

# Procedures

## Procedures

### Introduction to Procedure

A procedure is a block of code that performs a specific task. Procedures allow you to organize your code into manageable sections, make it reusable, and avoid repetition.

### How to Create a Procedure

Create a standard module and either type the procedure manually or use the Insert menu: click "Procedure", enter a name, choose the type (Sub, Function, or Property), select the scope (Public or Private), and click OK.

### Naming Rules for Procedures

- Must start with a letter (A-Z or a-z)
- Can contain letters, numbers, and underscores (\_)
- Cannot contain spaces or special characters (@, #, \$, %, &, \*, etc.)
- Must be unique within a VBA project
- Can be up to 255 characters long (shorter names are recommended for readability)

**Note:** Treat these rules as general naming conventions in VBA, whether you're naming modules, variables, procedures, or other elements.

### Types of Procedure

**Subroutines (Sub):** A subroutine, or "Sub" procedure, is a block of VBA code that performs a series of actions but does not return a value. Subs can accept arguments (optional or required) and can be executed using the Run button in the VBA editor, a keyboard shortcut, or by calling it in another procedure.

```
Sub ProcedureName(Optional arguments As DataType)
    ' Your code here
End Sub
```

**Functions:** A Function is a procedure that performs a task or a calculation and returns a value. Functions are often used to calculate and return results and can be used within other procedures. Function procedures can be used to create custom Excel formulas to extend Excel's functionality

```
Function FunctionName(arguments As DataType) As ReturnType
    ' Your code here
    FunctionName = Result
End Function
```

**Property Procedures:** Property procedures allow you to define custom properties for objects. These properties can be read-only, write-only, or read/write. These procedures are used mainly in class modules to define object properties and are useful for advanced programming and object-oriented design. Types of "Property Procedures" are:

1. Property Let: Assigns a value to a property.
2. Property Get: Retrieves the value of a property.
3. Property Set: Assigns an object reference to a property.

The scope of a procedure determines where it can be accessed. Procedures in a worksheet or workbook module are tied to specific events and cannot be called directly from other modules.

### Scope of Procedure in Standard Module

**Private Procedures:** Accessible only within the module in which they are defined. "Private Procedures" are defined using the "Private" keyword. Example:

```
Private Sub SecretMessage()
    MsgBox "This is private!"
End Sub
```

**Public Procedures:** These procedures are accessible from anywhere within the workbook or even from other workbooks (if referenced). Public Procedures are defined using the "Public" keyword (or no keyword, as "Public" is the default for modules). Example:

```
Public Sub HelloWorld()
    MsgBox "Hello, World!"
End Sub
```

### Procedure Arguments

**Required Arguments:** Must be provided when the procedure is called.

```
Sub GreetUser(name As String)
    MsgBox "Hello, " & name
End Sub
```

Call:  
GreetUser "Hamed"

**Optional Arguments:** Do not need to be provided; you can define a default value.

```
Sub GreetUser(Optional name As String = "Guest")
    MsgBox "Hello, " & name
End Sub
```

**ByRef (default):** The argument is passed by reference, meaning changes affect the original variable.  
**ByVal:** The argument is passed by value, meaning changes do not affect the original variable.

```
Sub DoubleValue(ByRef x As Integer)
    x = x * 2
End Sub
```