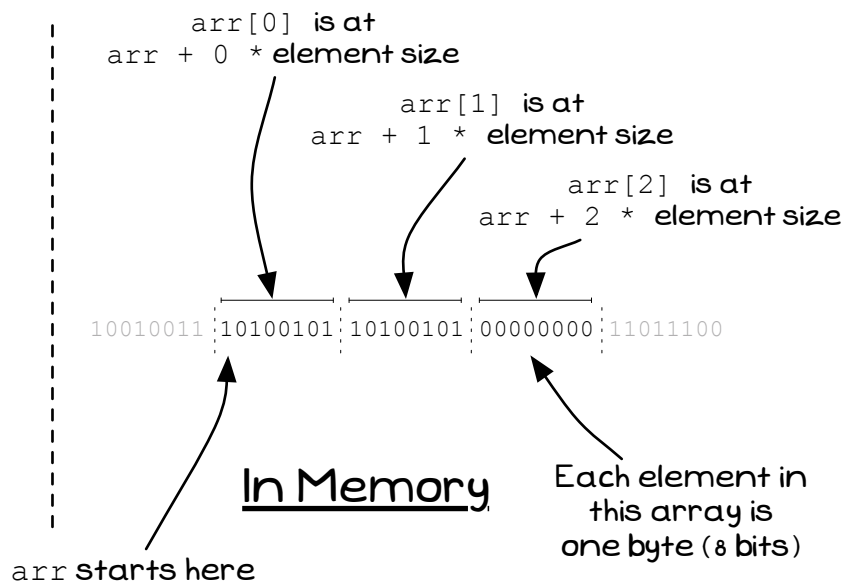
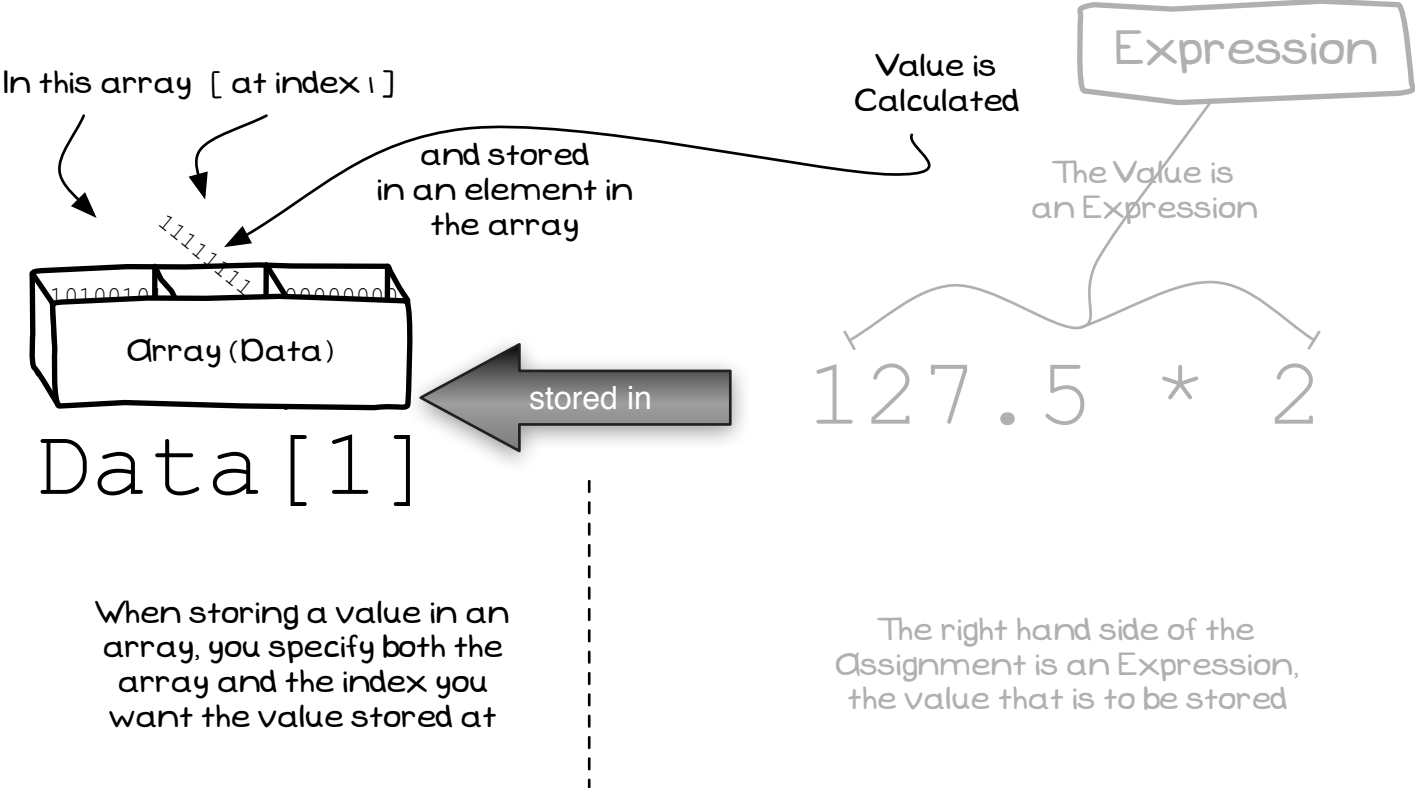
