

length: 3 ▼

reverse ▼ of list 1 3 5 ◀▶

Working with Data - Lists



set shape ▾ to list ▸

To store a list in a variable and add values during a program, you need to **initialize the variable as an (empty) list**.

add thing to

delete 1 ▾ of

You can edit the list with a program using the **add** and **delete** blocks.

shape

1	90
2	0

length: 2 ▾

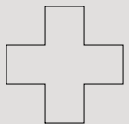
add direction to shape

for each item in list

do something with item

The **for each** block is a **special loop** that only **works with lists** and runs the block in the c-shaped input for each item of the list.

This example creates a design from a list of directions. This list could also be created automatically with a program that controls the turtle with the keyboard.



for each item in list 90 0 90 0 270 0 270 180 270 180 90 180 ◀▶

point in direction item

move 100 steps