# PSEUDOCODE – Python EQUIVALENTS

## **PSEUDOCODE**

Python

### Variable and Constant Data Types

Declare Integer varName

Set *varName* = initialValue

Declare Real *varName*

Set *varName* = initialValue

Declare Character *varName*

Set *varName* = ‘x’

Declare String *varName*

Set *varName* = “initial value”

Declare Boolean *varName*

Set *varName* = “initial value” *varName* = initialValue

*varName* = initialValue

*varName* = ‘x’

*varName* = ‘initial value’

*varName* = ‘initial value’

Constant Integer *varName =* value

Constant Real *varName* = value

Constant Character *varName*  = ’x’

Constant String varName = “value”

Constant Boolean *varName = “value”*

*VAR\_NAME* = value

*VAR\_NAME* = value

*VAR\_NAME* = ‘x’

*VAR\_NAME* = ‘value’

*VAR\_NAME* = ‘value’

**Note:** Initial values and values must be appropriate for the specified data type.

### FUNCTIONS, MODULES AND PROCEDURES

Declare Module calcInc (Real sal, Integer mo, reference Real inc)

Set inc = sal × mo

End Module calcIncdef calcInc (sal, mo, inc):

inc = sal \* mo

Call with this statement: calcInc(200.00, 6, total)

Declare Real Function calcInc (Real sal, Integer mo)

Declare Real inc

Set inc = sal × mo

Return inc

End Function calcIncdef calcInc (sal, mo):

inc = salary \* months

return inc

Call with this statement: total = calcInc (200.00, 6)

Module main

Declare Real inc

Set inc = calcInc(200.00, 6)

Display inc

End main def main():

inc = calcInc(200.00, 6)

print(inc)

main()

**MATHEMATICAL OPERATIONS**

|  |  |  |
| --- | --- | --- |
| Symbol | Operation | Description |
| + | Addition | Adds two numbers |
| - | Subtraction | Subtracts one number from another |
| \* | Multiplication | Multiplies one number by another |
| / | Division | Divides one number by another and gives the result as a floating number |
| // | Integer division | Divides one number by another and gives the result as an integer |
| % | Remainder | Divides one number by another and gives the remainder |
| \*\* | Exponent | Raises a number to a power |

**Mathematical Order of Operations**

1. Parenthesis [**Note:** this is the only *truly* reliable way to perform operations in the way you want]
2. Exponent
3. Multiplication, division and remainder in the order they appear, left to right
4. Addition and subtraction in the order they appear, left to right