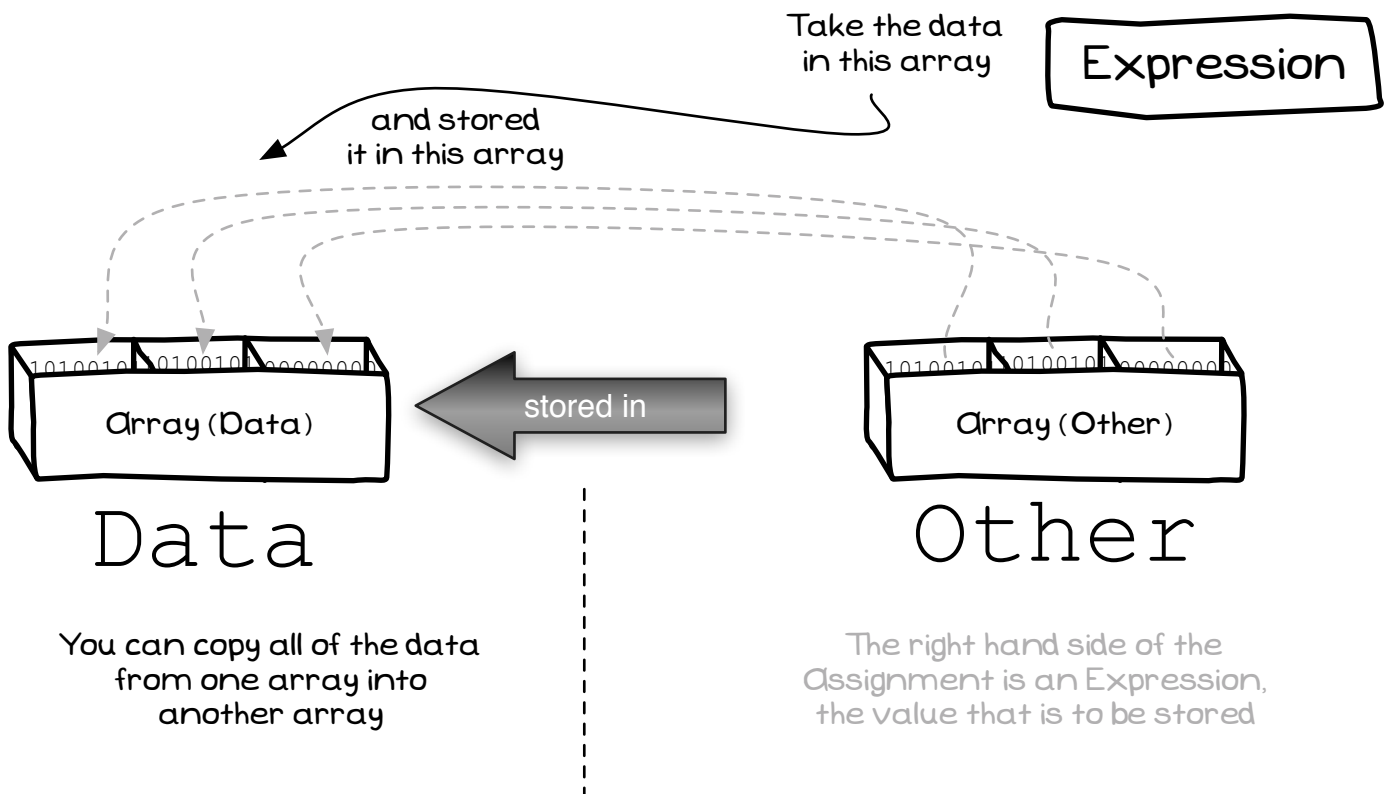


