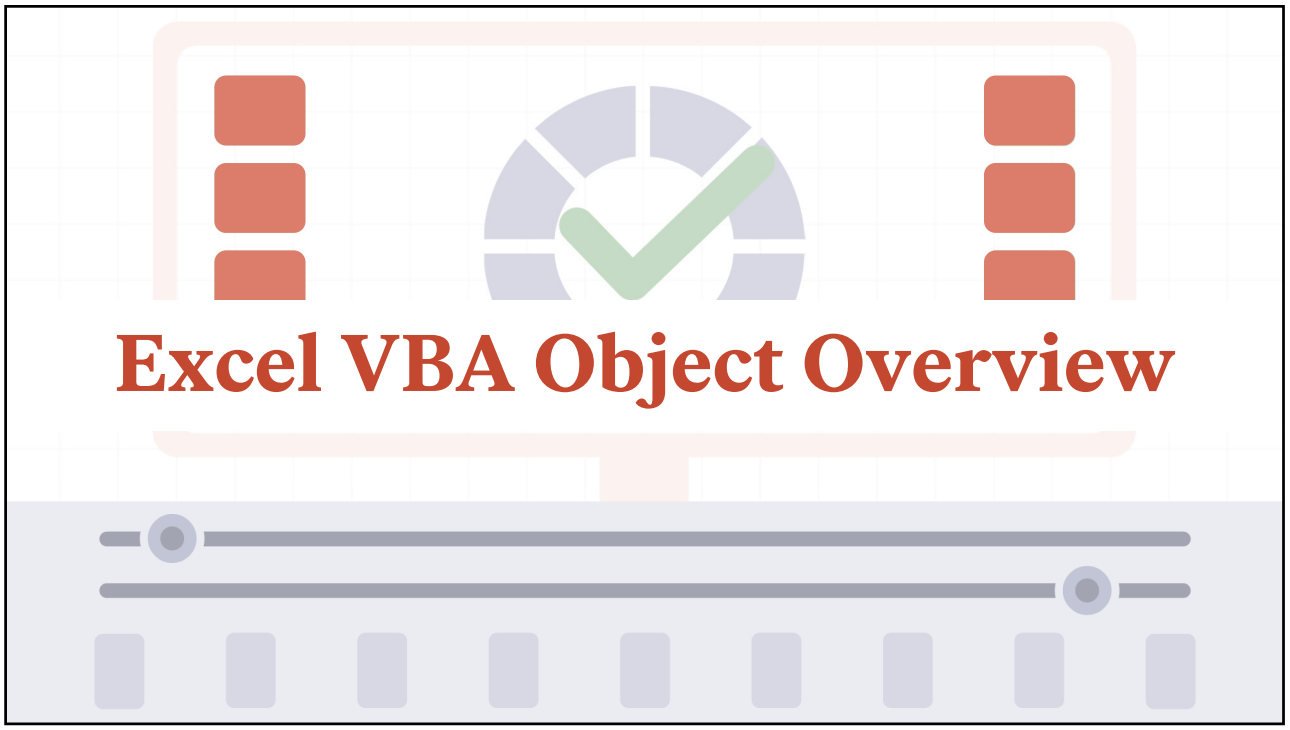
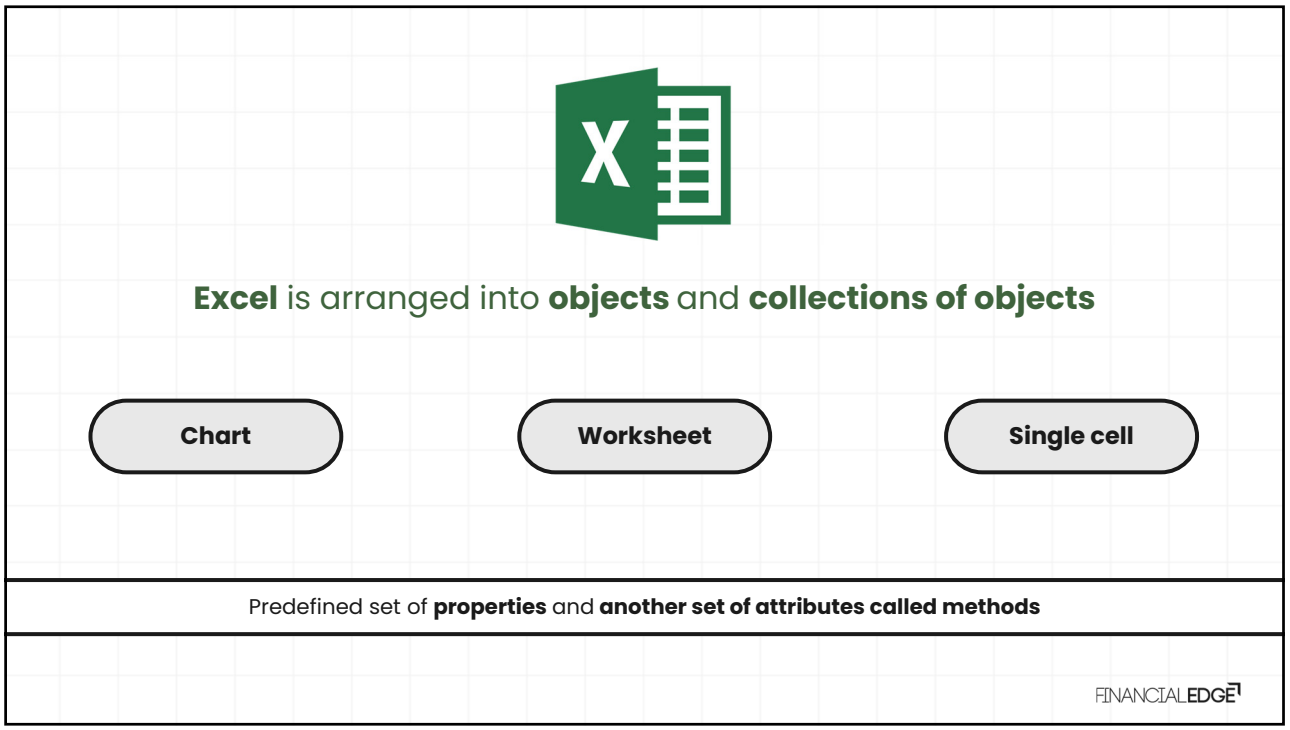
