

Is a value.

Expression

Part or all of the value can be read from variables and constants

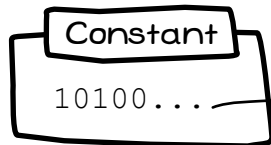
Calculations are evaluated first and then used

Calculations are evaluated using BODMAS order of operations

2 * PI * radius

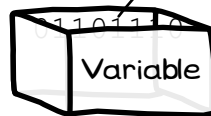
Reads value from the Constant

10100...



Reads value from the Variable

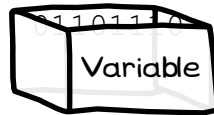
01101110



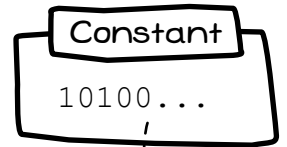
Is a value.

Expression

Part or all of the value can be Calculated by a Function, or read from Variables and Constants



Reads value from the Variable



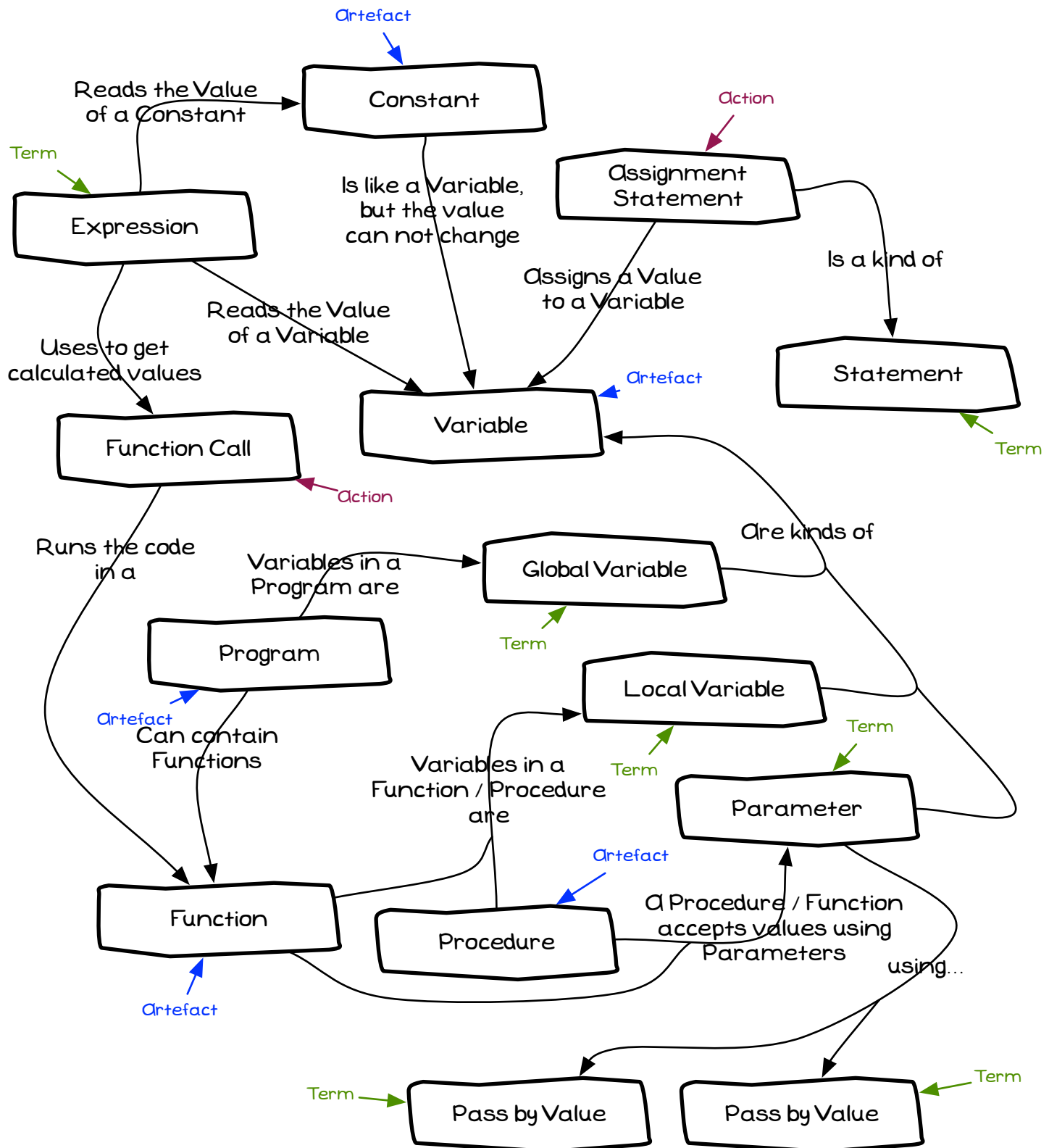
Reads value from the Constant

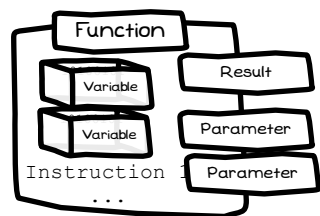
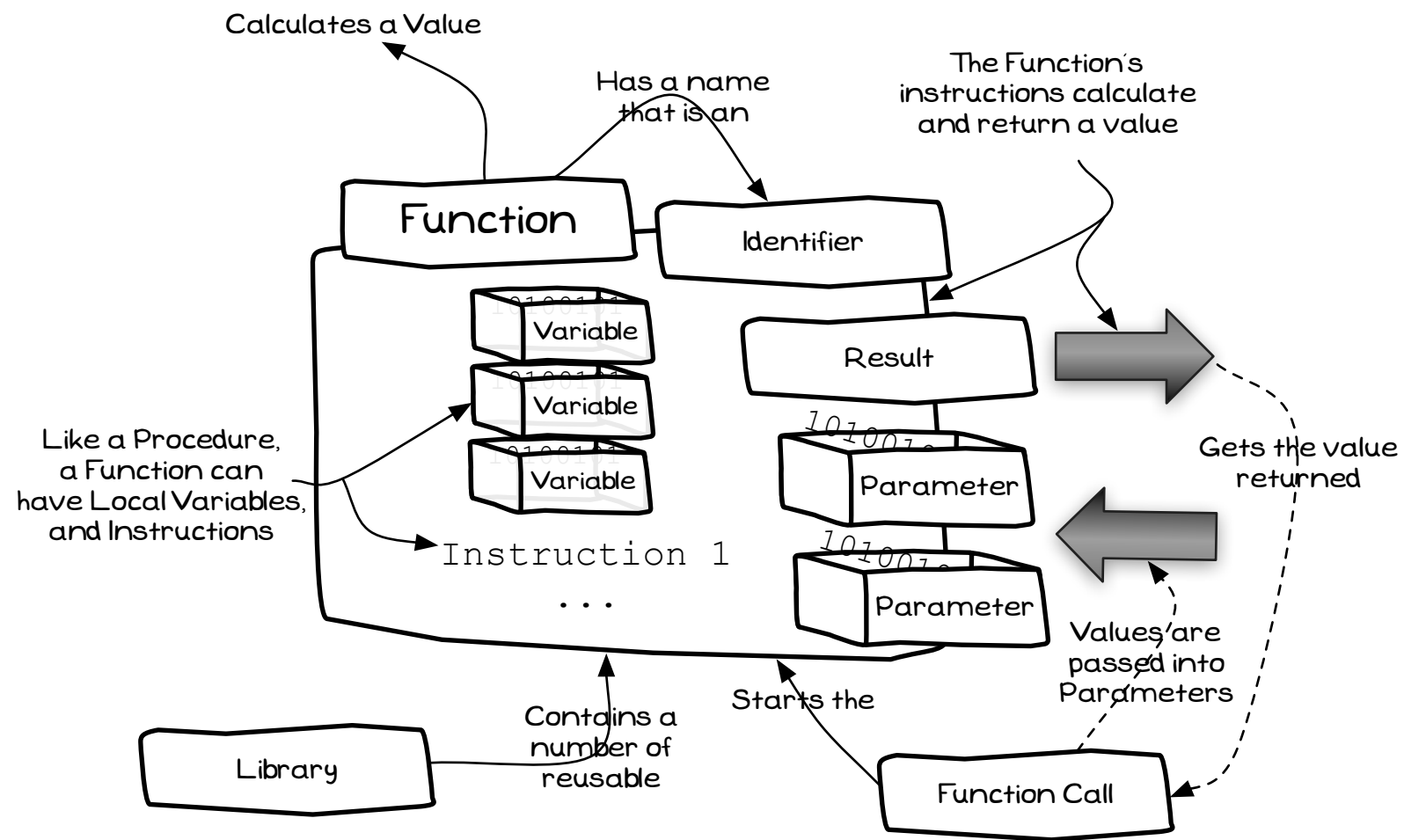
$\sin(\text{deg} * \text{PI} / 180)$