Conceptually







Is a value.

Expression

Values are read  
from array elements



Part or all of the  
value can be read  
from array(s)

$3 * \text{Data}[0] + \text{Data}[i+1]$

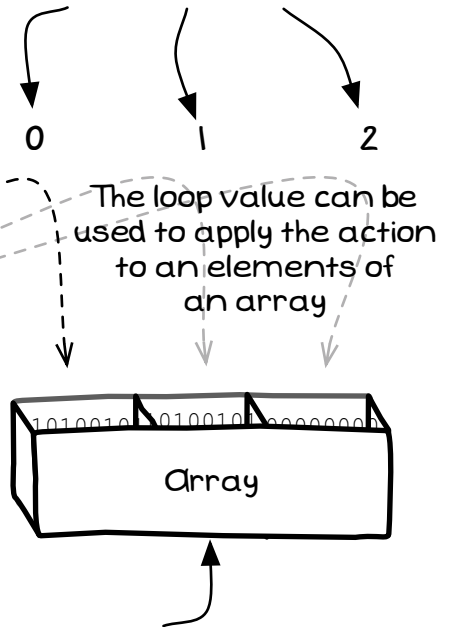
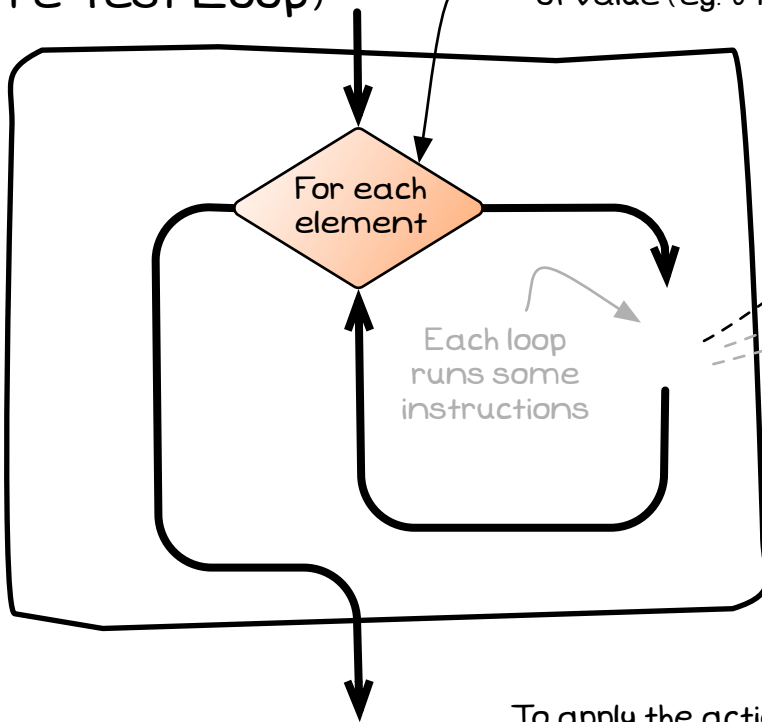
The array index is  
an Expression

Array index values can be calculated.  
If the value in  $i$  is 1, then this has the value  
2, indicating the 3rd element of the array

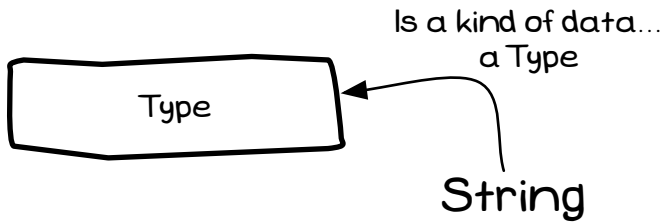
## For Loop (Pre-Test Loop)

Can be thought of  
as a counting loop,  
looping over a range  
of value (eg. 0 to 2)

Iterates over a  
range of values



To apply the actions to all elements in an  
array the for loop is used to count from the array's  
lowest index to its highest index.

