technocamps

technocamps

Analysing Climate Change

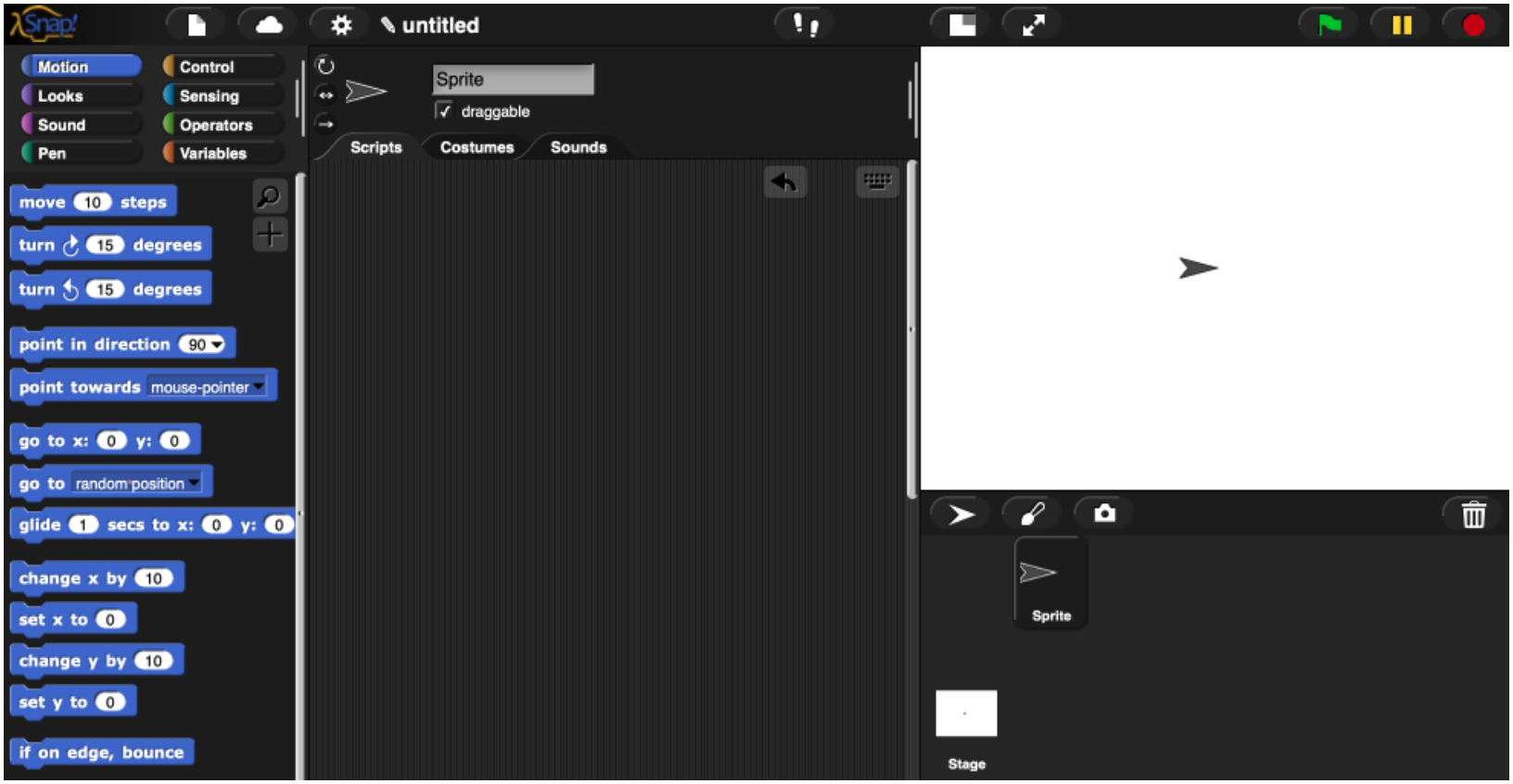
technocamps

The Basics of Snap!

technocamps

Snap!

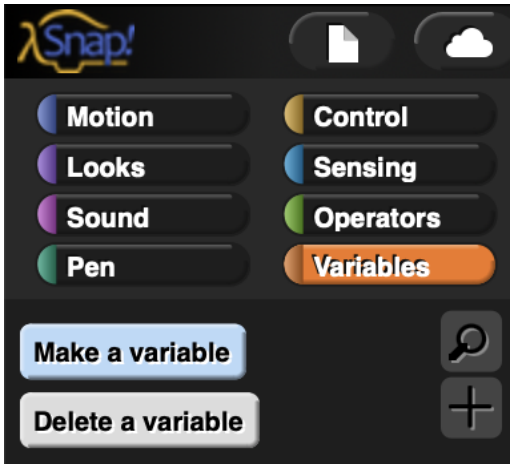
tc1.me/snap



Make a Variable

A variable is a container that can hold some value for us, whether that value is a number, a string, a Boolean or even a list.

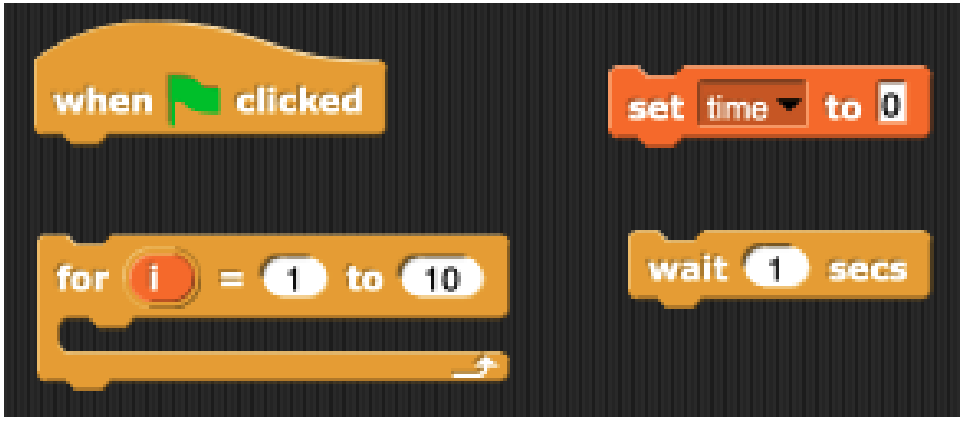
We're going to start by creating a new variable, naming it **time**.



Make a Timer

Using our new variable, we're going to create a simple timer.

We will use a for loop to accomplish this, setting **time** to **i**; the counter variable of the loop.

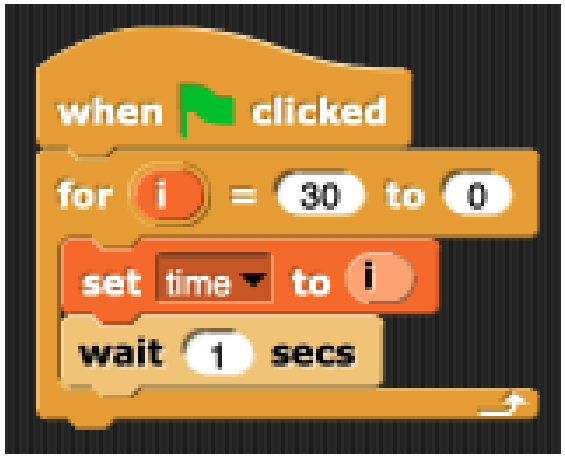


Hint – we can drag the variable **i** from the loop!

Make a Timer

Using our new variable, we're going to create a simple timer.

We will use a for loop to accomplish this, setting **time** to **i**; the counter variable of the loop.



Hint – we can drag the variable **i** from the loop!

technocamps

Using Data in Snap!

technocamps

Download Data Here

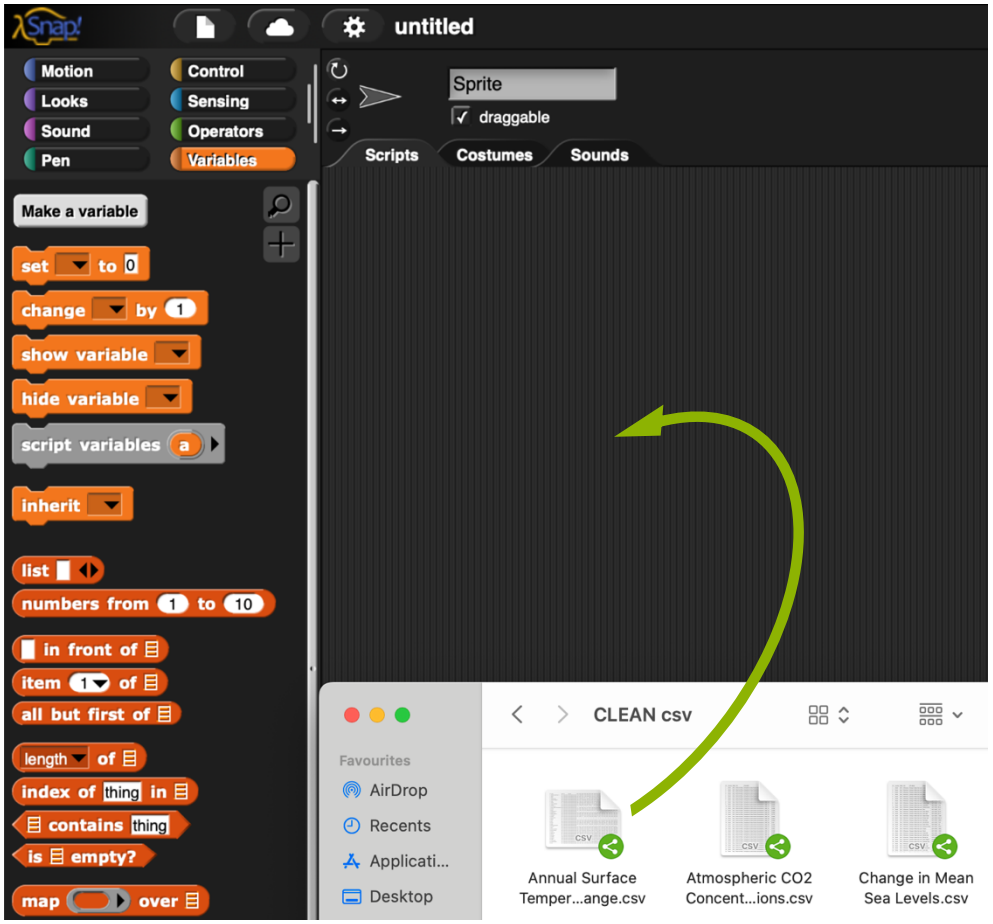
tc1.me/ClimateSnap

Drag and Drop

With *Snap!* we can drag and drop our data straight into the browser!

Use this method to import your downloaded data into *Snap!* so that we can begin analysing.

Be sure to drag in all 3 files!