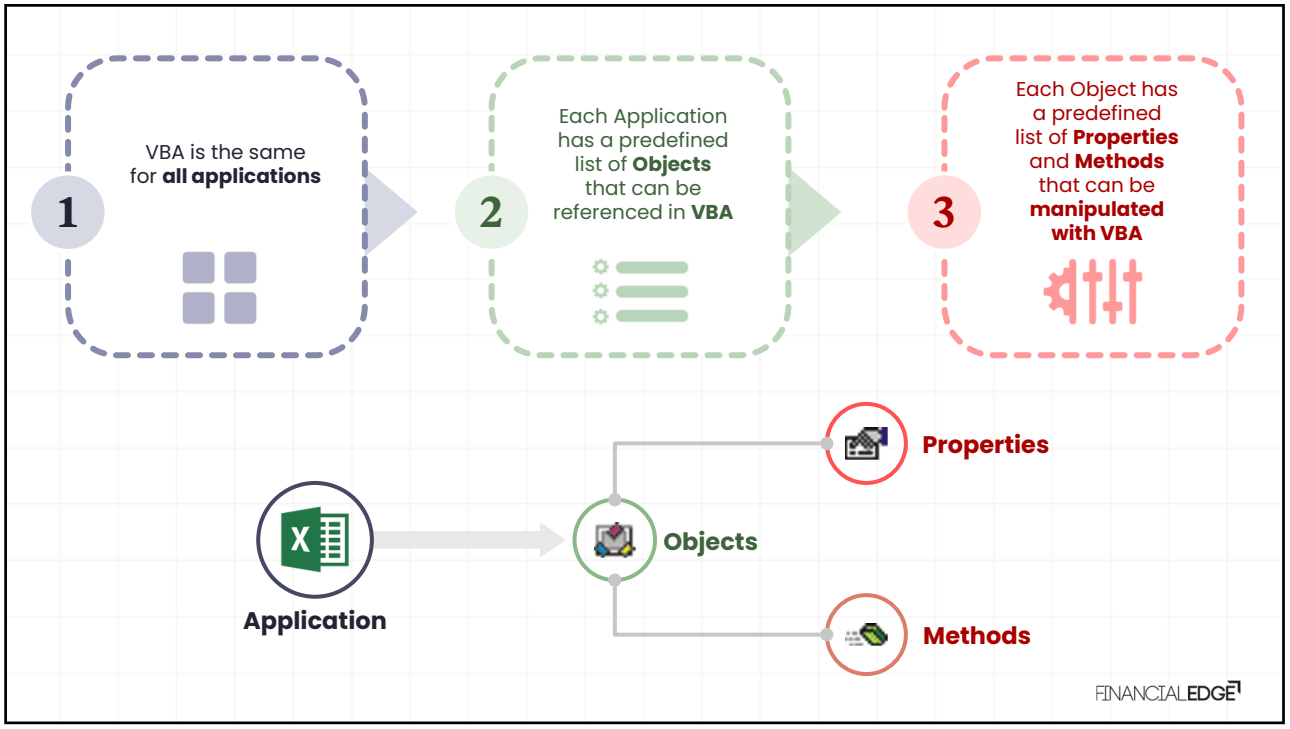