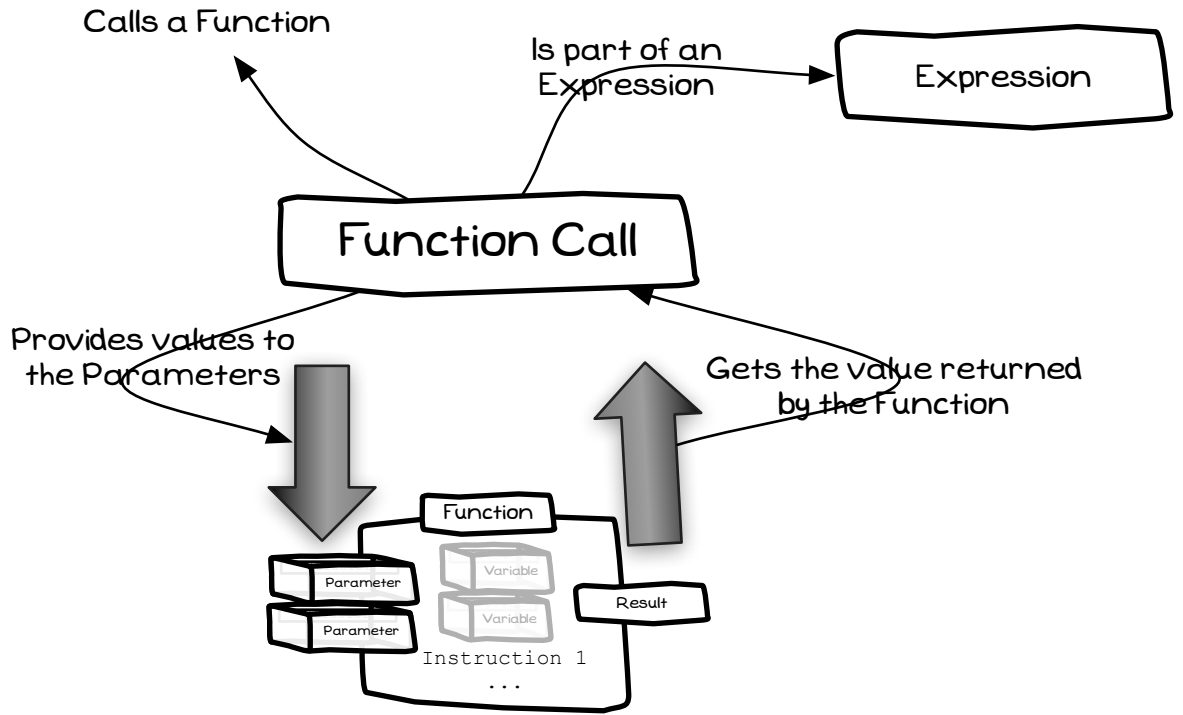
Calls the Function, and then reads its result