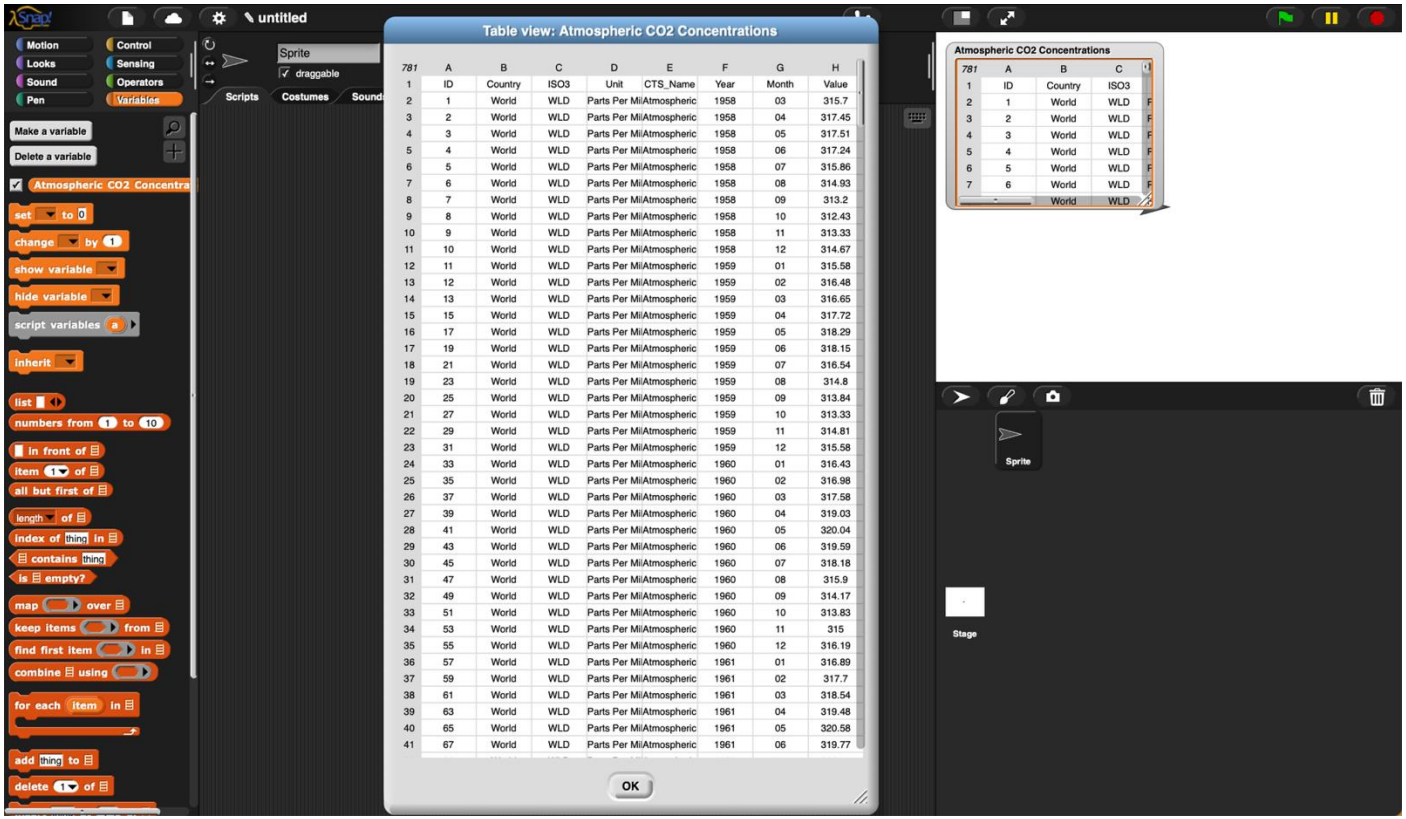


Data as a Variable

Your data will open, appear as a new variable, and be shown on the stage.

Variable

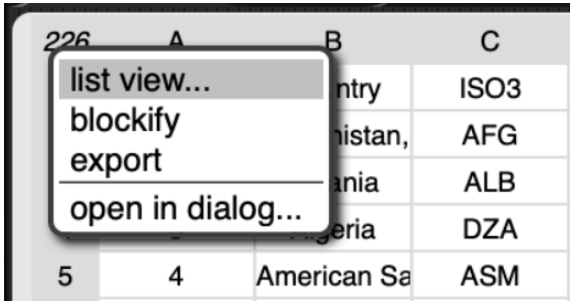
(this can be unchecked to remove from the stage)



List View and Table View

We can drag our variable into the scripting area to use it.

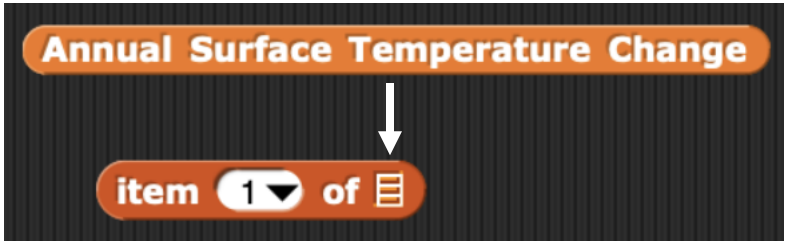
And by clicking it we can open our data to view its contents. A right click will give us the option to see it in list view.



This shows us that a table in *Snap!* is just a list of lists, where each row is stored as an inner list within a larger outer list.

Opening Rows

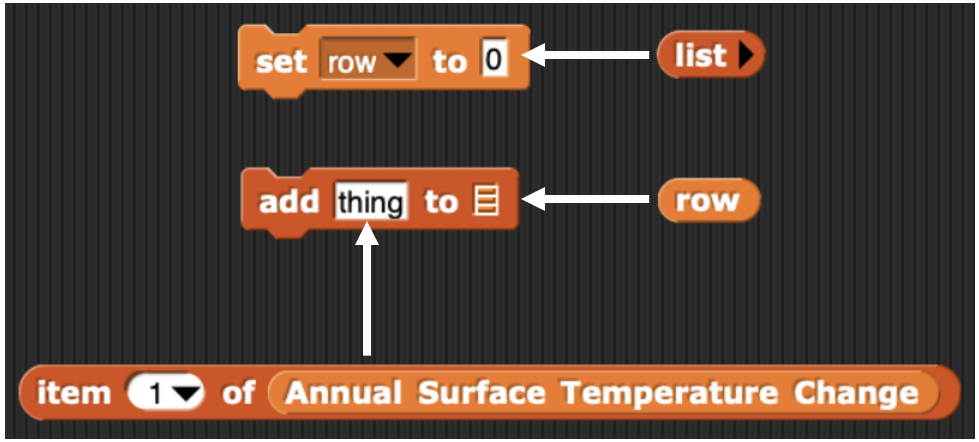
As each row is just an item (a list of values) within our larger list of data, it is very simple to open each row of data.



Storing Rows

To store a row so that it can be used freely in other functions we will need to assign it to a new variable.

Create a variable called **row**, and add a row from our data set.



Note: we will have to set our new variable to have a list value!

Storing Rows

To store a row so that it can be used freely in other functions we will need to assign it to a new variable.

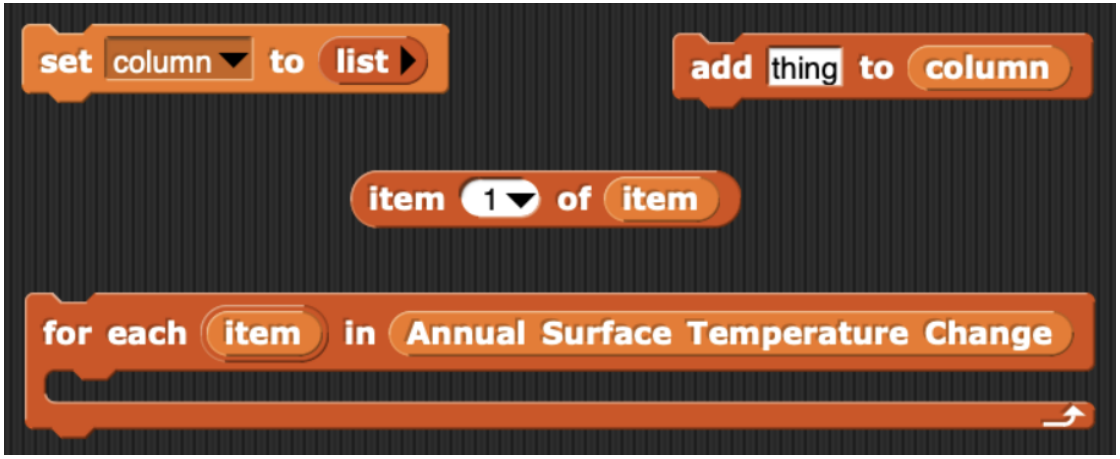
Create a variable called **row**, and add a row from our data set.

Note: we will have to set our new variable to have a list value!

Storing Columns

Storing columns is much the same as rows, however we need to go through each of our lists (rows) one at a time and add the same item (column) from each to our **new** variable.

This requires us to use a for loop.

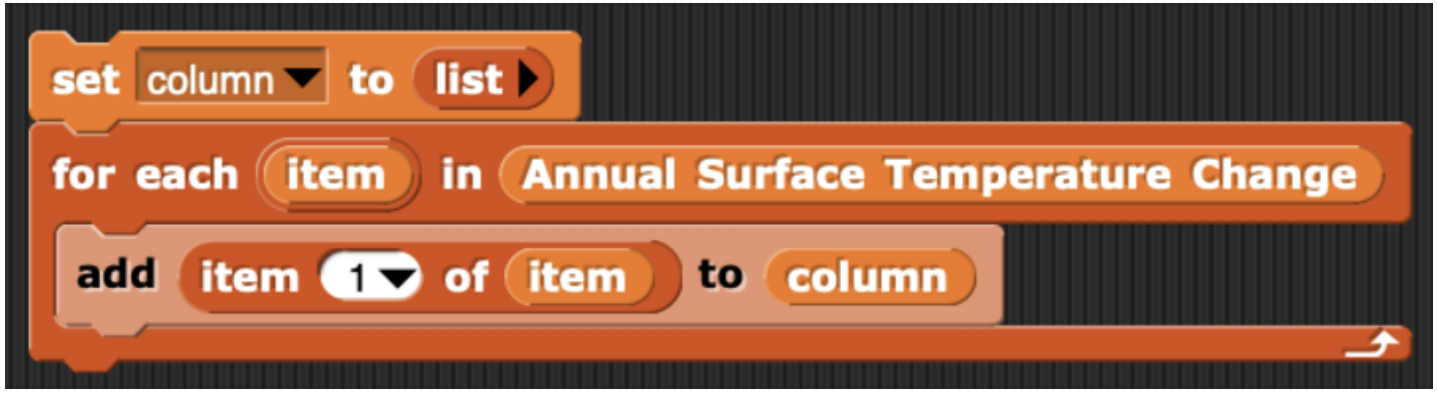


Hint: As before we can drag the variable **item** from the loop!

Storing Columns

Storing columns is much the same as rows, however we need to go through each of our lists (rows) one at a time and add the same item (column) from each to our **new** variable.

This requires us to use a for loop.

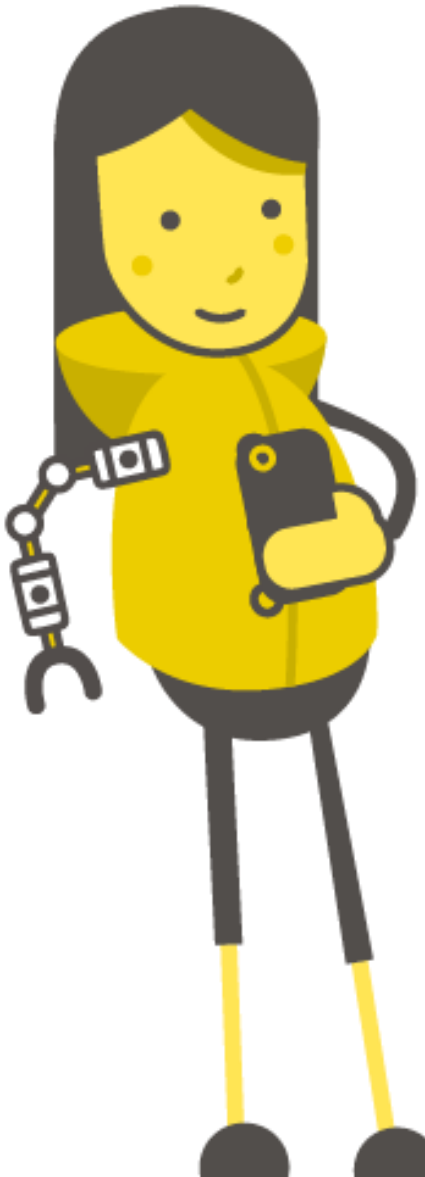


Hint: As before we can drag the variable **item** from the loop!

Map Function

Using loops is a slow process as our program will run through each row individually. *Snap!* instead has a higher order function that can pull columns from our data.