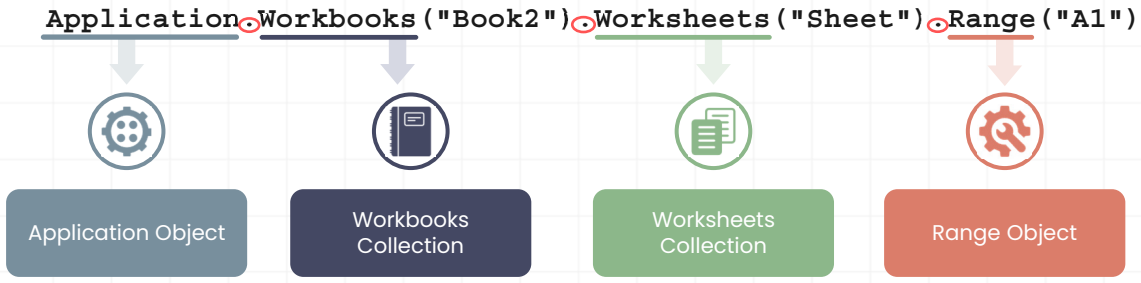
© 2024 Financial Edge Training





© 2024 Financial Edge Training

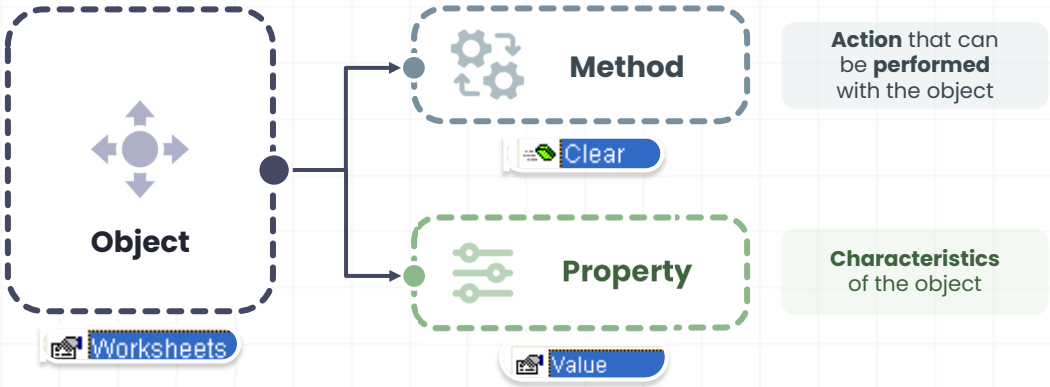
In Excel VBA code, think of referring to an object using its 'address':



If the Worksheet named Sheet is **active** then you could refer to the **range above** in your code as

```
Range("A1")
```

FINANCIALEDGE



In the Visual Basic Editor:

Tools > Options > Editor tab > Auto List Members

Tools > Options > Editor tab > Auto Quick Info

FINANCIALEDGE

© 2024 Financial Edge Training

Objects, Properties and Methods

Properties are the characteristics of an Object which can be **measured** and **quantified**