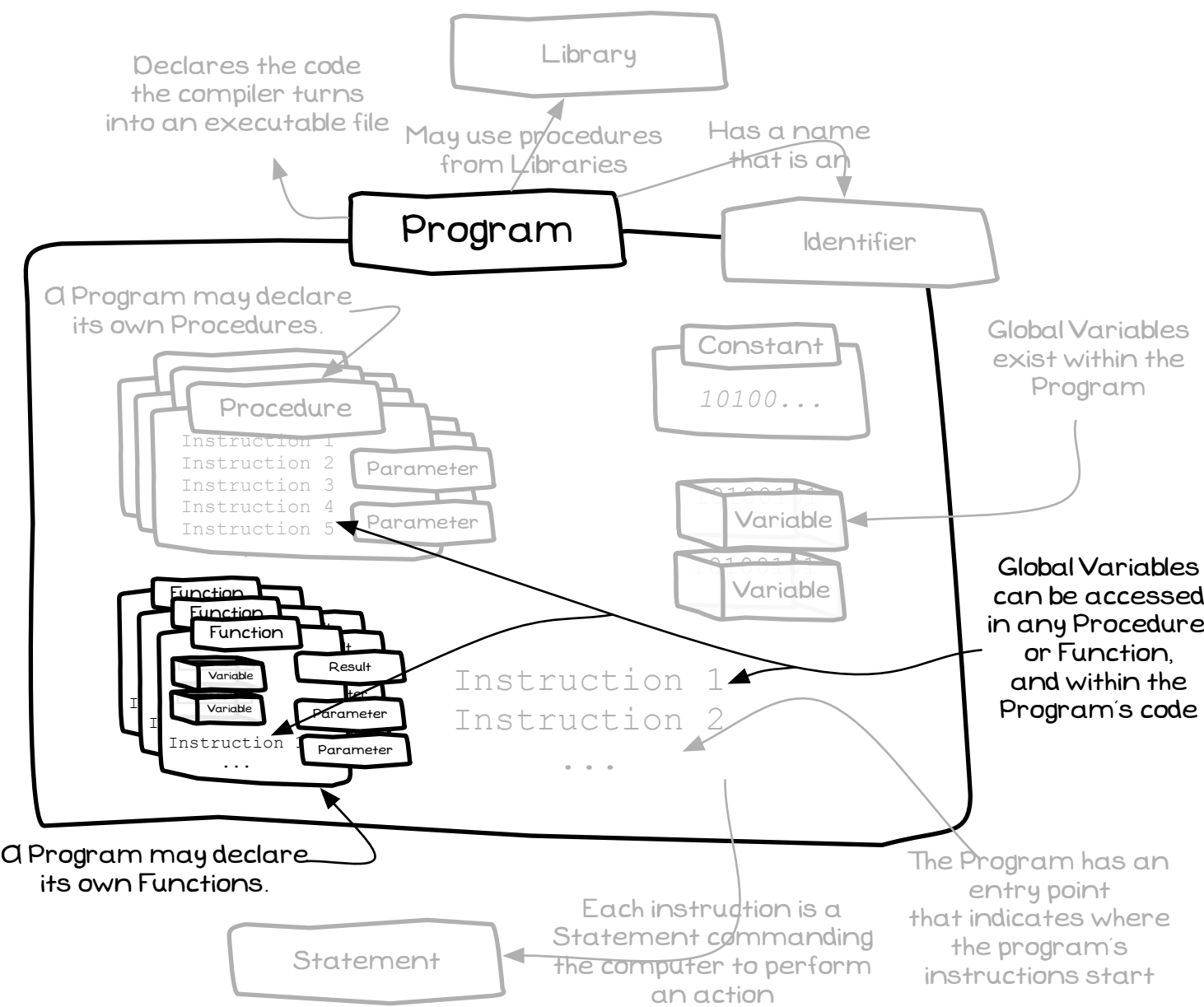
This expression is evaluated second. The `sin` Function is called to calculate the sine of 1.570796326794897. The `sin` Function will return the result 1.