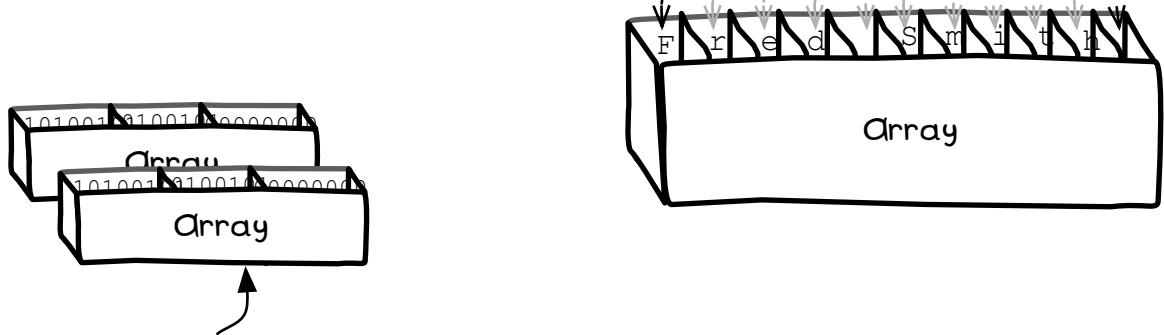


A String is text data,  
a sequence (string) of characters

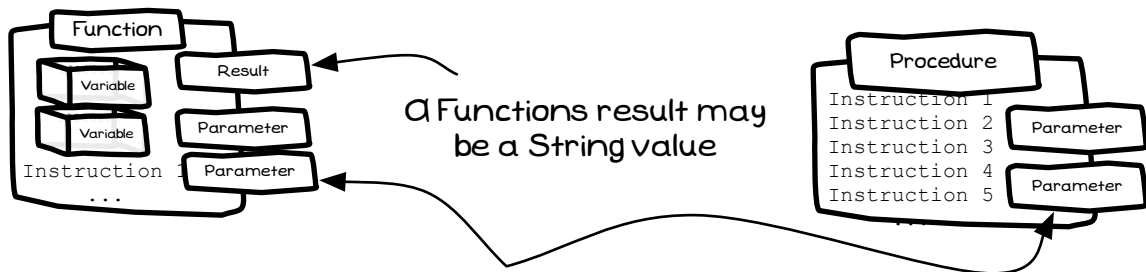
'Fred Smith'