Generically this is **written as:**

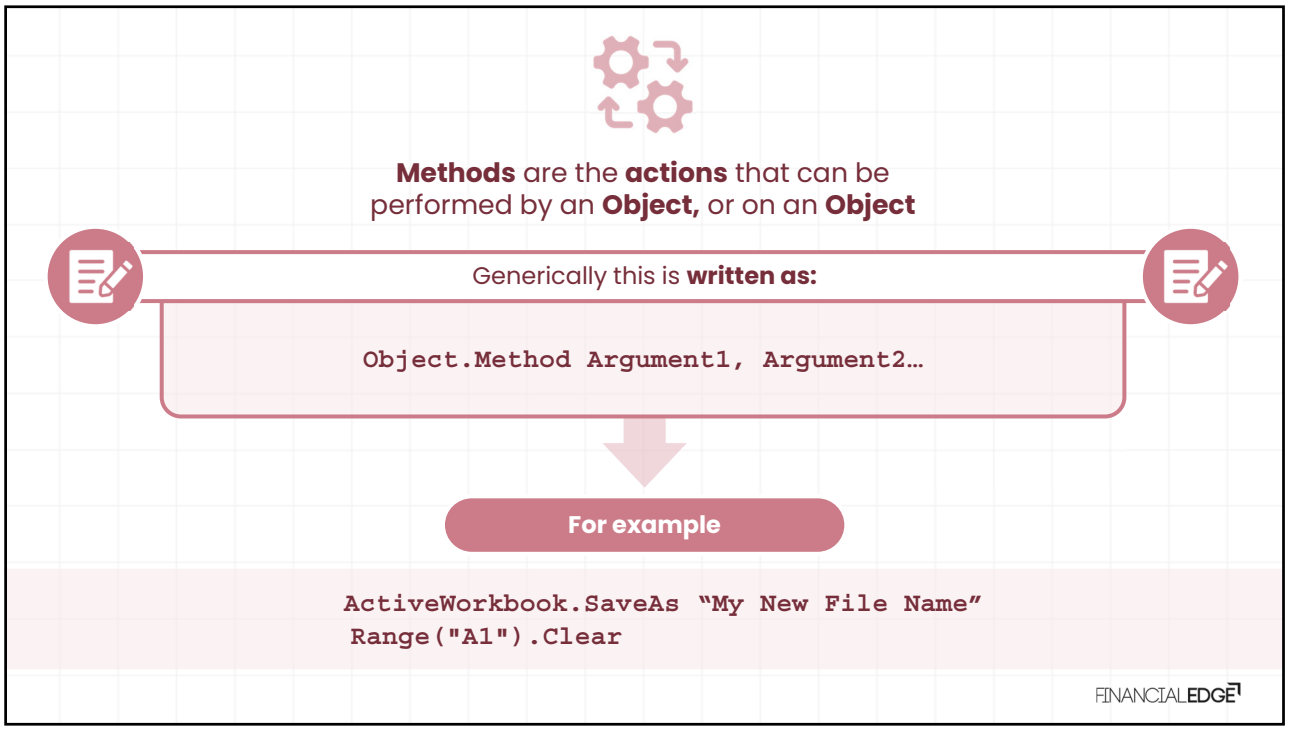
```
Object.Property = Value
```

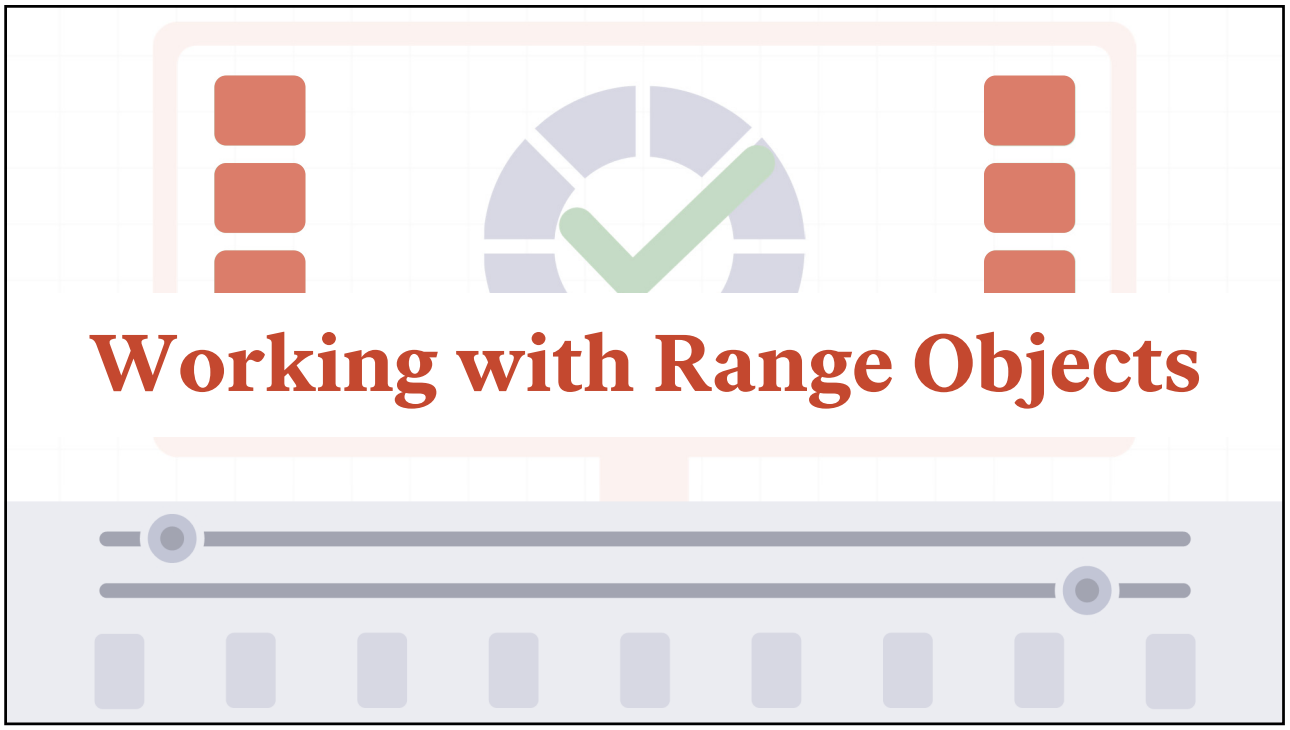
For example

```
ActiveSheet.Name = "My New Sheet Name"
Range("A1").Value = 10
```