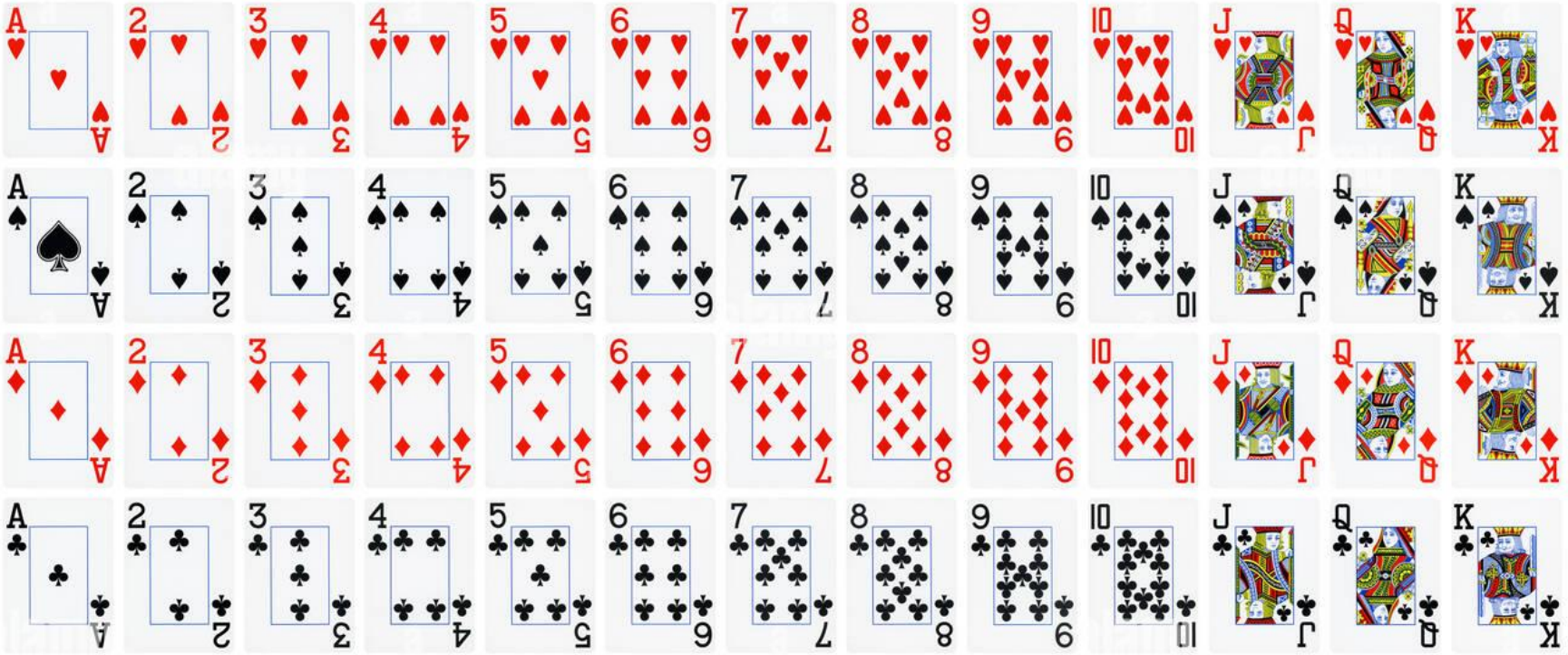
Using **map** will allow us to pull the same item from each inner list (row), as it maps the index of that item over the whole outer list.



Analysing Data in Snap!

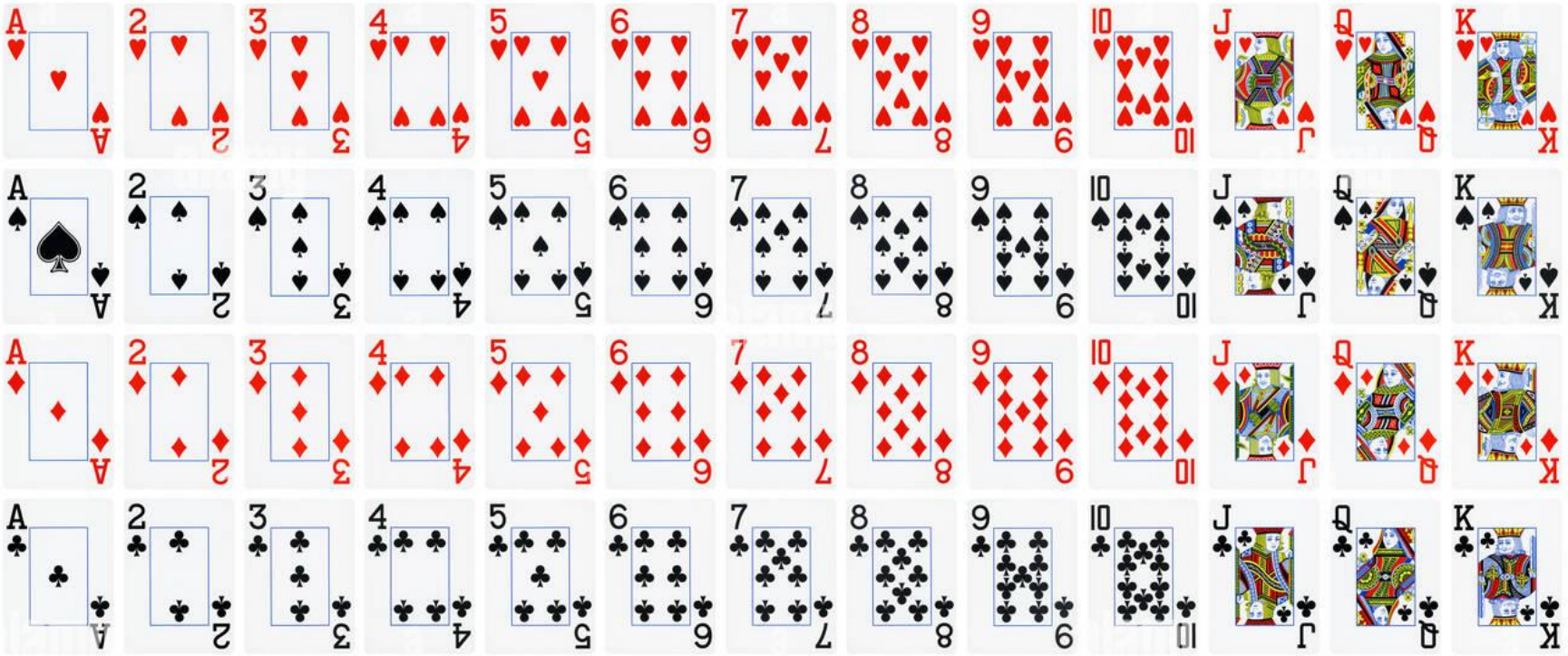
A Pack of Cards

We're going to use a pack of cards to visualise the action of each function as we progress.



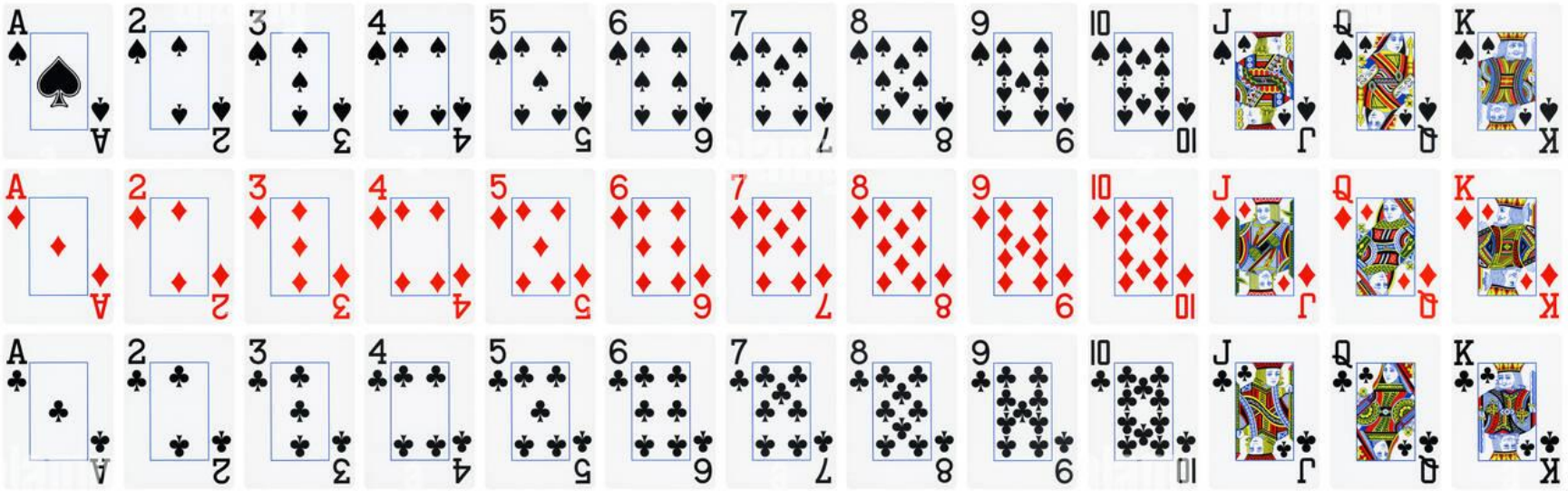
All But First

The **all but first** function will remove the first item from our data.



All But First

The **all but first** function will remove the first item from our data.



All But First

The **all but first** function will remove the first item from our data.

This is beneficial as we can easily remove the first row which contains the headers for each column

225	A	B	C	D	E	F	G	H
1	1	Afghanistan,	AFG	Degree Cels	Surface Tem	-0.113	-0.164	0.847
2	2	Albania	ALB	Degree Cels	Surface Tem	0.627	0.326	0.075
3	3	Algeria	DZA	Degree Cels	Surface Tem	0.164	0.114	0.077
4	4	American Sa	ASM	Degree Cels	Surface Tem	0.079	-0.042	0.169
5	5	Andorra, Prir	AND	Degree Cels	Surface Tem	0.736	0.112	-0.752

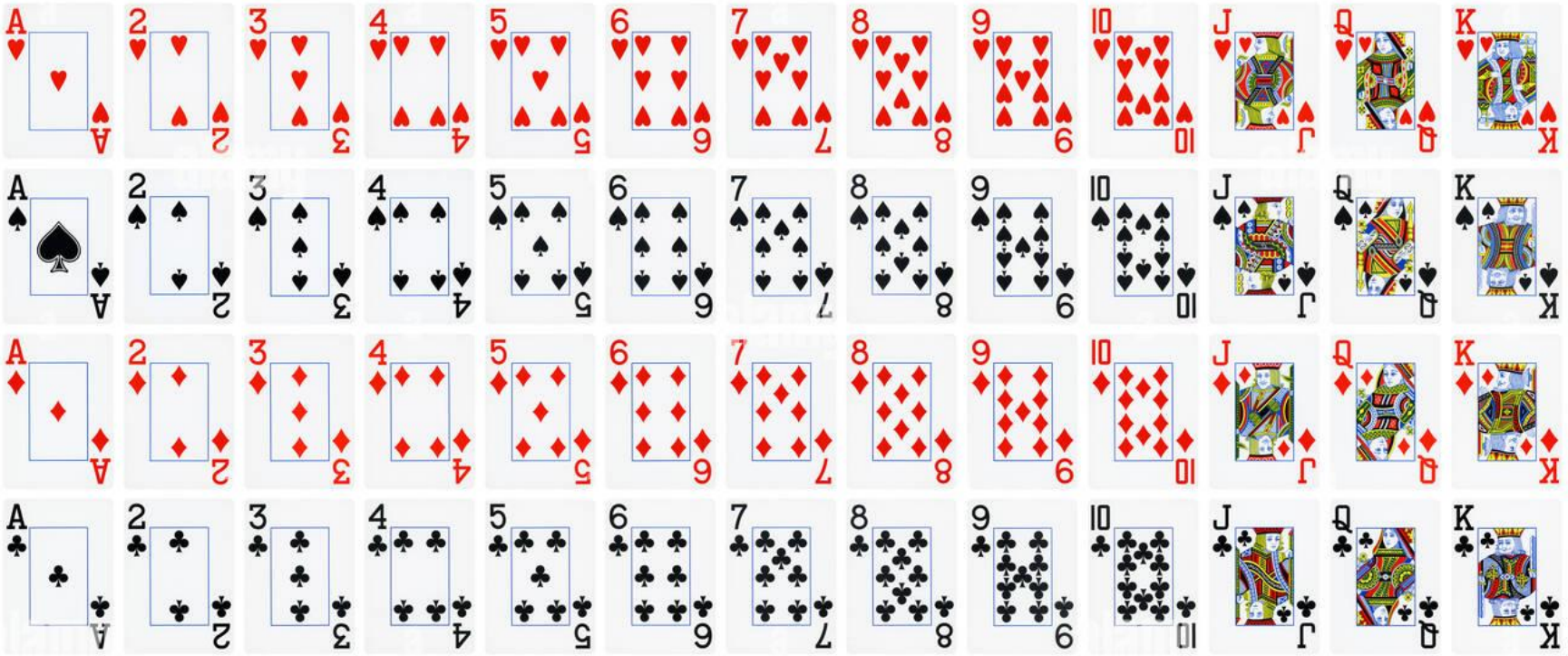
All But First

What will happen if we chain together many **all but first** blocks?

