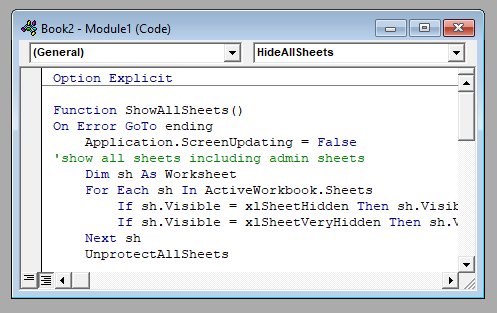
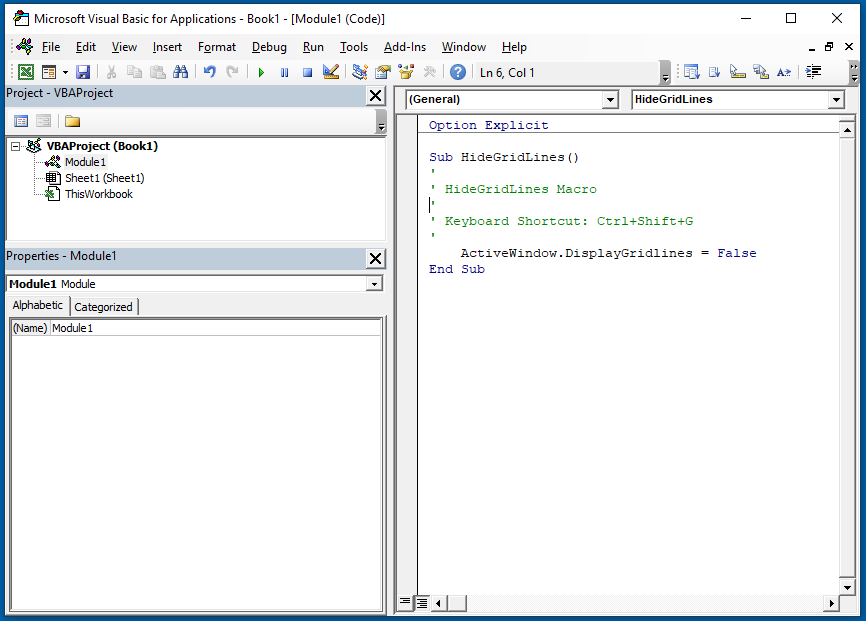
1. What are modules in VBA and describe in detail the importance of

creating a module?

A VBA module is used to store any VBA code that you have written in the VBE (Visual Basic Editor).

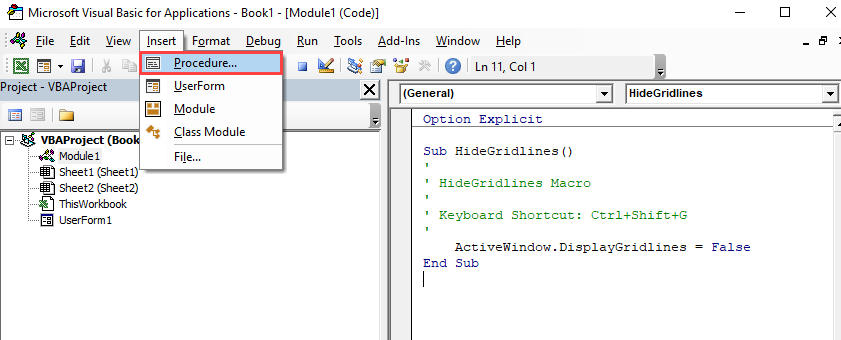


The modules are contained within a VBA Project and when the file is saved – be it an Excel workbook, Word document or Access database, the module or modules are saved within that file – that file is essentially the parent application of the module.

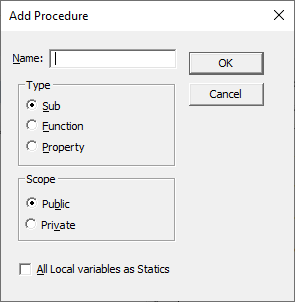


Modules can also be exported out of the parent file and saved as their own individual files.  This is useful when you want to re-use code in a different file, and therefore perhaps import that module into a new file.

The easiest way to create a procedure is by typing directly into the Module Window, but if you forget how to create a procedure, you can go to **Insert > Procedure** instead:



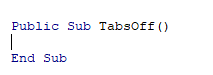
The following dialog box will appear



This dialog box is a good way to learn about the options for Procedures.