FINANCIALEDGE

© 2024 Financial Edge Training





© 2024 Financial Edge Training



A **cell range in Excel** can be assigned a **value**

FINANCIALEDGE

ActiveCell (only ever one cell)

```
ActiveCell.Value = "My Cell Value"
```

Selection (not always a range)

```
Selection.Value = 10
```

Range (includes named ranges)

```
Range("A1").Value = 20
Range("MyRangeName").Value = "My New Value"
Range("A1").value = Range("A2").Value
```

Cells

```
Cells(1, 1).Value = "My New Value"
```

FINANCIALEDGE

© 2024 Financial Edge Training

Object.property.ColorIndex =

One of Excel VBAs **56 color index codes**

Color Index 3	**Red**
Color Index 1	**Black**
Color Index 5	**Blue**

FINANCIALEDGE

Object.Property.Color =

Colors including:

vbBlack	**vbBlue**	**vbGreen**
vbRed	**vbWhite**	**vbYellow**

FINANCIALEDGE

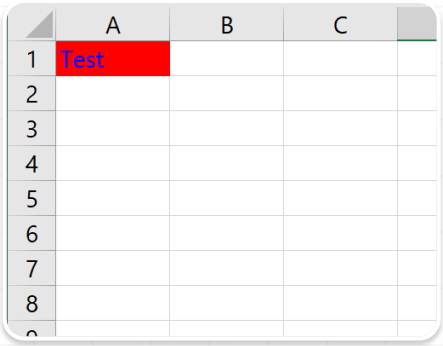
© 2024 Financial Edge Training

Object.Property.Color =

Object.Property.Color =

= RGB(0, 255, 0)

= vbGreen




```
Range("A1").Font.Color = vbBlue
```

The font object has a color property

The value of vbBlue

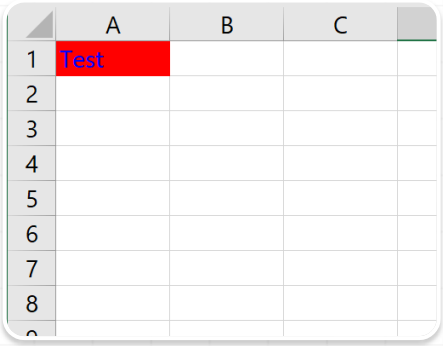
FINANCIALEDGE

Object.Property.Color =

Object.Property.Color =

= RGB(0, 255, 0)

= vbGreen




```
Range("A1").Font.Color = vbBlue
```

```
Range("A1").Interior.ColorIndex = 3
```

FINANCIALEDGE

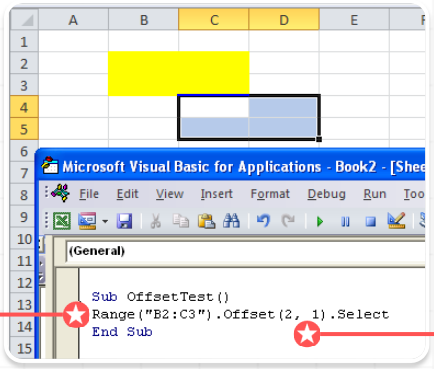
© 2024 Financial Edge Training

Range Object Properties and Methods

Object.Offset (rowOffset, columnOffset)

Only applies to a **range object**

Useful in **looping procedures**



Range object

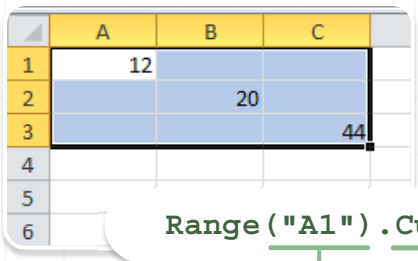
Move 2 rows down, 1 column across

FINANCIALEDGE

© 2024 Financial Edge Training

Range( ).CurrentRegion

The current region is a **range bounded** by any combination of **blank rows** and **blank columns**




```
Range("A1").CurrentRegion.Select
```

Reference to the **range object**

Further properties can be added to **CurrentRegion**

FINANCIALEDGE

Worksheets().UsedRange

The UsedRange returns the range of used **cells on a sheet**

The range is bound by the **rectangle** with the most **extreme cells** at **its corners**

UsedRange is a property of the **Worksheet object** and returns a **range object**

```
ActiveSheet.UsedRange
```

Reference to the worksheet. **ActiveSheet** is a useful **application property**

Further properties can be added to **UsedRange**

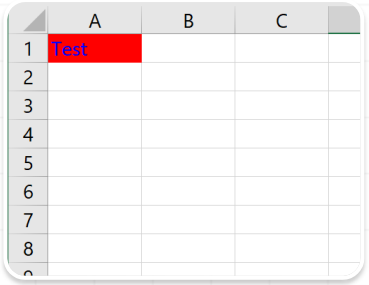
FINANCIALEDGE

© 2024 Financial Edge Training



The selected method will perform an **action** with the **selected object**

```
Range("A1").Copy Range("A3")
```