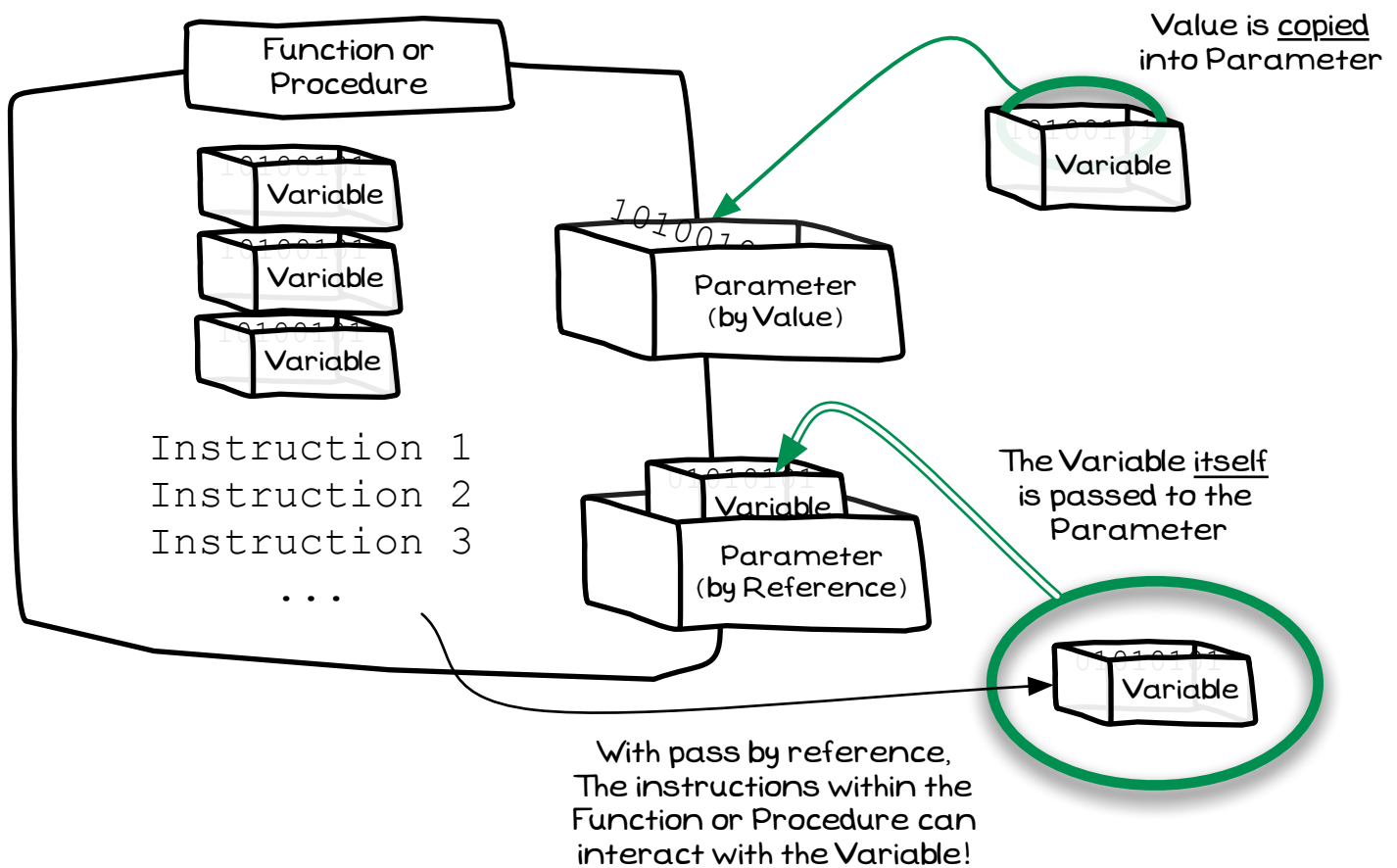
This expression is evaluated first. If the value in `deg` is 90 then this has the value 1.570796326794897 this value is passed to the first parameter in the `sin` Function.

