

Programs contain instructions,  
grouped into procedures

Developers organise program code using procedures to manage complexity

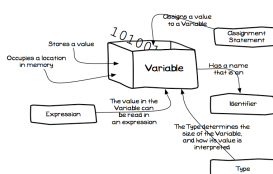
Even with procedures  
programs can only produce  
the one fixed outcome

Ideally programs should be able to respond to input, creating different output for different input values

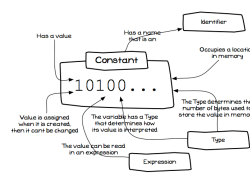
Dramatically enhance the utility of  
your programs by introducing  
artefacts to store data

## What artefacts exist to help you work with data?

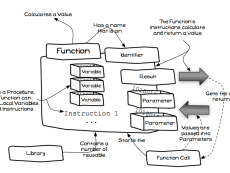
A variable is like a container where a single data value can be stored



A constant is like a variable, but its value cannot be changed

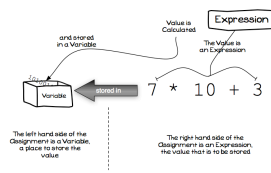


A function is like a procedure, except that it returns a value

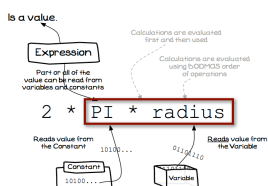


## What actions exist to work with variables?

The **assignment statement** is used to store a value in a variable

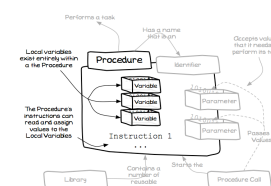


You use the name of the variable/constant to read its value

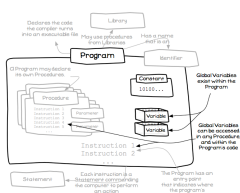


Where can variables and constants be created?

Variables can be coded in functions and procedures: these are called **local variables**.

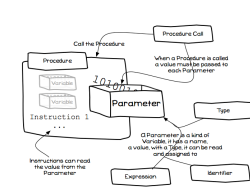


Variables and constants can also be declared in the program



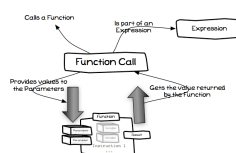
You need ways to share data  
between your functions and  
procedures

## You can give data to a Procedure using Parameters



## How do functions calculate a value?

A function call is like a procedure call, except it is used in an expression



Will you be able to use data to enhance your programs?

Without data, programs always behave the same, generating the same fixed outputs

Dramatically enhance the utility of  
your programs by introducing  
artefacts to store data

## Harness the number crunching power of the computer