Range A3 or **cell A3** in **the spreadsheet**

Will be **copied to cell A3**

FINANCIALEDGE

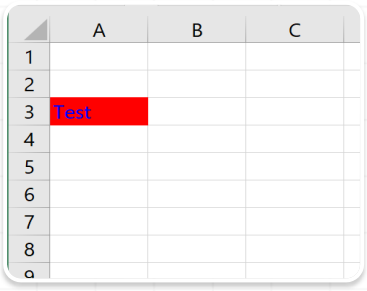


The selected method will perform an **action** with the **selected object**

```
Range("A1").Copy Range("A3")
Range("A1").Clear
```

The clear method **specified**

The contents of **cell A1 will be cleared**



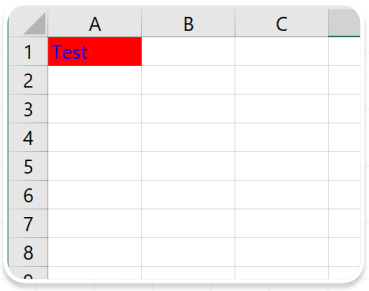
FINANCIALEDGE

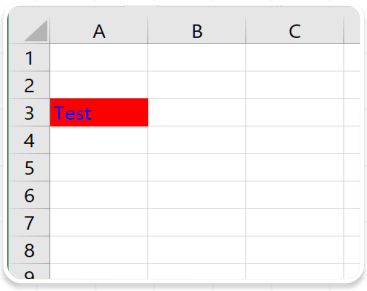
© 2024 Financial Edge Training



The selected method will perform
an **action** with the **selected object**

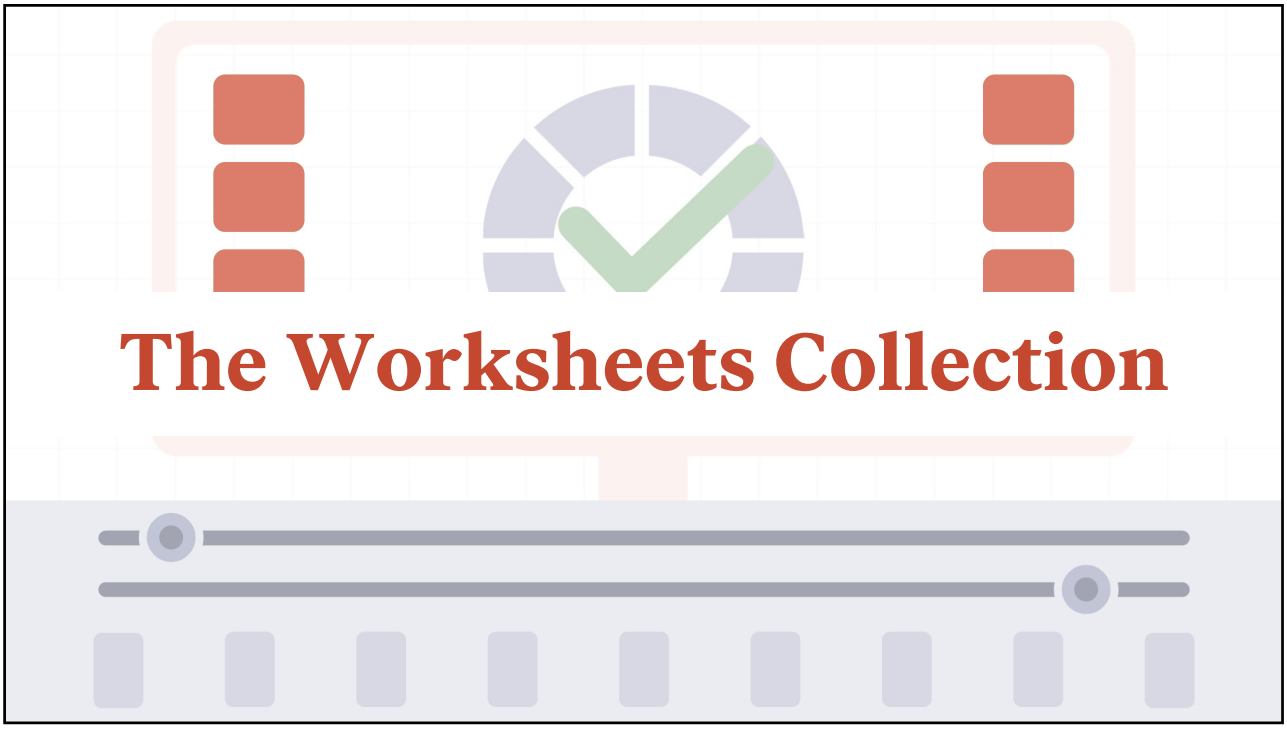
```
Range("A1").Copy Range("A3")
Range("A1").Clear
```



