



4.1.1 Local and Global Variables

Variables

A **variable** is a named storage location in memory that holds a value.

Variables can have different **scope**, which determines where they can be accessed in a program.

Local Variables

A local variable is **declared inside a function or procedure**.

It can only be used within that function and is destroyed when the function finishes executing.

```
def calculate_area():  
    length = 5      # Local variable  
    width = 4       # Local variable  
    area = length * width  
    return area
```

After `calculate_area()` returns, the variables `length`, `width` and `area` no longer exist.

Global Variables

A global variable is **declared outside all functions**.

It can be accessed by any function in the program and exists the whole time the program is running.

```
length = 10        # Global variable  
width = 8          # Global variable  
  
def calculate_area():  
    area = length * width  
    return area
```

After `calculate_area()` returns, the variables `length`, `width` still exist.





Local vs Global Variables

Local Variable	Global Variable
Reduce the chance of accidental changes from other parts of the program.	Allow data to be shared between multiple functions.
Make functions independent and easier to test.	Avoid passing the same variable as a parameter repeatedly.
Use memory only while the function is running.	Useful for values that should be available throughout the program (e.g. program settings).
Improve readability by keeping data relevant to one function.	