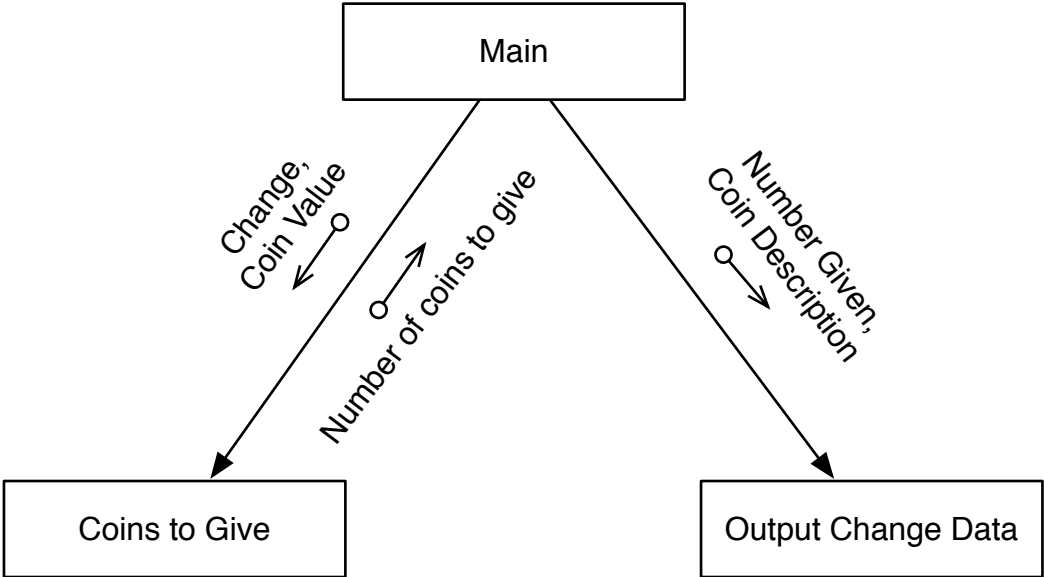


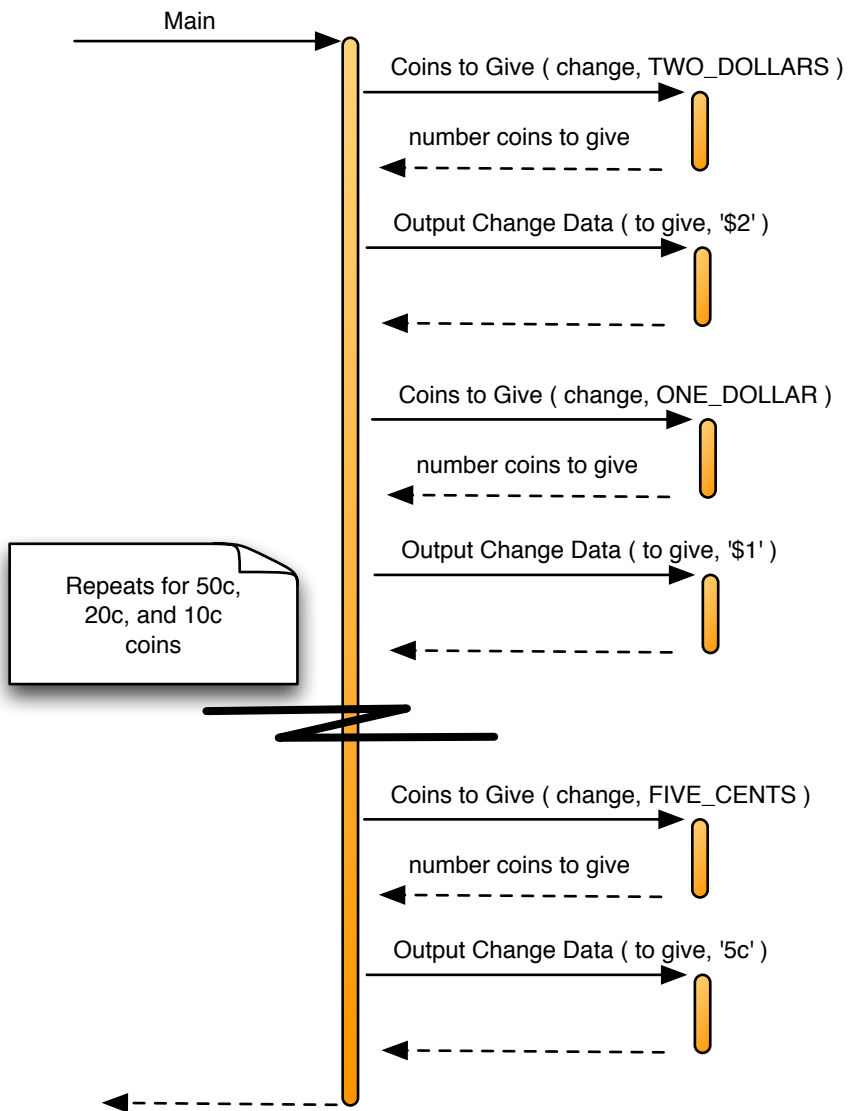












3

Values are read from the val and range variables.

Main

val: 23

range: 1.23

Current Instruction:

```
-----  
Procedure: Write Procedure  
-----  
...  
  
Program: Test Assignment  
-----  
Variables:  
* val (Integer)  
* range (Double)  
Steps:  
1: Assign to val, 23  
2: Assign to range, 1 + val / 100  
3: Output val and range to the Terminal
```

2

Expressions are evaluated first

4

Values read are passed to the Write Procedure

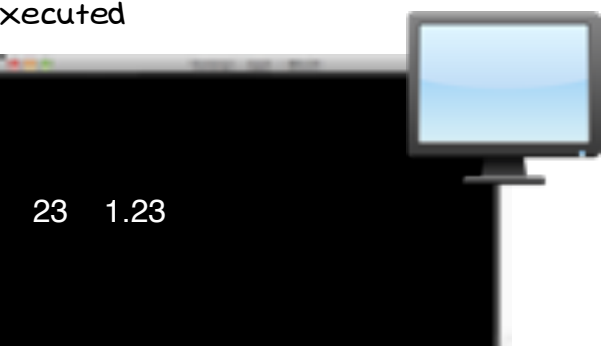
1

Third Instruction is executed



5

Output appears in Terminal



Test Params is added to the Stack. The val parameter get the value from the first argument.

3

Test Params
val: 1
local: 9823
Current Instruction:

Main
good: 1
Current Instruction:

Program: Test Globals, Locals, and Params

Global Variable:
* evil (Integer)

Procedure: Test Params

Parameters:

1: val (Integer)

Variables:

* local (Integer)

Steps:

- 1: Assign local, val - evil
- 2: Output val, local, and evil
- 3: Decrease val by 1
- 4: Decrease local by 1
- 5: Decrease evil by 1

Variables:

* good (Integer)

Steps:

- 1: Assign to evil, 1
- 2: Assign to good, 1
- 3: Call Test Params (good)
- 4: Output good and evil to the Terminal

To setup the call the Computer must evaluate the Expression and pass the value to the Parameter in Test Params

2

evil: 1

The Computer calls Test Params

1



Return value is stored
in the Function's result,
and the Function ends