Test it out!

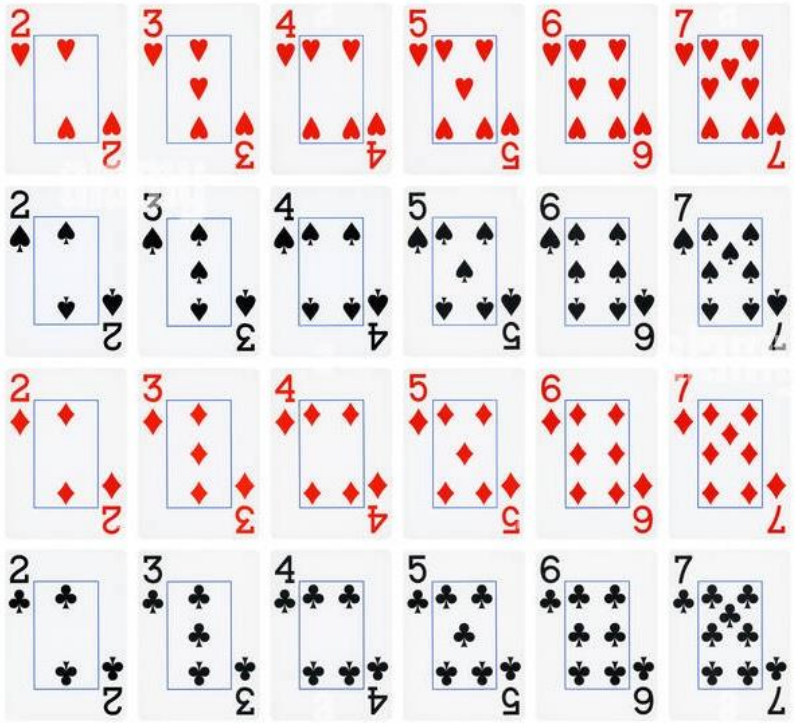
Using Operators

We can begin filtering our data by using operators. A **less than** operator will usually filter out non-numbers too. How about **< 8:**



Using Operators

We can begin filtering our data by using operators. A **less than** operator will usually filter out non-numbers too. How about **< 8:**

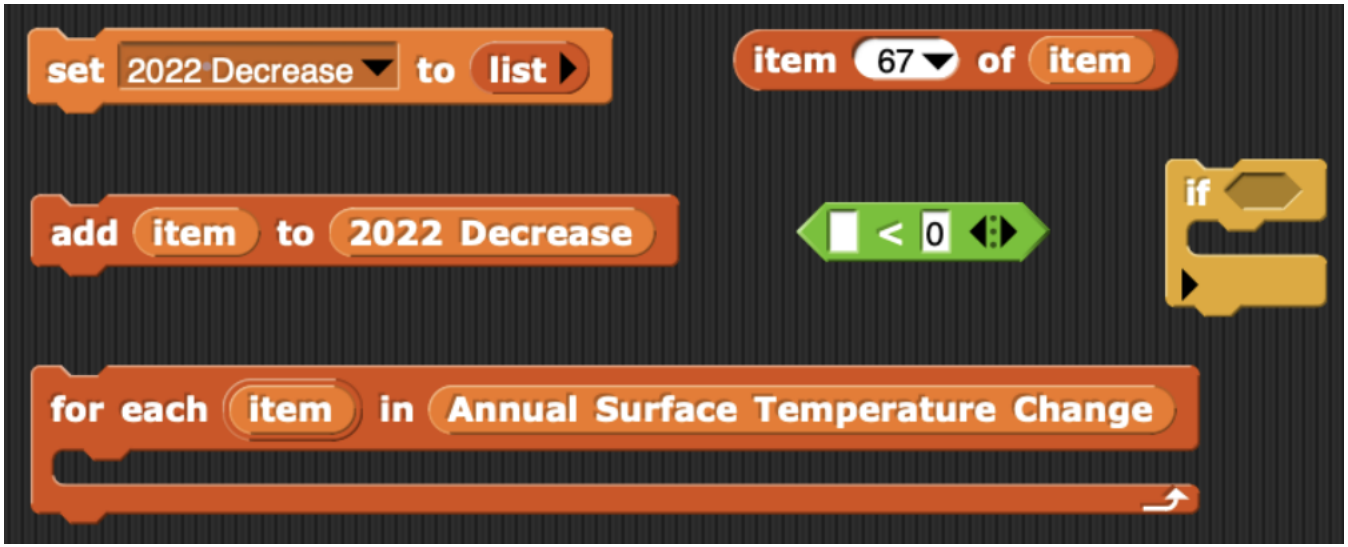


Using Operators

We can begin filtering our data by using operators.

Using the less than block, we can filter our results to just the countries whose Surface Temperature decreased in 2022.

Note: You will need to make a new variable!



Using Operators

We can begin filtering our data by using operators.

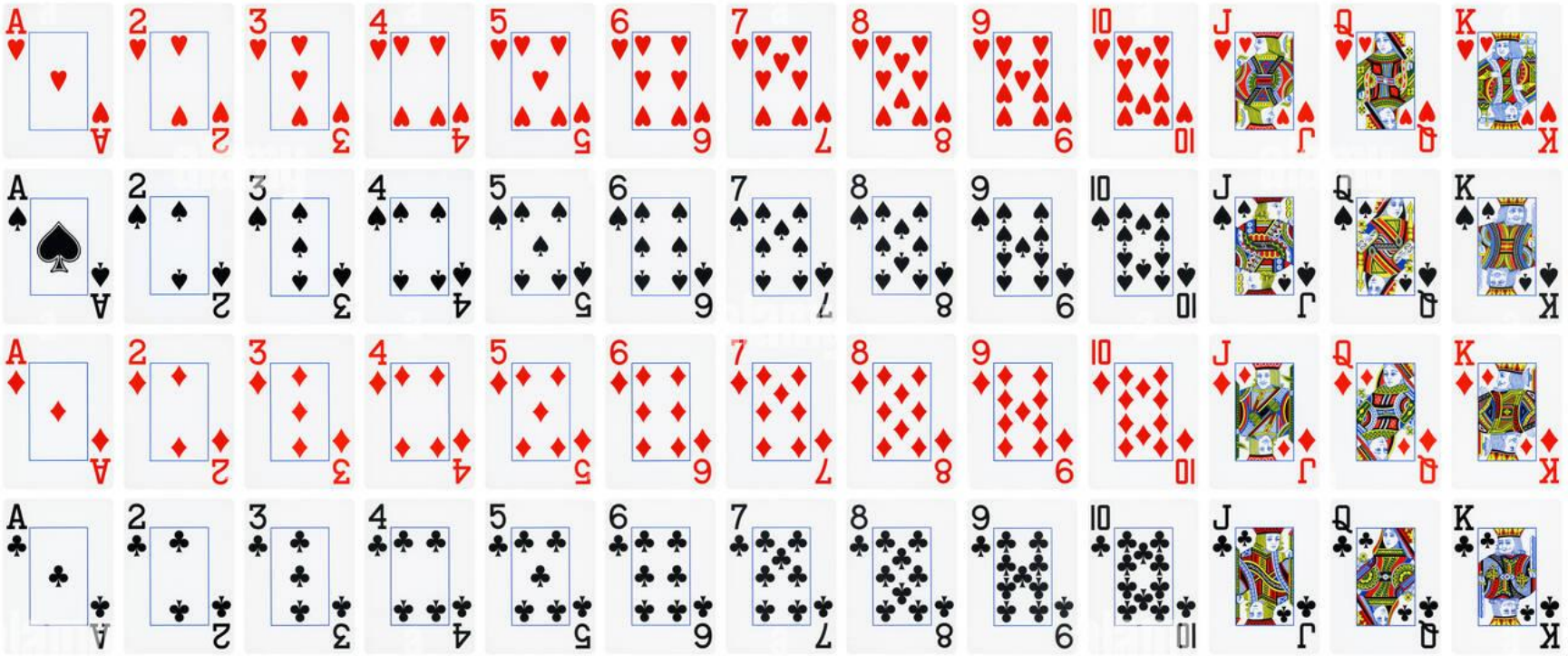
Using the less than block, we can filter our results to just the countries whose Surface Temperature decreased in 2022.

Note: You will need to make a new variable!

Adding To Our Filter

Using an and block we can check for multiple conditions!

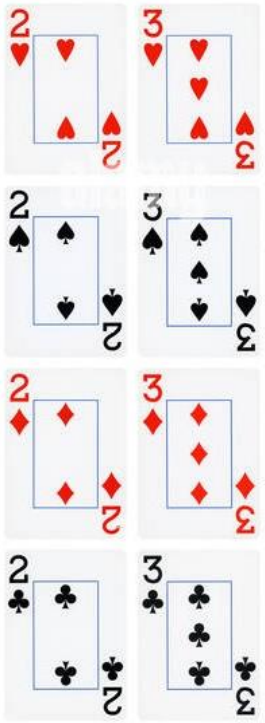
How about **< 4 and > 8:**

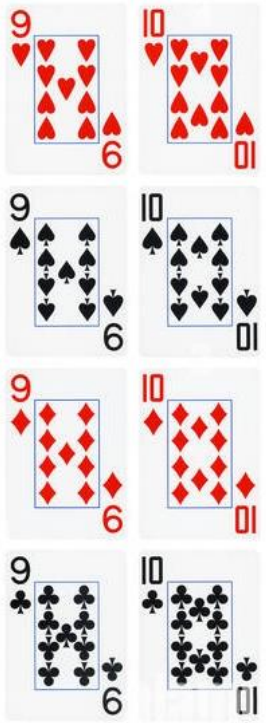


Adding To Our Filter

Using an and block we can check for multiple conditions!

How about **< 4 and > 8:**

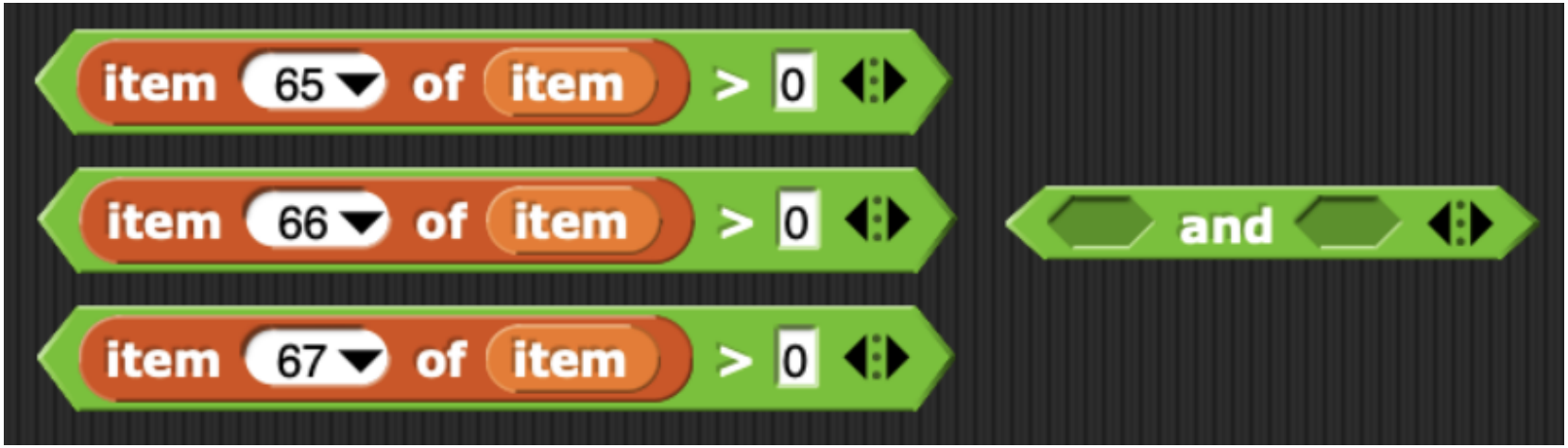




Adding To Our Filter

We can expand our search by checking the values of additional years. We can do this by using an **and** block!