Strings are stored  
as arrays  
of characters

One byte  
overhead

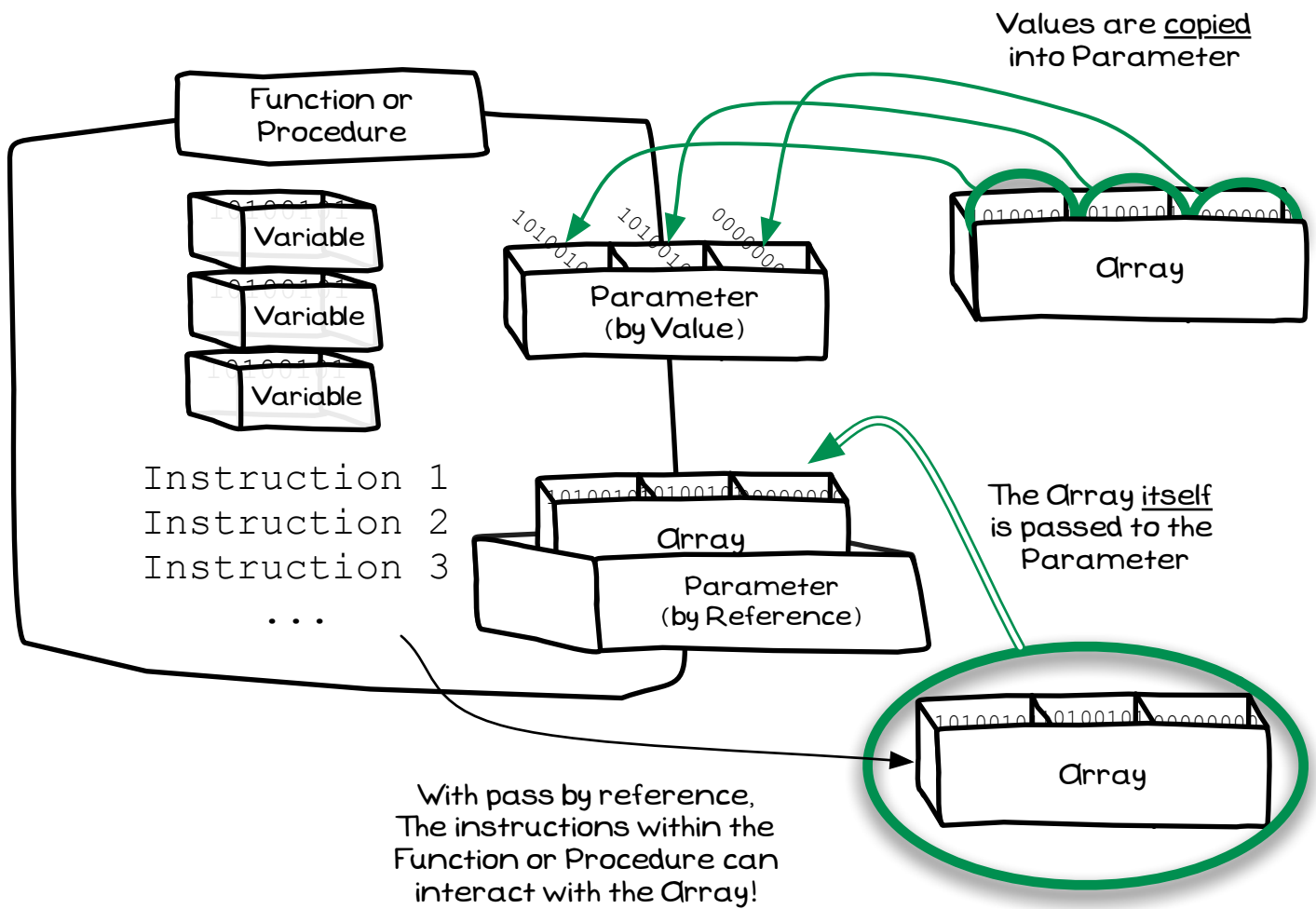


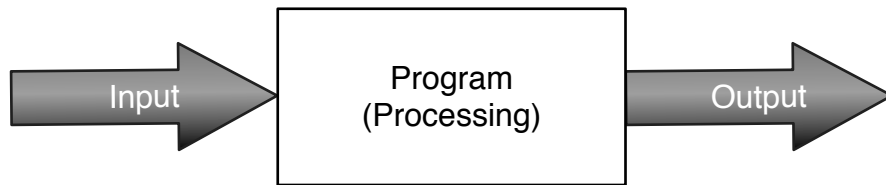