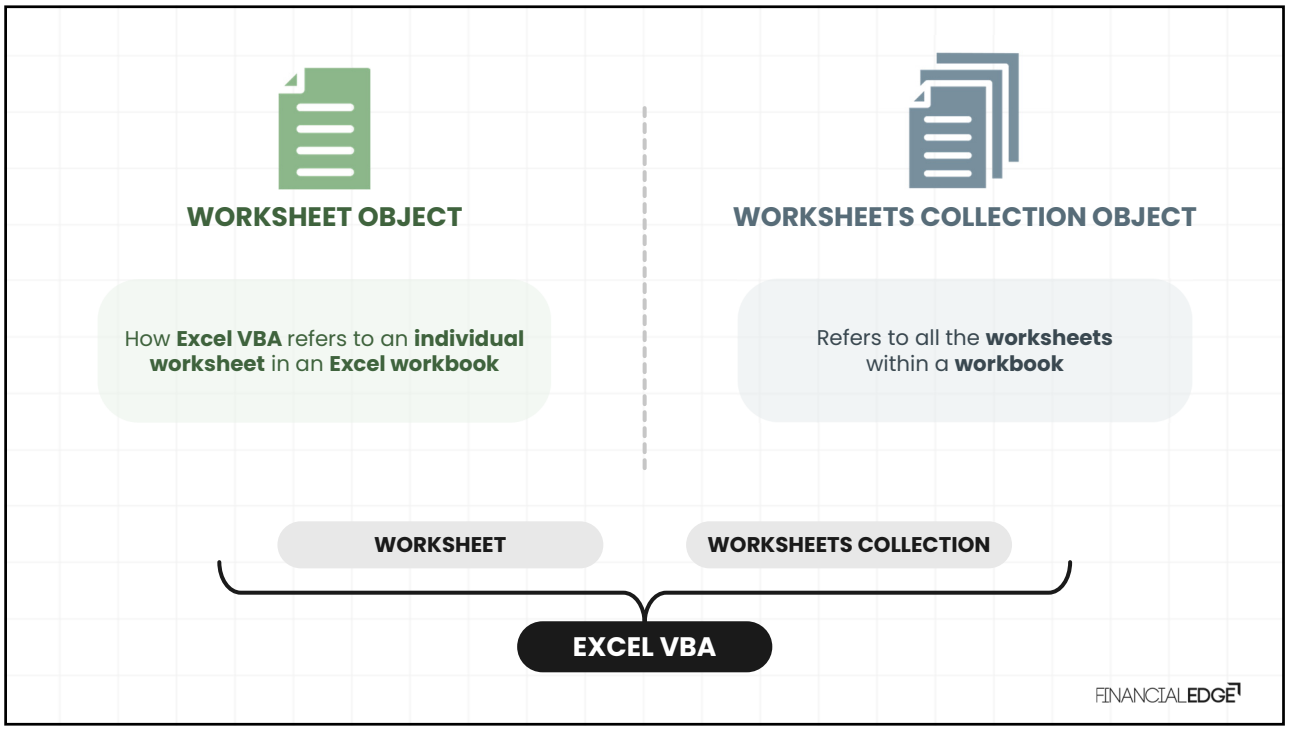


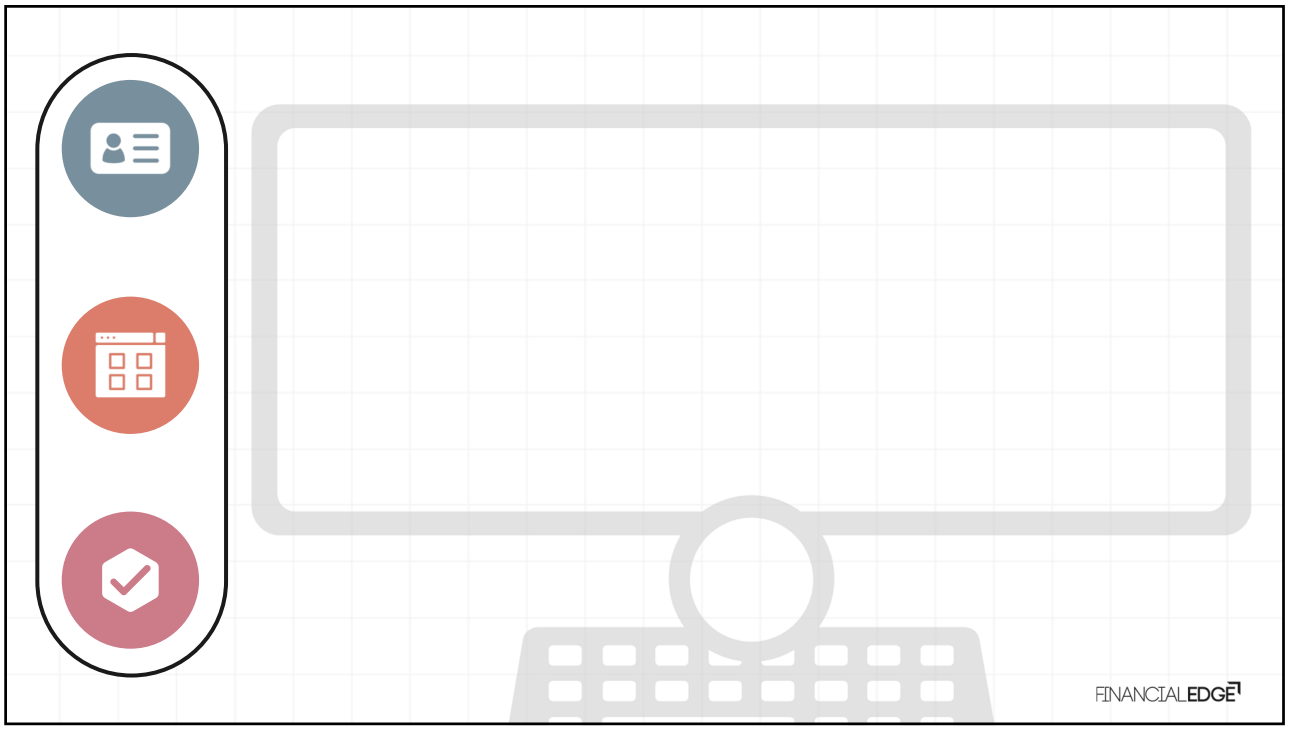
When we run Excel VBA code, **each line of code is run sequentially**