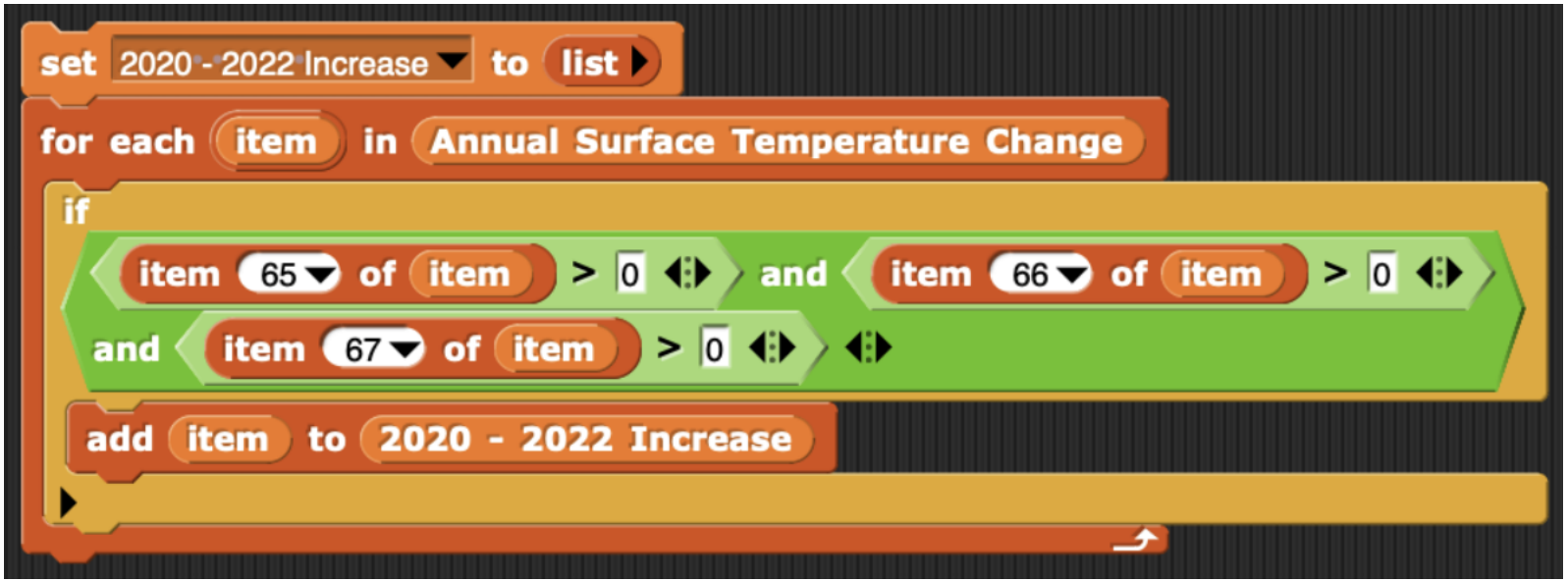
Note that the symbols have changed – it is now **> 0**



Adding To Our Filter

We can expand our search by checking the values of additional years. We can do this by using an **and** block!

Note that the symbols have changed – it is now **> 0**



Adding To Our Filter

You can attempt searching though your data using these operators and the other various operators available!

Test it out yourself and see what you can discover!

Adding Additional Libraries

In *Snap!* we can add additional functions by importing libraries.

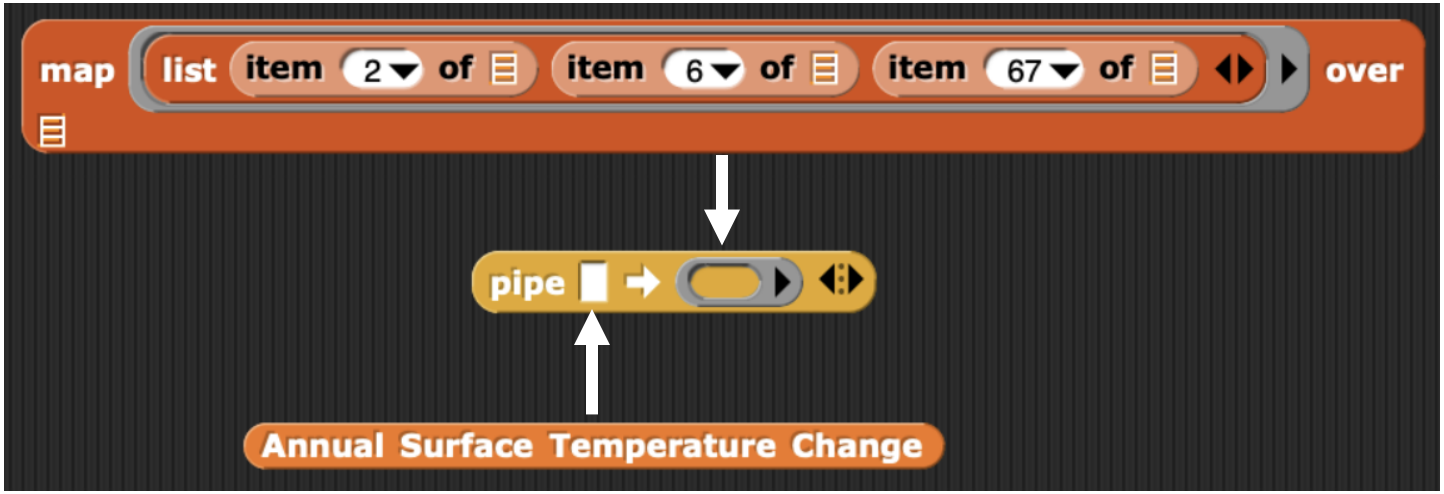
These can be found in the menu at the top of the page.

Add the Frequency Distribution Analysis library.

Pipe Function

The **pipe** function makes it easier to see how we're manipulating the data step by step. This will become clearer as we add to it.

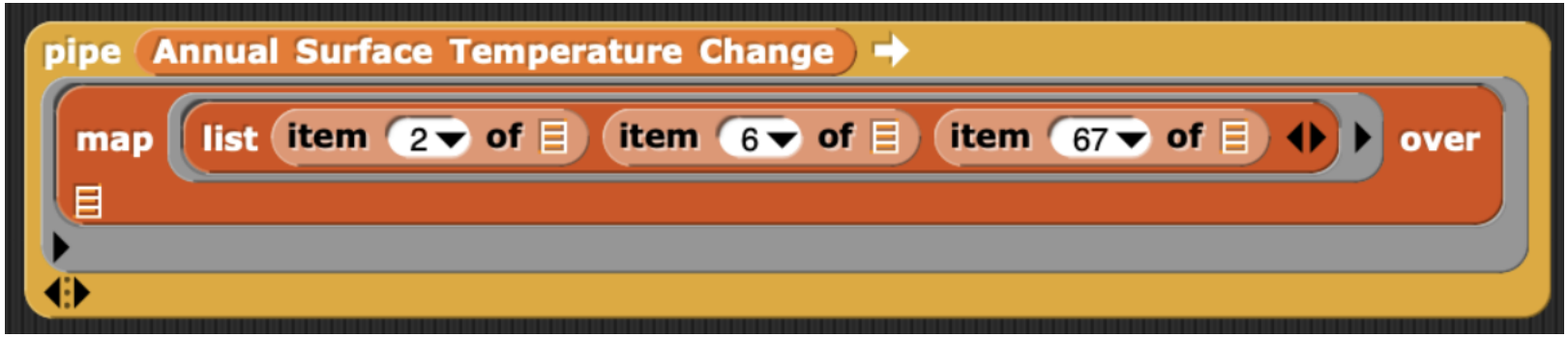
The result of each function is passed to the next, chaining them together and avoiding the confusion of one large nested expression.



Pipe Function

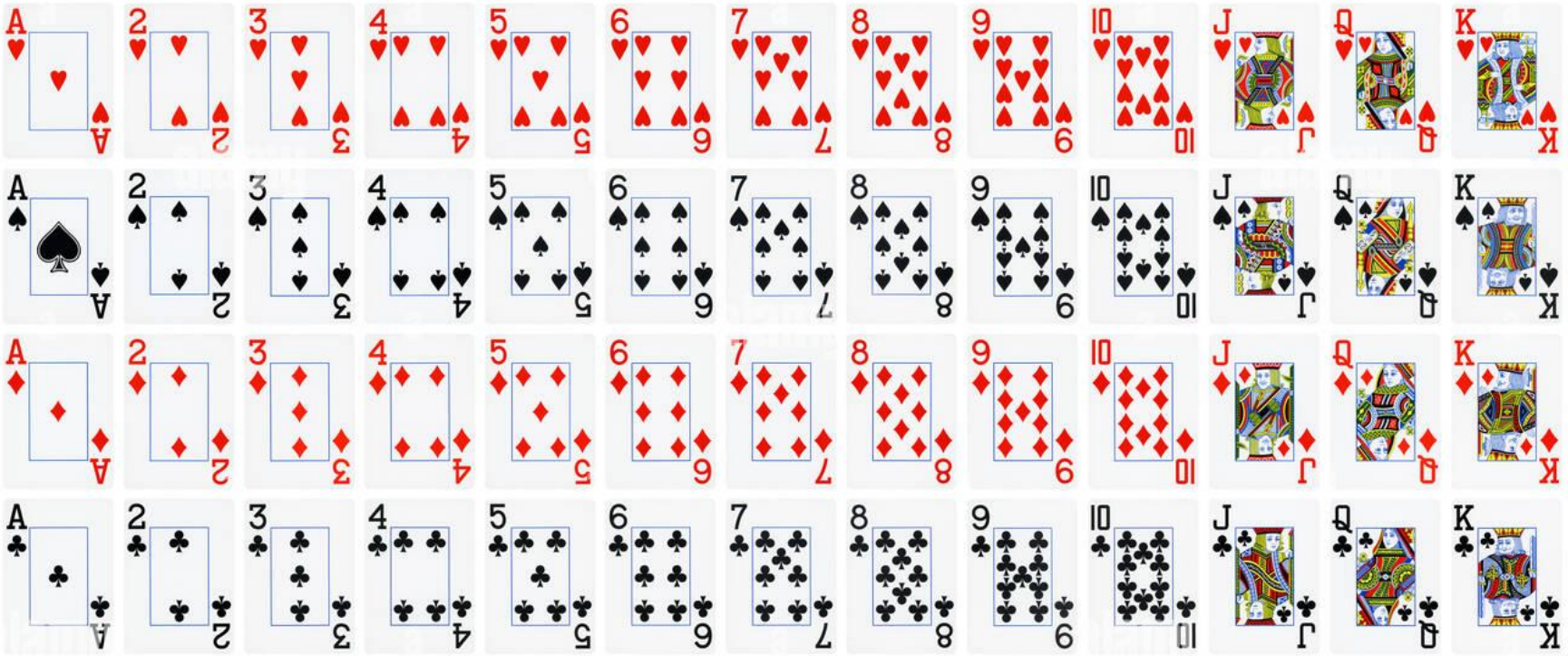
The **pipe** function makes it easier to see how we're manipulating the data step by step. This will become clearer as we add to it.

The result of each function is passed to the next, chaining them together and avoiding the confusion of one large nested expression.