Variables can store  
String values



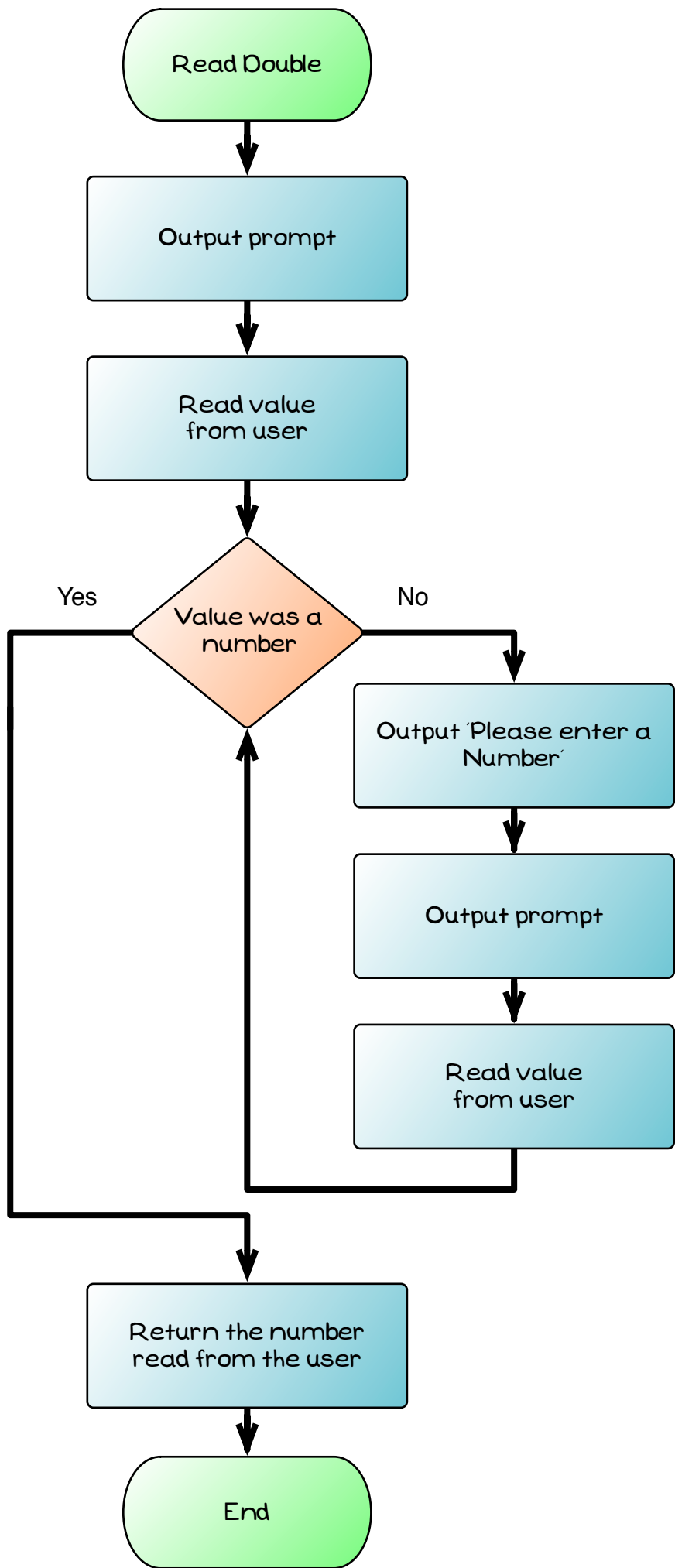
A Functions result may  
be a String value