* Type the name of your new procedure in the name box – this must start with a letter of the alphabet and can contain letters and number and be a maximum of 64 characters.
* You can have a **Sub procedure**, a **Function procedure** or a Property procedure. *(Properties are used in Class modules and set properties for ActiveX controls that you may have created).*
* You can make the scope of the procedure either **Public** or **Private**. If the procedure is public (default), then it can be used by all the modules in the project while if the procedure is private, it will only be able to be used by this module.
* You can declare local variables in this procedure as Statics (this is to do with the Scope of the variable and makes a local procedure level variable public to the entire module). We will not use this option.

When you have filled in all the relevant details, click on **OK**.



You then type your code between the Sub and End Sub statements.

**Sub vs Function Procedures**

You may have noticed that there are 2 types of procedures you can create – a **SUB PROCEDURE** or a **FUNCTION PROCEDURE**.

If you have recorded a macro in Excel or Word, your macro will be put into a Standard module and will be put into a sub procedure. The macro recorder can only record sub procedures.  A Sub procedure does things.  They perform actions such as formatting a table, creating a pivot table, or changing the view settings of your active window.  The majority of procedures written are Sub procedures.  **All macros are Sub procedures**.

A Function procedure returns a value.  This value may be a single value, an array, a range of cells or an object.  Functions usually perform some type of calculation.   Functions in Excel (UDFs) can be used with the Function Wizard.

2. What is Class Module and what is the difference between a Class

Module and a Module?

Comparison between Module and Class:

|  |  |  |
| --- | --- | --- |
|  | **Module** | **Class** |
| Inheritance | Not Supported | supported |
| Implement interfaces | No | Yes |
| Members | Shared (implicitly in Visual Basic) | Control the type of a member of a class as a shared or instance member |
| Object oriented | No, modules cannot be instantiated | Yes |
| Copy of the data | Only one copy | Object data exists separately for each instantiated object. |
| Typical application | Generally, used at the outer level for separating compilation, inner structures and privacy outside of the module boundaries | Generally as components of modules |
| Preferred | For abstract datatypes used with binary operations, such as sets with a merge operation | When inheritance is required |
| Members | Methods, constants and classes | Methods, constants and variables |
| Inclusion | Can be included by using include commands in classes or modules | Cannot be included |
| Super class | Object | Module |

3. What are Procedures? What is a Function Procedure and a Property

Procedure?

A procedure is a method for completing something with steps and instructions for each aspect of the task. Procedures often stay consistent once they're established to maintain a general understanding of what employees should do in various circumstances.

A Function procedure is a series of Visual Basic statements enclosed by the Function and End Function statements. The Function procedure performs a task and then returns control to the calling code. When it returns control, it also returns a value to the calling code.

Each time the procedure is called, its statements run, starting with the first executable statement after the Function statement and ending with the first End Function, Exit Function, or Return statement encountered.

You can define a Function procedure in a module, class, or structure. It is Public by default, which means you can call it from anywhere in your application that has access to the module, class, or structure in which you defined it.

A Function procedure can take arguments, such as constants, variables, or expressions, which are passed to it by the calling code

A property procedure is a series of Visual Basic statements that manipulate a custom property on a module, class, or structure. Property procedures are also known as *property accessors*.

Visual Basic provides for the following property procedures:

* A Get procedure returns the value of a property. It is called when you access the property in an expression.
* A Set procedure sets a property to a value, including an object reference. It is called when you assign a value to the property.

4. What are Procedures? What is a Function Procedure and a Property

Procedure?

A procedure is a method for completing something with steps and instructions for each aspect of the task. Procedures often stay consistent once they're established to maintain a general understanding of what employees should do in various circumstances.

A Function procedure is a series of Visual Basic statements enclosed by the Function and End Function statements. The Function procedure performs a task and then returns control to the calling code. When it returns control, it also returns a value to the calling code.

Each time the procedure is called, its statements run, starting with the first executable statement after the Function statement and ending with the first End Function, Exit Function, or Return statement encountered.

You can define a Function procedure in a module, class, or structure. It is Public by default, which means you can call it from anywhere in your application that has access to the module, class, or structure in which you defined it.

A Function procedure can take arguments, such as constants, variables, or expressions, which are passed to it by the calling code

A property procedure is a series of Visual Basic statements that manipulate a custom property on a module, class, or structure. Property procedures are also known as *property accessors*.

Visual Basic provides for the following property procedures:

* A Get procedure returns the value of a property. It is called when you access the property in an expression.
* A Set procedure sets a property to a value, including an object reference. It is called when you assign a value to the property.

