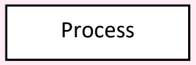
Flowchart Symbols

We can represent algorithms using flowcharts

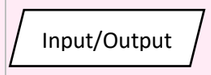
Start and Stop

Process – An operation that the algorithm performs

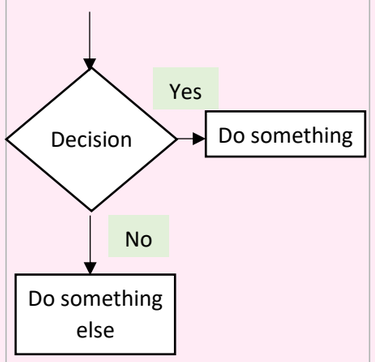


Connector – Links all the other symbols together

Input and Output of data that is read in and written out



Decision is the same as a selection (if then … else)




```
IF answer is "yes" THEN
      do something
ELSE IF answer is "no"
      do something else
ENDIF
```

Pseudocode

We can represent algorithms using pseudocode

	Example	Python equivalent
Variable assignment	a ← 10	`a = 10`
Constant assignment	constant PI ← 3.142	`PI = 3.142`
Input	a ← USERINPUT	`a = input()`
Output	OUTPUT "Bye"	`print("Bye")`

Arithmetic Operators
 	
Add
Multiply
Divide
Subtract
Integer division
Modulus (remainder)
	
	`+`
	`*`
	`/`
	`-`
	`a ← 7 DIV 2`
	`a ← 7 MOD 2`
		
		`+`
		`*`
		`/`
		`-`
		`a= 7 // 2`
		`a = 7 % 2`

Relational Operators Less than	 `<`	 `<`
Greater than	`>`	`>`
Equal to	`=`	`==`
Not equal to	`≠ or <>`	`!=`
Less than or equal to	`≤`	`<=`
Greater than or equal to	`≥`	`>=`

Boolean Operators
 	
AND
OR
NOT
	
	`AND`
	`OR`
	`NOT`
		
		`AND`
		`OR`
		`NOT`

Selection
 	
if ..
	
	`IF i > 2 THEN`
	`  j ← 10`
	`ENDIF`
		
		`if i  > 2:`
		`  j=10`

if .. else …	`IF i > 2  THEN`
	`  j ← 10`
	`ELSE`
	`  j ← 3`
	`ENDIF`
		`if i > 2:`
		`  j=10`
		`else:`
		`  j=3`

if … else if … else	`IF i ==2 THEN`
	`  j ← 10`
	`ELSE IF i==3`
	`THEN`
	`  j ← 3`
	`ELSE`
	`  j ← 1`
	`ENDIF`
		`if i ==2:`
		`  j=10`
		`elif i==3:`
		`  j=3`
		`else:`
		`  j=1`

Iteration
 	
While loops
	
	`a ← 1`
	`WHILE a < 4`
	`  OUTPUT a`
	`  a ← a + 1`
	`ENDWHILE`
		
		`while a<4:`
		`  print(a)`
		`  a=a+1`

For loops	`FOR a ← 0 TO 3`
	`  OUTPUT a`
	`ENDFOR`
	`a ← 1`
		`for a in`
		`range(3):`
		`  print(a)`

Repeat loops	`REPEAT`
	`  OUTPUT a`
	`  a ← a + 1`
	`UNTIL a←4`	

Arrays

	Example	Python equivalent
Set up array	`a ← [1,2,3,4,5]`	`a=[1,2,3,4,5]`
Access element	`a[0]`	`a[0]`
Update element	`a[0] ← 4`	`a[0] = 4`
Set up 2D array	`a ← [[1,2],[3,4]]`	`a = [[1,2],[3,4]]`
Access 2D element	`a[0][1]`	`a[0][1]`
Update 2D element	`a[0][1] ← 4`	`a[0][1] = 4`

Subroutines		

procedure	`SUB hello()`
	`  OUTPUT "hello"`
	`ENDSUB`
		`def hello():`
		`  print("hello")`

Function (with paramerters and return)	`SUB add(n)`
	`  a ← 0`
	`  FOR a ← 0 TO n`
	`  a ← a + n`
	`  ENDFOR`
	`  RETURN a`
	`ENDSUB`
		`def add(n):`
		`  a=0`
		`  for a in`
		`range(n+1):`
		`  a=a+n`
		`  return a`

Built-in functions		
Length of array	LEN(a)	len(a)
Random integer	RANDOM_INT(0, 9)	import random random.randint(0,9)