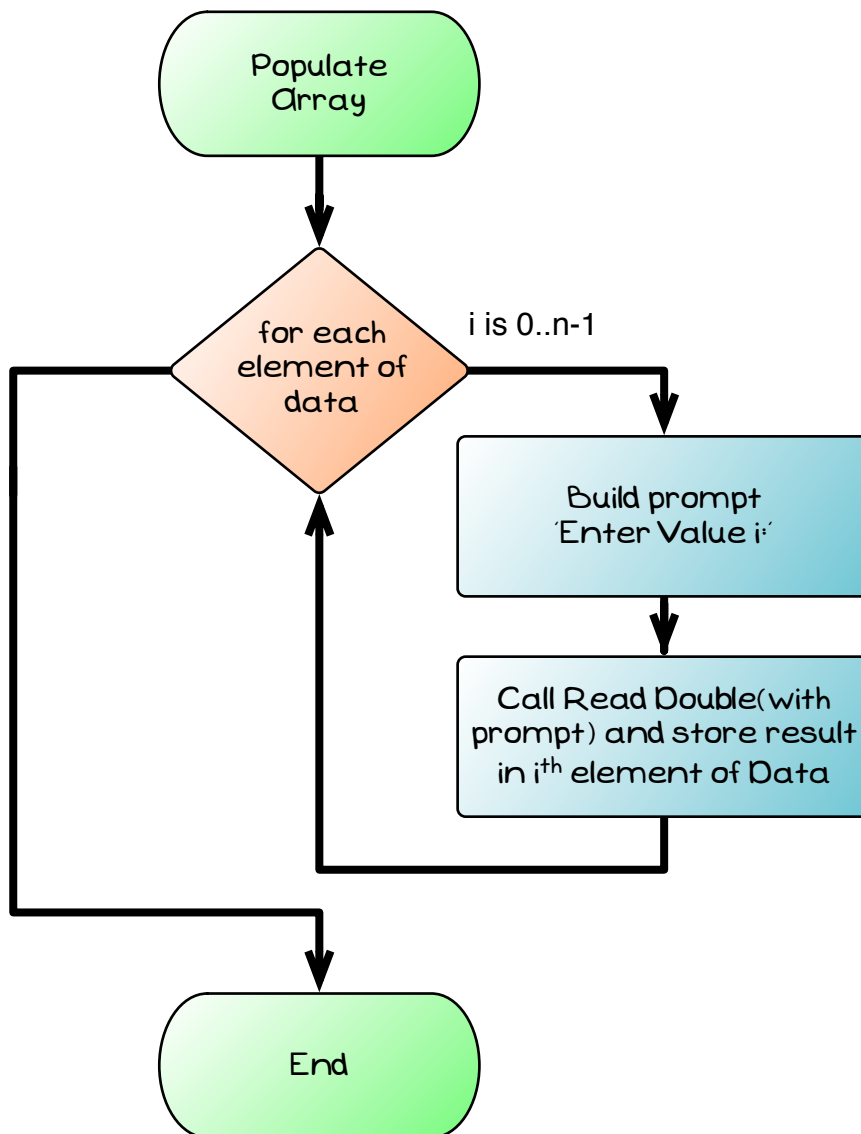
Parameters can accept  
String values

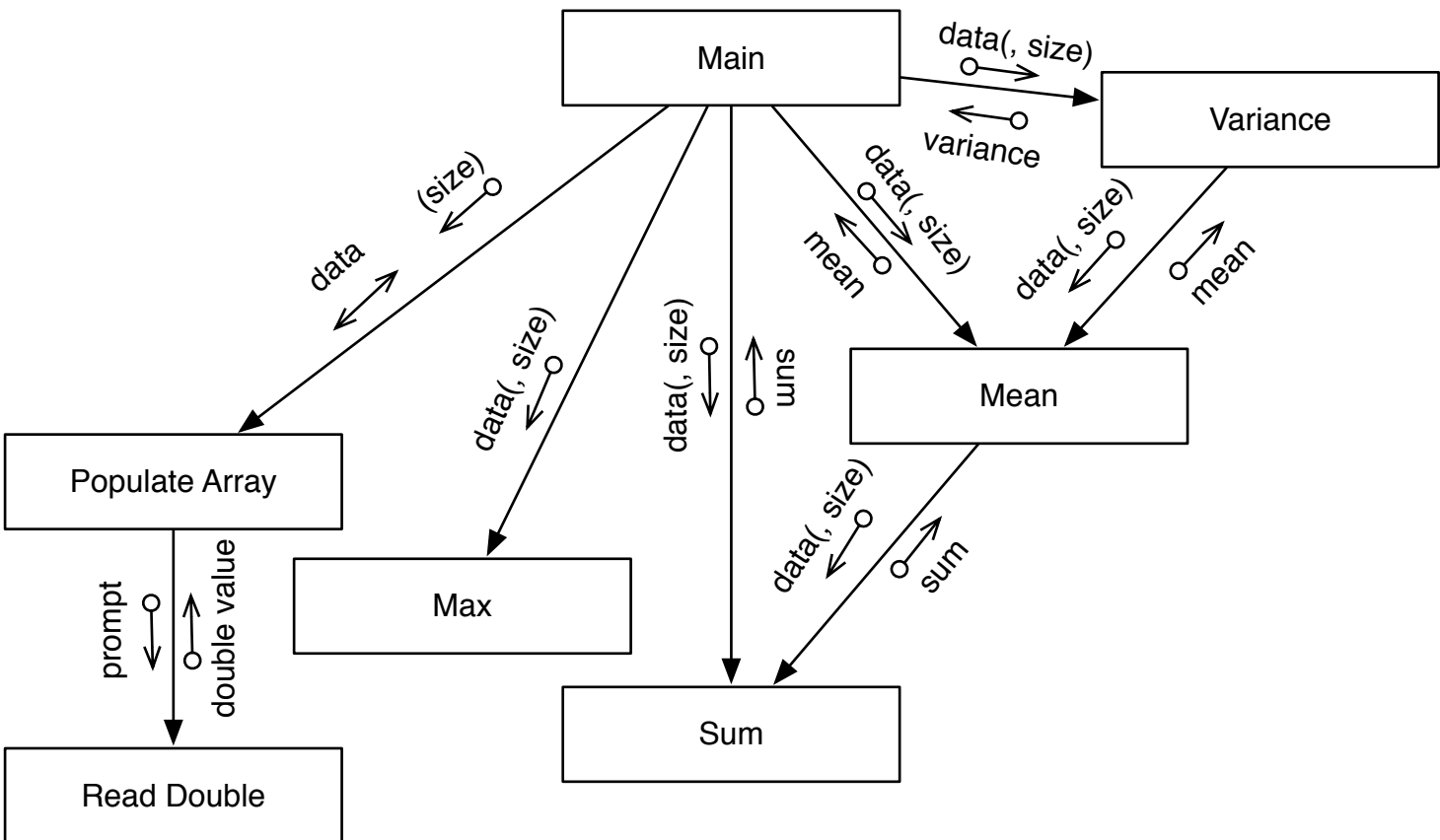


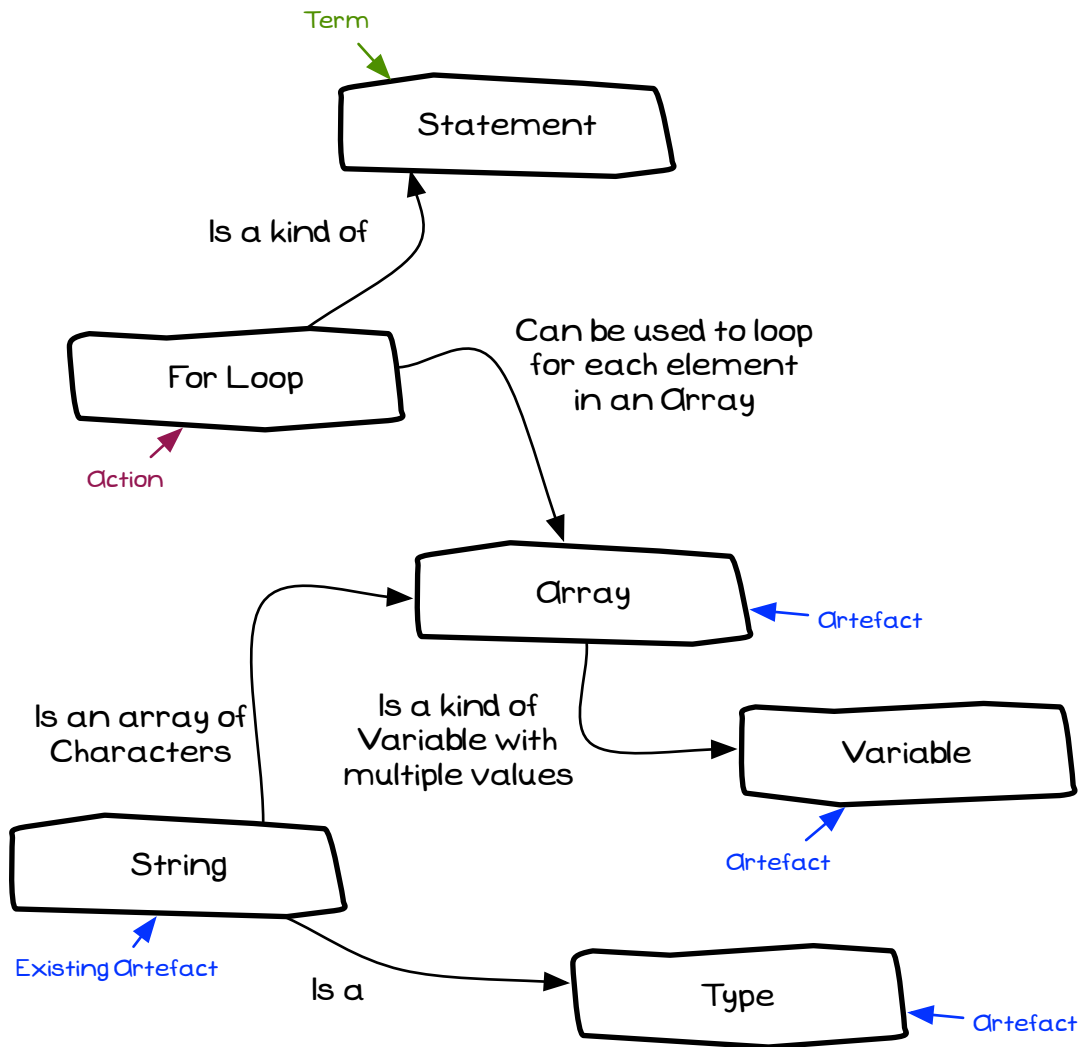














#### Procedure: Populate Array

##### Parameters:

- 1: data (by ref, array of Double) - the list of values to sum
- 2: size (Integer) - the number of elements in data (C only)

##### Local Variables:

- \*: i (Integer) - index of the current element in the array
- \*: prompt (String) - the prompt to be shown to the user

##### Steps:

- 1: For i, loops from lowest to highest index of data
- 2: Assign prompt, 'Enter value ' + (i + 1) + ': '
- 3: Assign data[i], the result of calling Read Double(prompt)

#### Procedure: Main

##### Local Variables:

- \*: my\_data (array containing 3 Double values) - data array

##### Steps:

- 1: Call Populate Array ( my\_data, 3 )