5. What is a sub procedure and what are all the parts of a sub procedure

and when are they used?

A Sub procedure is a series of Visual Basic statements enclosed by the Sub and End Sub statements. The Sub procedure performs a task and then returns control to the calling code, but it does not return a value to the calling code.

Each time the procedure is called, its statements are executed, starting with the first executable statement after the Sub statement and ending with the first End Sub, Exit Sub, or Return statement encountered.

You can define a Sub procedure in modules, classes, and structures. By default, it is Public, which means you can call it from anywhere in your application that has access to the module, class, or structure in which you defined it. The term *method* describes a Sub or Function procedure that is accessed from outside its defining module, class, or structure. For more information, see Procedures.

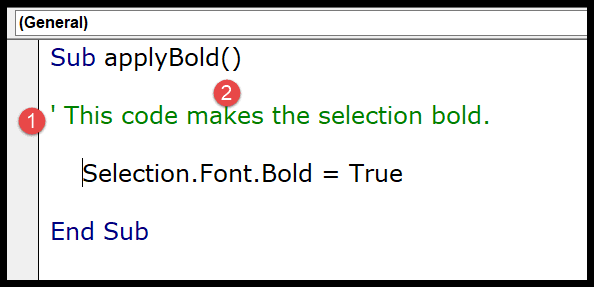
A Sub procedure can take arguments, such as constants, variables, or expressions, which are passed to it by the calling cod

7. How do you add comments in a VBA code? How do you add multiple

lines of comments in a VBA code?

**Add a Comment in a VBA Code**

Steps you need to follow to add a comment in a VBA code:

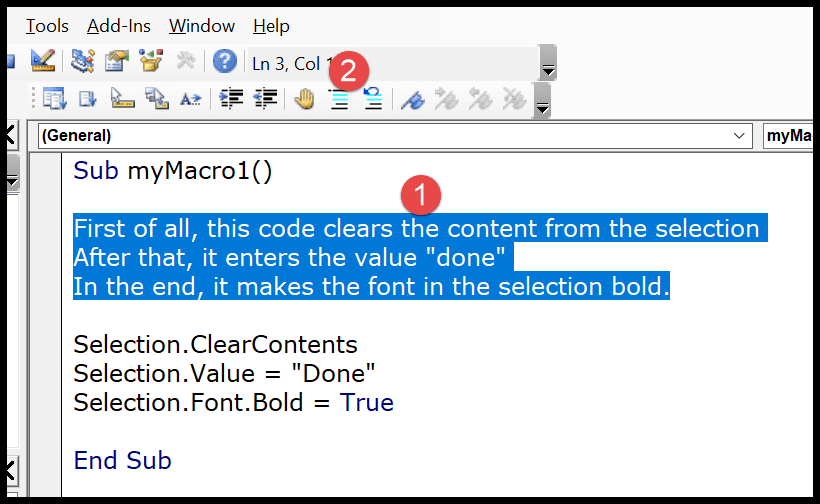
* First, **click on the line** where you want to insert the comment.
* After that, **type an APOSTROPHE** using your keyboard key.
* Next, **type the comment** that you want to add to the code.
* In the end, **hit enter** to move to the new line and the comment will turn green.
* 
* The moment you do this the entire line of the code will turn green which means that line is comment now.
* If you look at the below code where I have used a comment to add a description of the procedure.
* You simply need to add an apostrophe before turning it into a comment and VBA will ignore it while executing the code

## Enter a Multi-Line VBA Comment

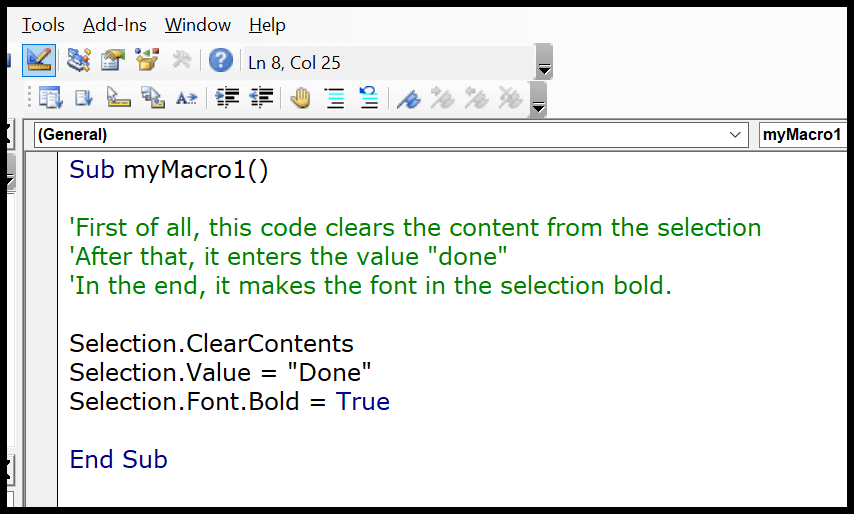
There could be a situation where you need to enter a comment in multiple lines, like a block of the comments.

