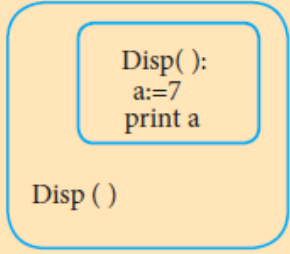


# Types of Variable Scope with Example

## 1. Local Scope

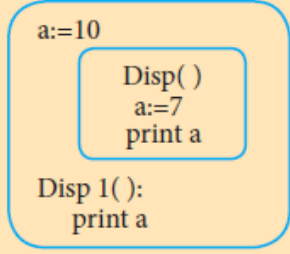
Local scope refers to variables defined in current function. Always, a function will first look up for a variable name in its local scope. Only if it does not find it there, the outer scopes are checked.

Look at this example

	Entire program	Output of the Program
1. <b>Disp():</b>		7
2. a:=7		
3. print a		
4. Disp()		

## 2. Global Scope

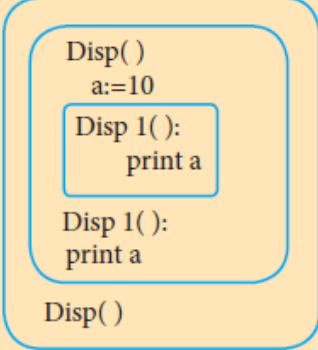
A variable which is declared outside of all the functions in a program is known as global variable. This means, global variable can be accessed inside or outside of all the functions in a program. Consider the following example

	Entire program	Output of the Program
1. a:=10		7 10
2. <b>Disp():</b>		
3. a:=7		
4. print a		
5. Disp()		
6. print a		

### 3. Enclosed Scope

A function (method) within another function is called a nested function. A variable which is declared inside a function which contains another function definition within it, the inner function can also access the variable of the outer function. This scope is called enclosed scope.

When a compiler or interpreter searches for a variable in a program, it first searches Local, and then searches Enclosing scopes. Consider the following example

	Entire program	Output of the Program
1. Disp():		
2. a:=10		10
3. Disp1():		10
4. print a		
5. Disp1()		
6. print a		
7. Disp()		

### 4. Built-in Scope

5.

The built-in scope has all the names that are pre-loaded into the program scope when we start the compiler or interpreter. Any variable or module which is defined in the library functions of a programming language has Built-in or module scope. They are loaded as soon as the library files are imported to the program