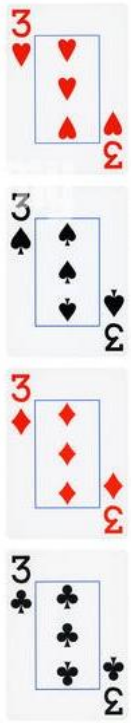
Keep Items

The **keep items** block performs the same function as the **if statement** filter we built earlier i.e. **keep items = 3**:



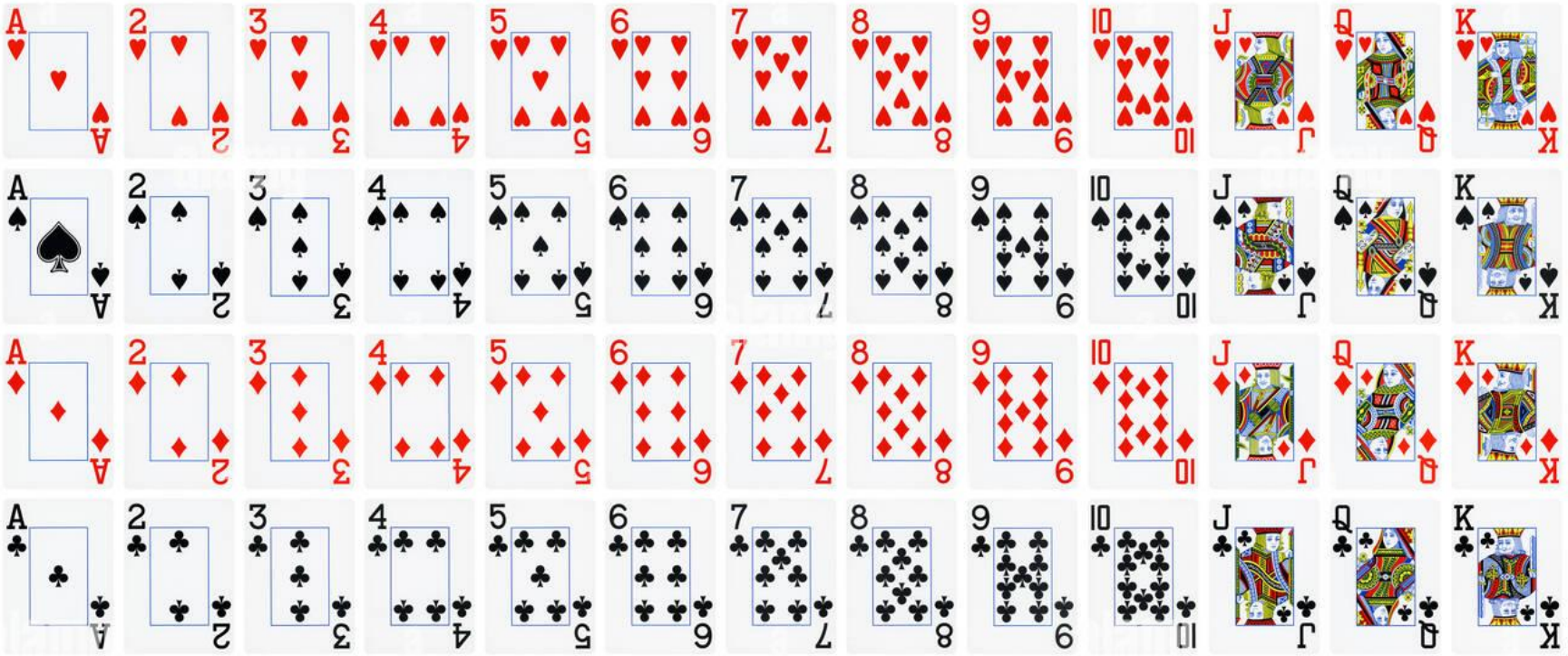
Keep Items

The **keep items** block performs the same function as the **if statement** filter we built earlier i.e. **keep items = 3**:



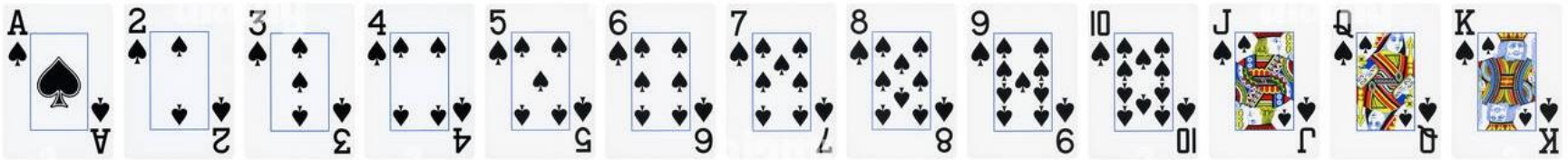
Keep Items

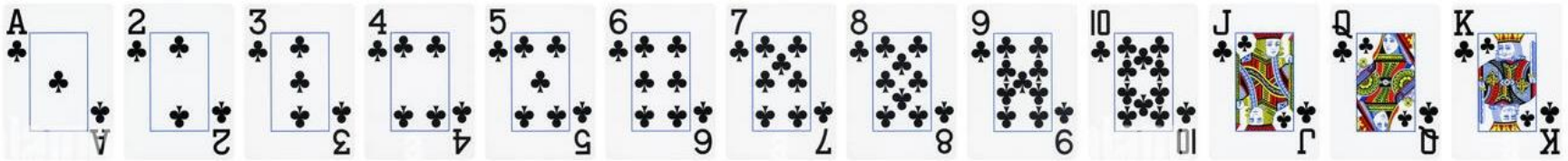
Or **keep items = ♠ and ♣**:



Keep Items

Or **keep items = ♠ and ♣**:

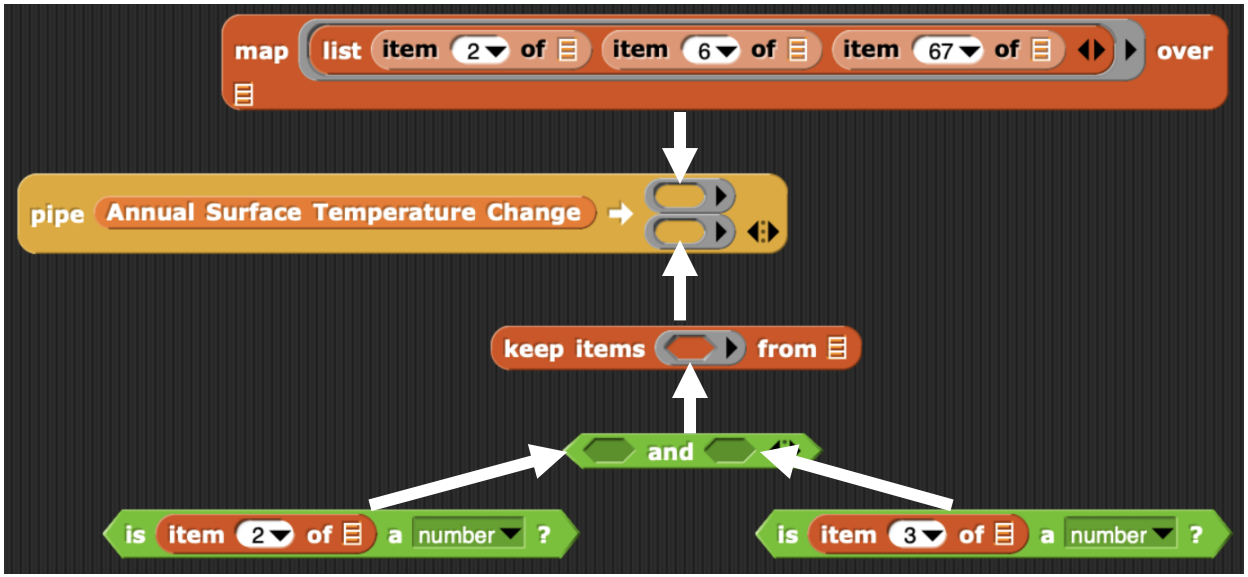




Keep Items

The **keep items** block performs the same function as the **if statement** filter we built earlier.

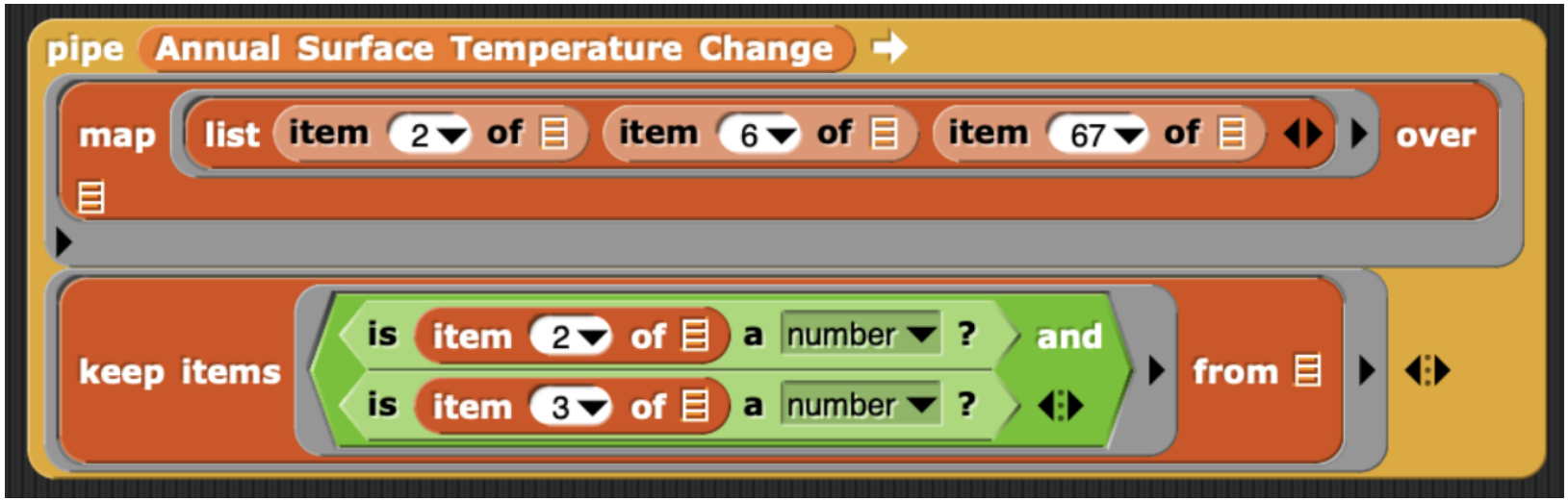
Using an operator to specify a condition will only return the data that fits that condition.