**But here is**one thing which you need to note down, every line of comment needs to start with an apostrophe, so if you want to add multiple lines of comments every line should have an APOSTROPHE.

The easiest way is to select all the lines and then use the comment button from the toolbar or you can also add an APOSTROPHE at the starting of each line.



The moment you click the comment button it will convert all the lines into a multi-line comment block.

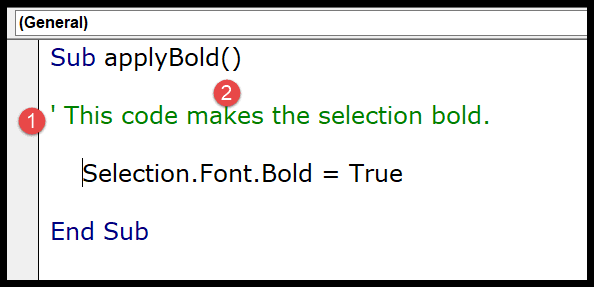


6. How do you add comments in a VBA code? How do you add multiple

lines of comments in a VBA code?

**Add a Comment in a VBA Code**

Steps you need to follow to add a comment in a VBA code:

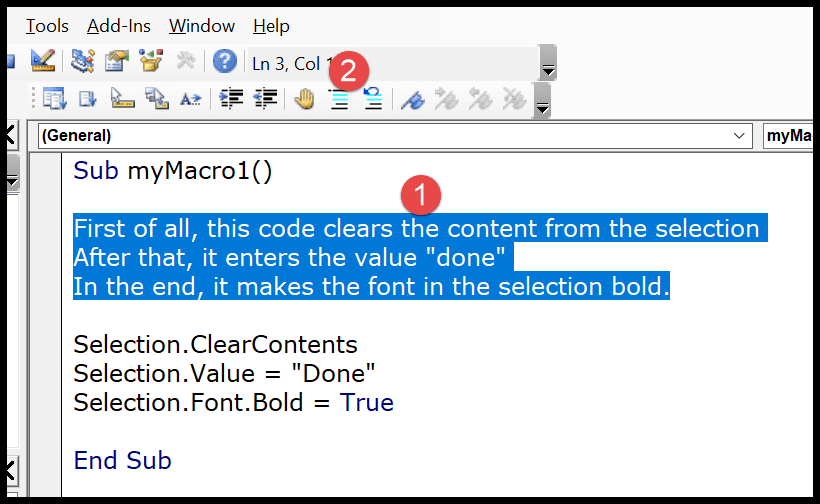
* First, **click on the line** where you want to insert the comment.
* After that, **type an APOSTROPHE** using your keyboard key.
* Next, **type the comment** that you want to add to the code.
* In the end, **hit enter** to move to the new line and the comment will turn green.
* 
* The moment you do this the entire line of the code will turn green which means that line is comment now.
* If you look at the below code where I have used a comment to add a description of the procedure.
* You simply need to add an apostrophe before turning it into a comment and VBA will ignore it while executing the code