### RELATIONAL OPERATORS

Greater Than

Less Than

Greater Than or Equal To

Less Than or Equal To

Equal To

Not Equal To>

<

>=

<=

==

!=

### LOGICAL OPERATORS

AND

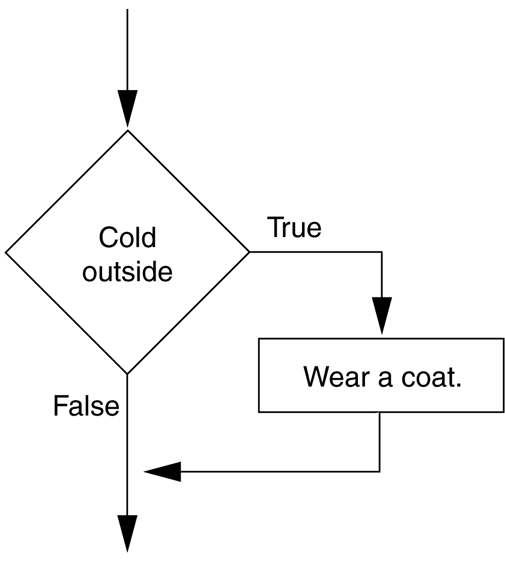
OR

NOTand

or

not

### SINGLE ALTERNATIVE DECISION STRUCTURE

****

If *condition* Then

statement

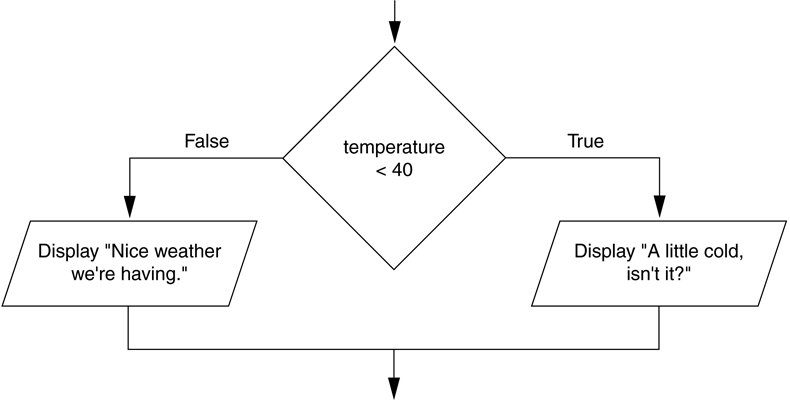
statement

End If**if** (condition):

statement

statement

### DUAL ALTERNATIVE DECISION STRUCTURE



If *condition* Then

statement

statement

Else

statement

*statement*

End If

If ( (*condition) AND ( condition) )* Then

statement

statement

Else

statement

*statement*

End If**if** (condition):

statement

statement

**else:**

statement

statement

**if** ( (*condition) and ( condition)):*

statement

statement

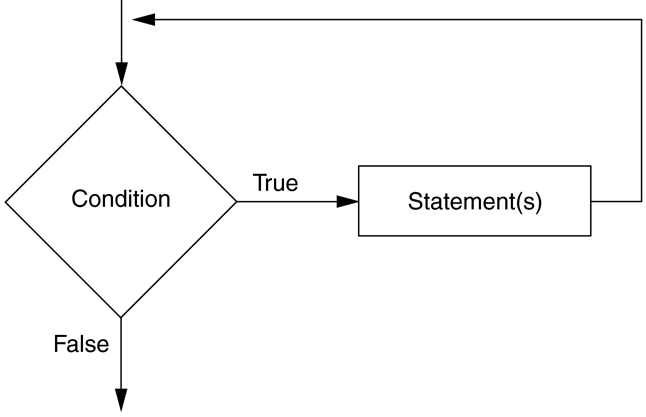
**else:**

statement

statement

## PRETEST LOOPS

### While loop



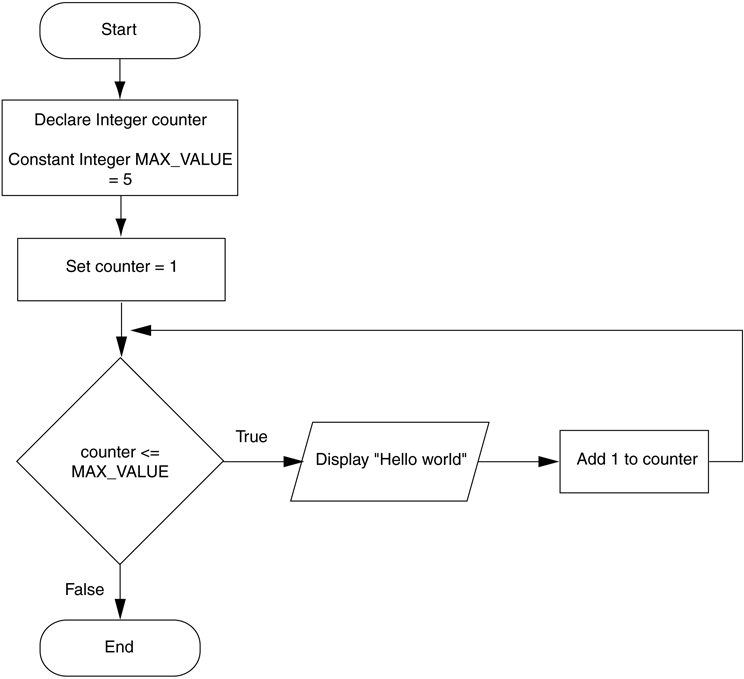
While *condition*  
 Statement  
 Statement  
End While

**while** (*condition*):

Statement

Statement

### For loop



For *counter* from *startVal* to *endVal*   
 Statement  
 Statement  
End For**for** *counter* in range(startVal, endVal+1):  
 Statement  
 Statement