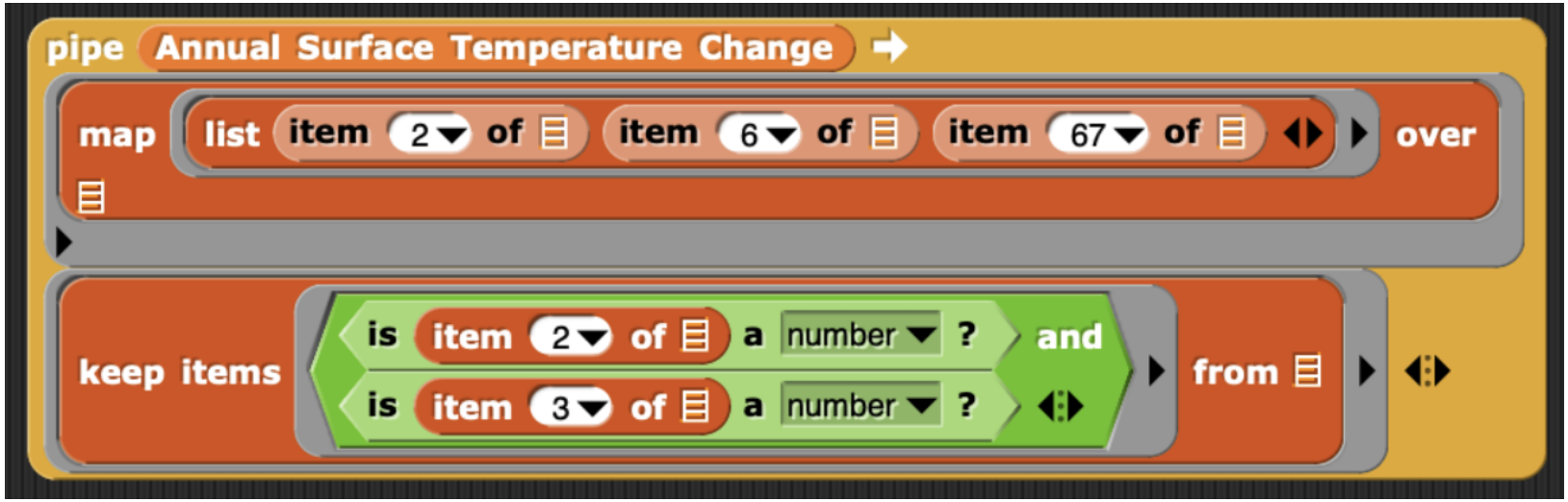
Keep Items

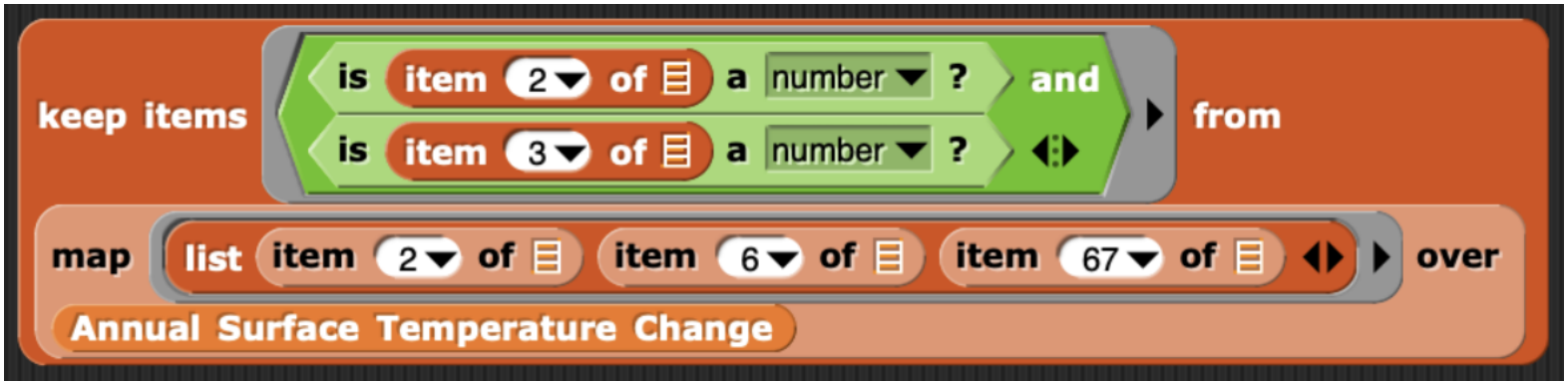
The **keep items** block performs the same function as the **if statement** filter we built earlier.

Using an operator to specify a condition will only return the data that fits that condition.



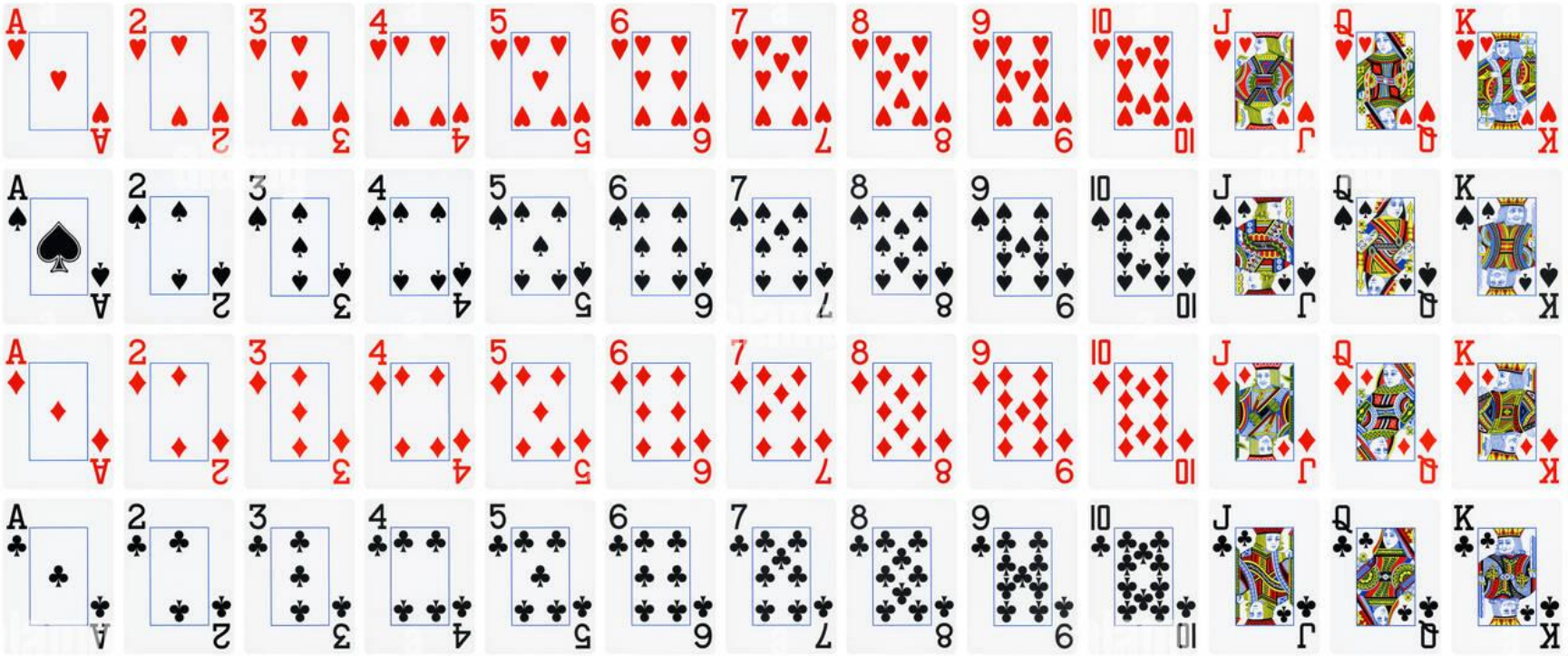
Pipe vs Nested





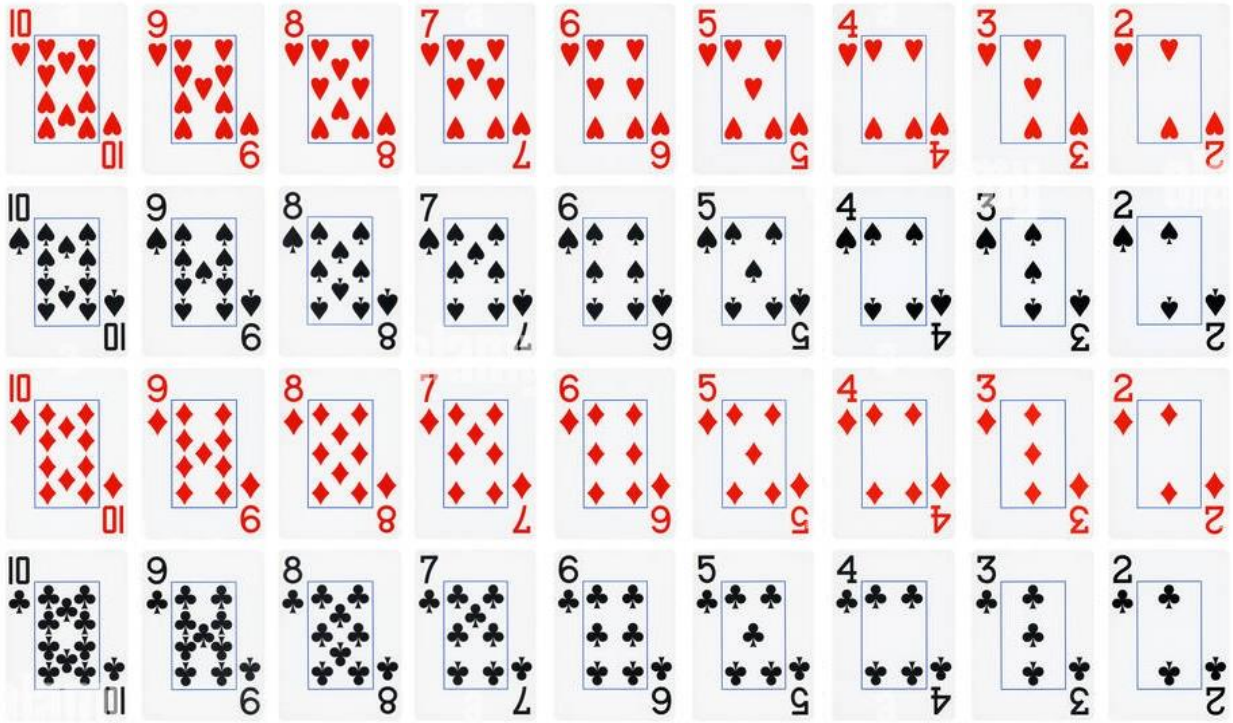
Sort Function

The **sort** function will arrange our data in any way we like. Using the **is … a number** and **>** operators we can sort from high to low:



Sort Function

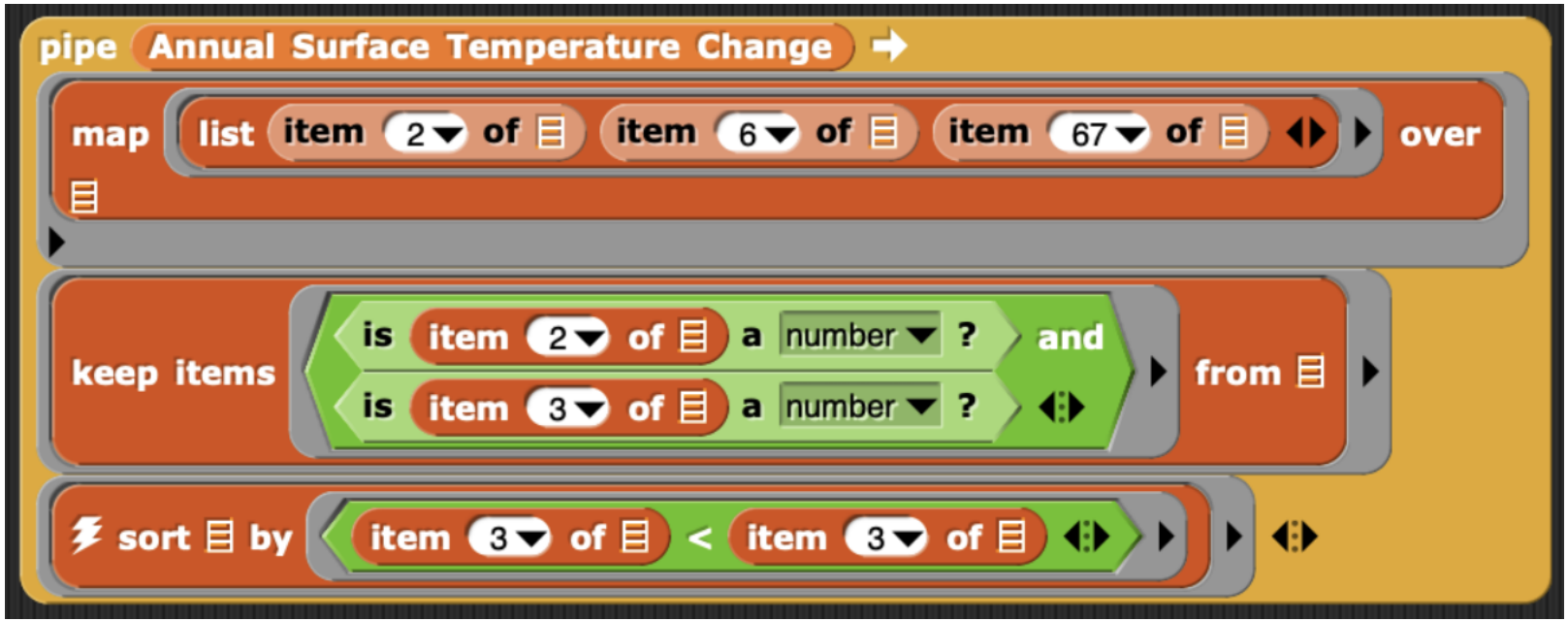
The **sort** function will arrange our data in any way we like. Using the **is … a number** and **>** operators we can sort from high to low:



Sort Function

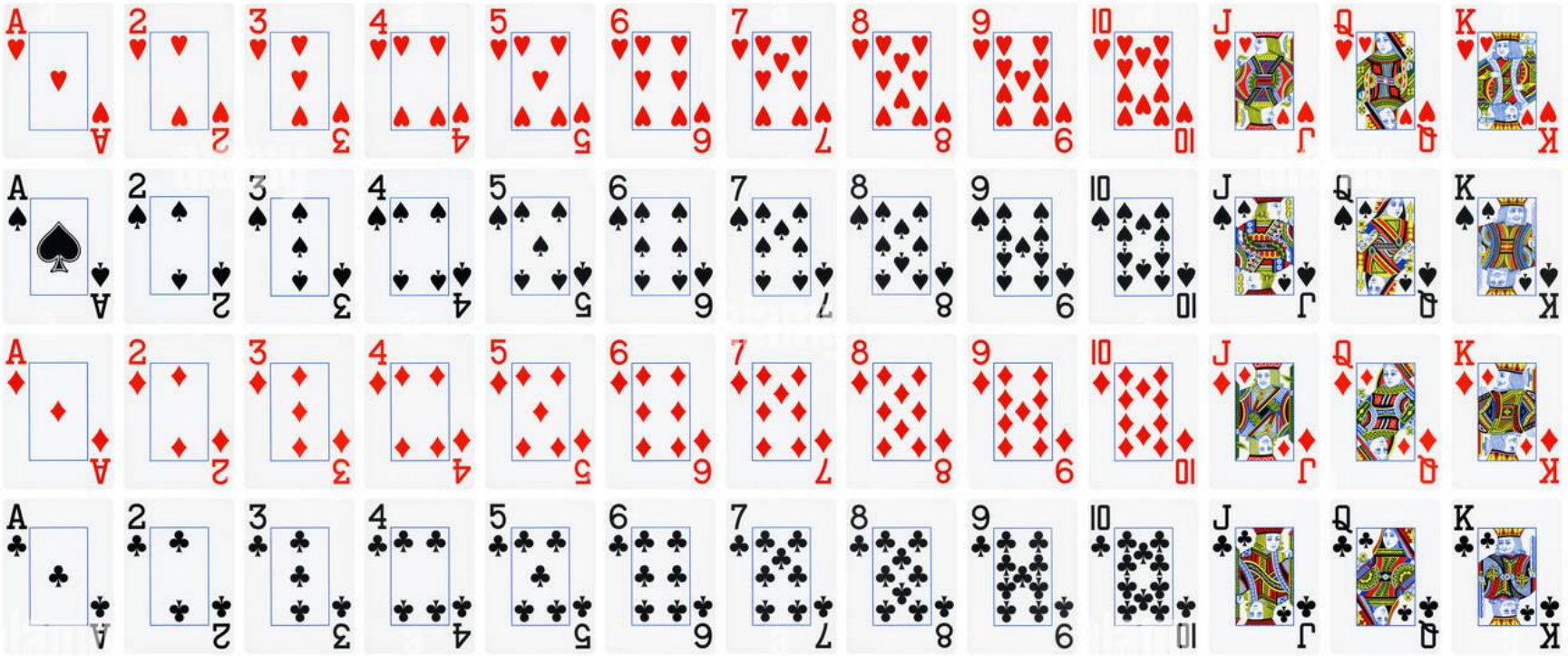
The **sort** function will arrange our data in any way we like.

Using the **less than (<)** operator we can arrange our data by a number value from lowest to highest.



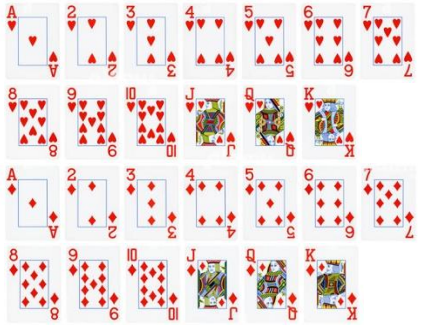
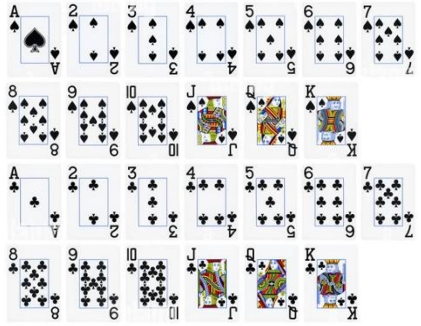
Group Function

The **group** function will group our data by each unique entry and give us the number of uses of that entry. In column C is a list containing each corresponding entry.



Group Function

The **group** function will group our data by each unique entry and give us the number of uses of that entry. In column C is a list containing each corresponding entry.

Red	**24**	
Black	**24**	

Group Function

The **group** function will group our data by each unique entry (column A) and give us the number of uses of that entry in our data (column B).

In column C is a list containing each corresponding entry.

This is functionally a histogram of our data.

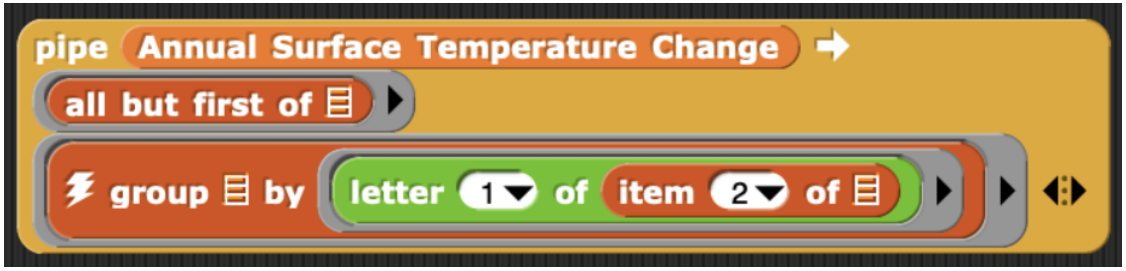
We can group our countries by the letter they begin with:

25	A	B	C
1	C	22	
2	A	14	
3	B	18	
4	D	4	
5	E	8	

Group Function

An improvement on our group function would be to use an **all but first** block.

This is because on its own **group** will also group our headers!



25	A	B	C
1	A	14	
2	B	18	
3	C	21	
4	D	4	
5	E	8	

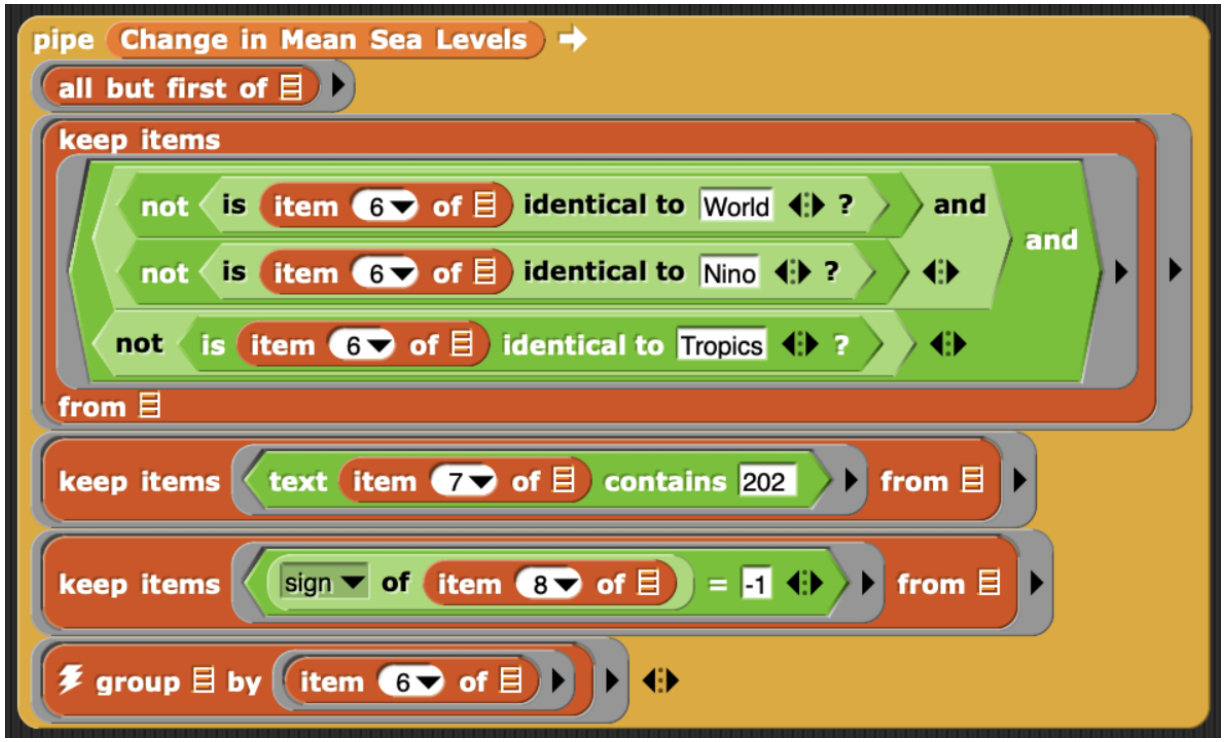
Keep, Sort & Group Functions

As before try using your knowledge of the keep, sort and group functions to analyse your data!

Try to look at your data in new ways, see what secrets you can find within, and don't be afraid of getting stuff wrong!

Don't forget that you have 3 data sets you can manipulating!

Keep, Sort & Group Functions



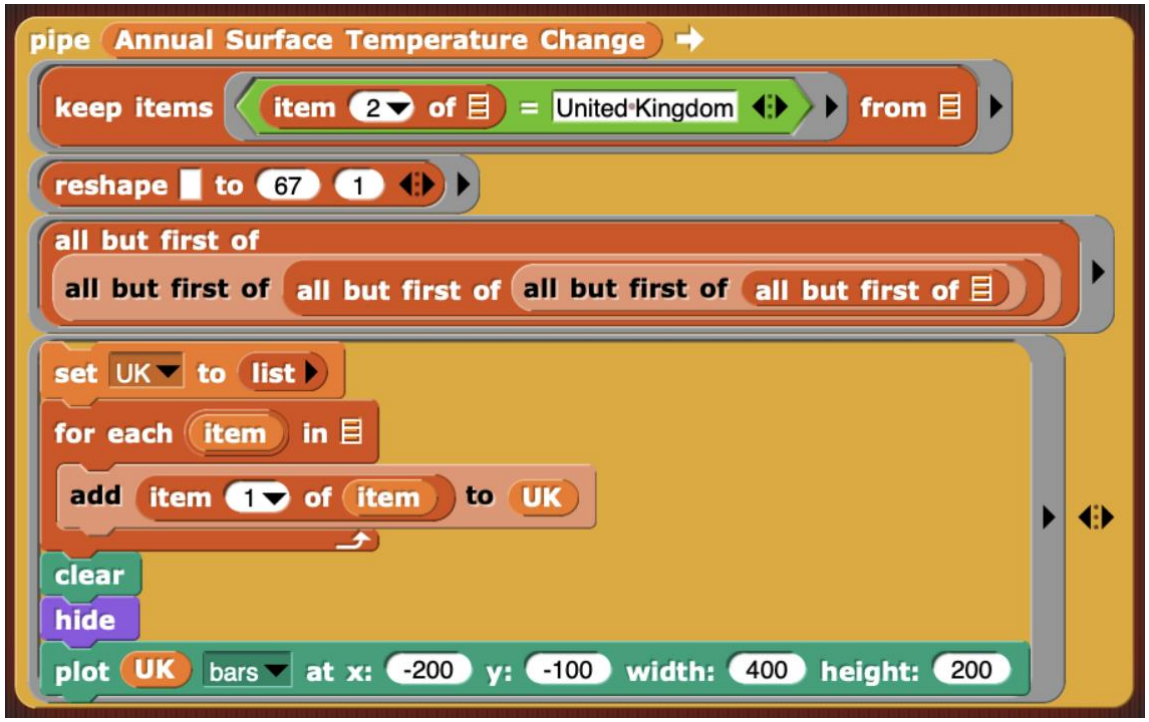
Can you work out what is being shown in this data manipulation?

Plot

Using all that we've learned we can now use the **plot** function to plot a histogram of the **Change in Mean Sea Level**.

We need to use the **map** function as only one list of data (the value column) can be inputted to plot a bar chart.

Plot



Due to the orientation of our data (data is stored in rows, not columns) this make plotting the Surface Temperature of a single country a bit trickier!

Plot

Try using the plot function to plot other parts of your data!

Don't forget that you have 3 data sets you can try plotting!