

Q1. What are modules in VBA and describe in detail the importance of creating a module?

Ans: 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 modules are organised into 3 different types.

1. **Standard modules** – most of your code will go into this type of module. When you record a macro, it gets put into a standard module. When you write a general procedure to be used throughout your workbook, it also normally goes into a standard module.
2. **Object modules** – these type of modules hold the code that is unique to that individual workbook or worksheet. Most of the code in these type of modules are known as events. An event can occur when a workbook is opened or closed for example, or when a sheet is clicked (the Click Event),. The module can also contain code that is written by yourself and used by the events. The module behind a custom form that you create is also an Object module.
3. **Class modules** – this module is used to create objects at run time. Class modules are used by Advanced VBA programmers and will be covered at a later stage.

When you write your VBA code, you will usually use more than one module. It is good coding practice to 'group' your code into relevant modules – for example put all the global variables in one module, all public functions in another module etc.

Q2. What is Class Module and what is the difference between a Class Module and a Module?

Ans: VBA Class Modules allow the user to create their own objects. If you are not familiar with objects then I would highly recommend that you first check out my previous post [VBA Objects – The Ultimate Guide](#).

In languages such as C# and Java, classes are used to create objects. Class Modules are the VBA equivalent of these classes. The major difference is that VBA Class Modules have a very limited type of Inheritance* compared to classes in the other languages. In VBA, Inheritance works in a similar way to Interfaces** in C#\Java. In VBA we have built-in objects such as the Collection, Workbook, Worksheet and so on. The purpose of VBA Class Modules is to allow us to custom build our own objects.

Difference between a Class Module and a Module:

As you probably know a standard module can store procedures and functions which can be either Private or Public (the default) and can be accessed either from within that module only (Private) or from anywhere in the project (Public). Modules are also used to store variables, constants and declarations (i.e. API calls) that will need to be accessed from anywhere in the project.

Class modules allow you to create your own objects which can have their own properties and methods like any other object (range, worksheet, Excel, chart, blah, blah). The best way of describing is by use of a simple example. Say you wanted to have code that would allow you to create an invoice object, assign it a number, customer name, net amount, and VAT. You'd then want to save that invoice as a new workbook.

Q3. What are Procedures? What is a Function Procedure and a Property Procedure?

Ans: A procedure is a program code that can carry out certain tasks or return a value. It can be called from other procedures. In Visual Basic 2017, there are two types of procedures; sub procedures and functions. A sub procedure(also call subroutine) is a procedure that performs a specific task and does not return a value while a function is a procedure that returns a value.

Function Procedure

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.

Property Procedure

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.

You usually define property procedures in pairs, using the Get and Set statements, but you can define either procedure alone if the property is read-only (Get Statement) or write-only (Set Statement).

You can omit the Get and Set procedure when using an auto-implemented property. For more information, see Auto-Implemented Properties.

Q4. What are Procedures? What is a Function Procedure and a Property Procedure?

Ans: A procedure is a program code that can carry out certain tasks or return a value. It can be called from other procedures. In Visual Basic 2017, there are two types of procedures; sub procedures and functions. A sub procedure(also call subroutine) is a procedure that performs a specific task and does not return a value while a function is a procedure that returns a value.

Function Procedure

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.

Property Procedure

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.

You usually define property procedures in pairs, using the Get and Set statements, but you can define either procedure alone if the property is read-only (Get Statement) or write-only (Set Statement).

You can omit the Get and Set procedure when using an auto-implemented property. For more information, see Auto-Implemented Properties.

Q5. What is a sub procedure and what are all the parts of a sub procedure and when are they used?

Ans: 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.

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

A sub procedure is usually used to accept input from the user, display information, print information, manipulate properties or perform some other tasks. It is a program code by itself and it is not an event procedure because it is not associated with a runtime procedure. It is called by other code whenever it is required to perform a certain task.

Sub procedures help to make programs smaller and seamless to manage. A sub procedure begins with a Sub keyword and ends with an End Sub keyword.

Q6. How do you add comments in a VBA code? How do you add multiple lines of comments in a VBA code?

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

1. First, click on the line where you want to insert the comment.
2. After that, type an APOSTROPHE using your keyboard key.
3. Next, type the comment that you want to add to the code.
4. 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.

So you simply need to add an apostrophe before turning it into a comment and VBA will ignore it while executing the code.

Multiple lines of comments

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.

Q7. How do you add comments in a VBA code? How do you add multiple lines of comments in a VBA code?

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

1. First, click on the line where you want to insert the comment.
2. After that, type an APOSTROPHE using your keyboard key.
3. Next, type the comment that you want to add to the code.
4. 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.

So you simply need to add an apostrophe before turning it into a comment and VBA will ignore it while executing the code.

Multiple lines of comments

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.