## Enter a Multi-Line VBA Comment

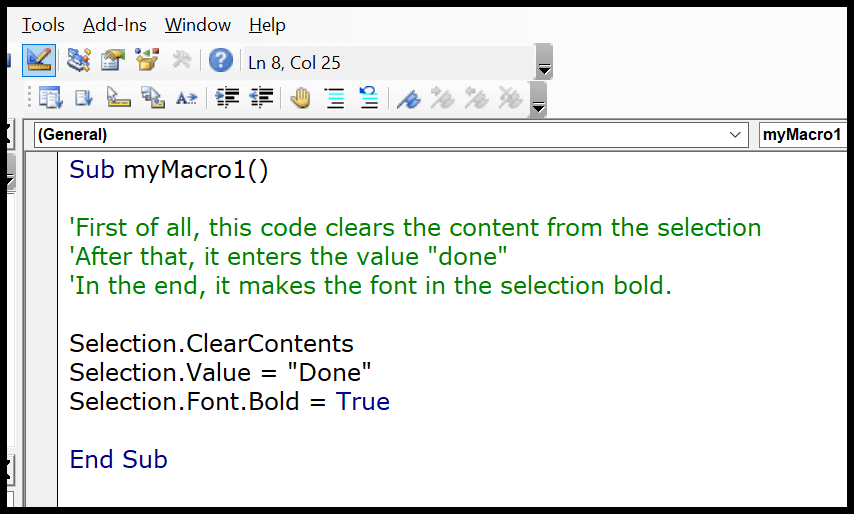
There could be a situation where you need to enter a comment in multiple lines, like a block of the comments.

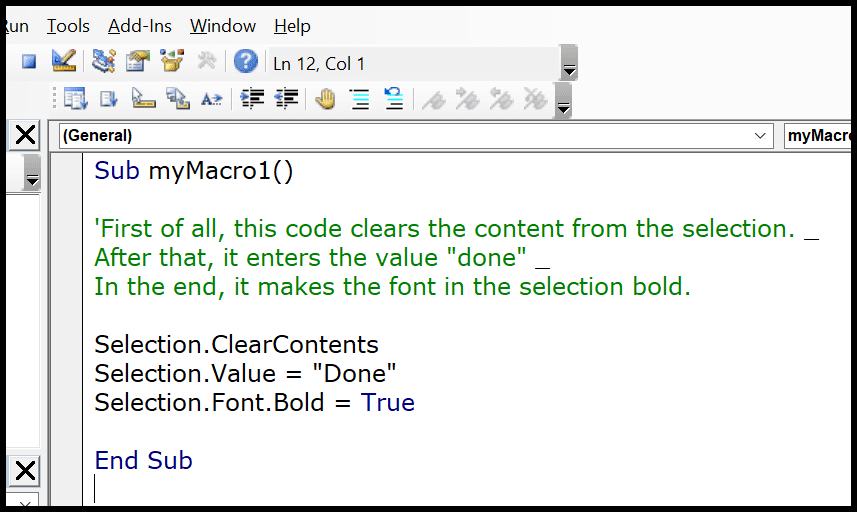
**But here is**one thing which you need to note down, every line of comment needs to start with an apostrophe, so if you want to add multiple lines of comments every line should have an APOSTROPHE.

The easiest way is to select all the lines and then use the comment button from the toolbar or you can also add an APOSTROPHE at the starting of each line.

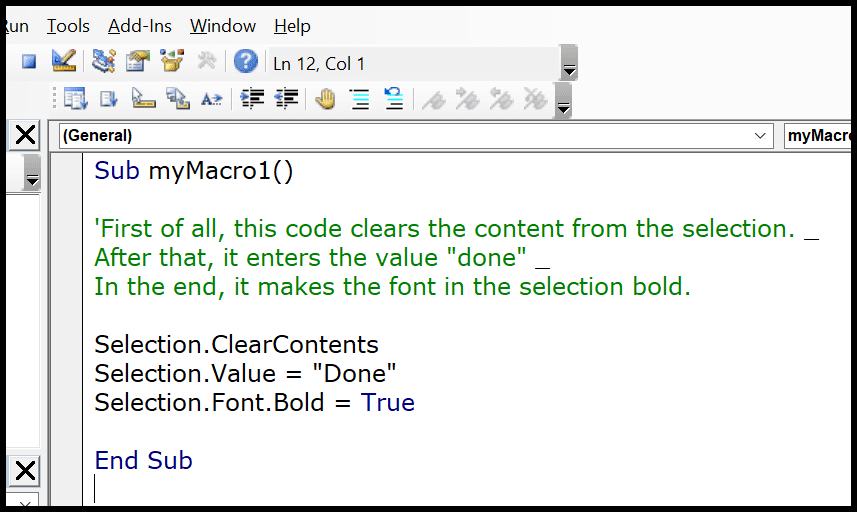


The moment you click the comment button it will convert all the lines into a multi-line comment block.





In the above example, I have used an apostrophe only at the start of the first line of the comment, the rest two-line don’t have an apostrophe but I have used line continuation character to give a line break at the end of the first line and the second line.



In the above example, I have used an apostrophe only at the start of the first line of the comment, the rest two-line don’t have an apostrophe but I have used line continuation character to give a line break at the end of the first line and the second line.