FINANCIALEDGE

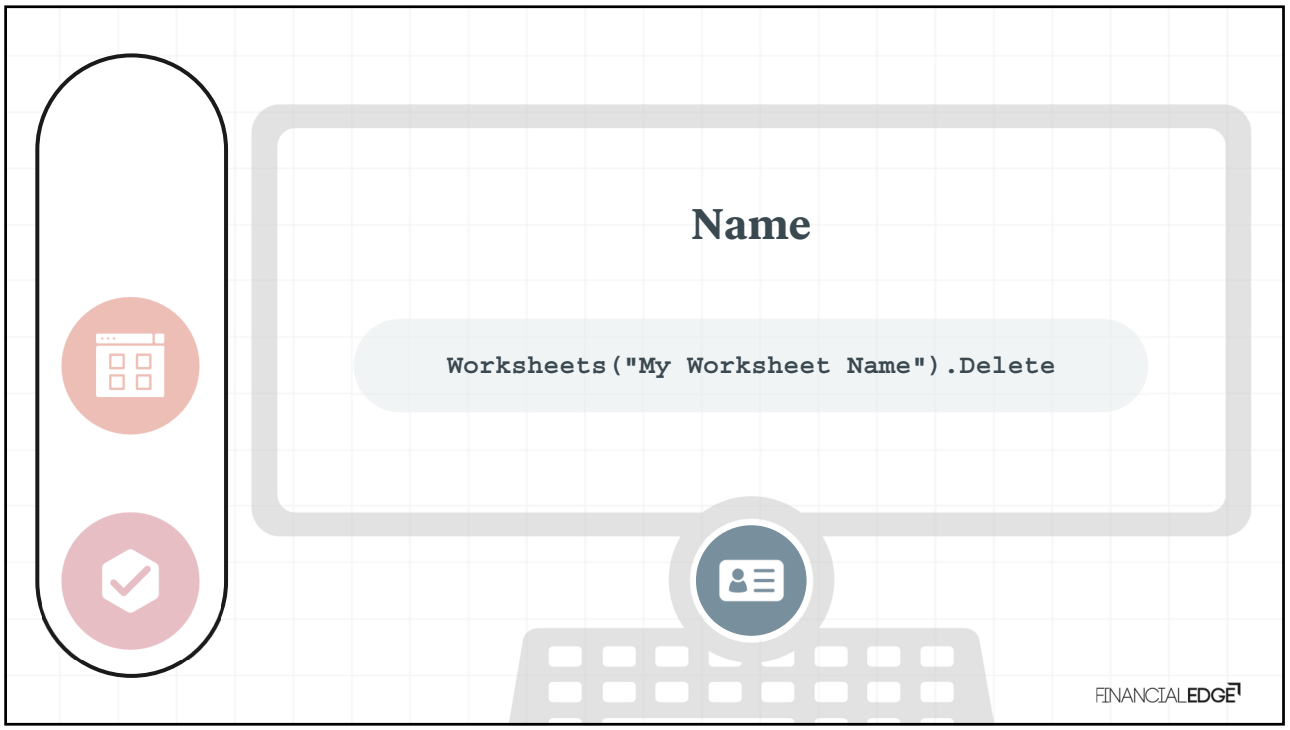


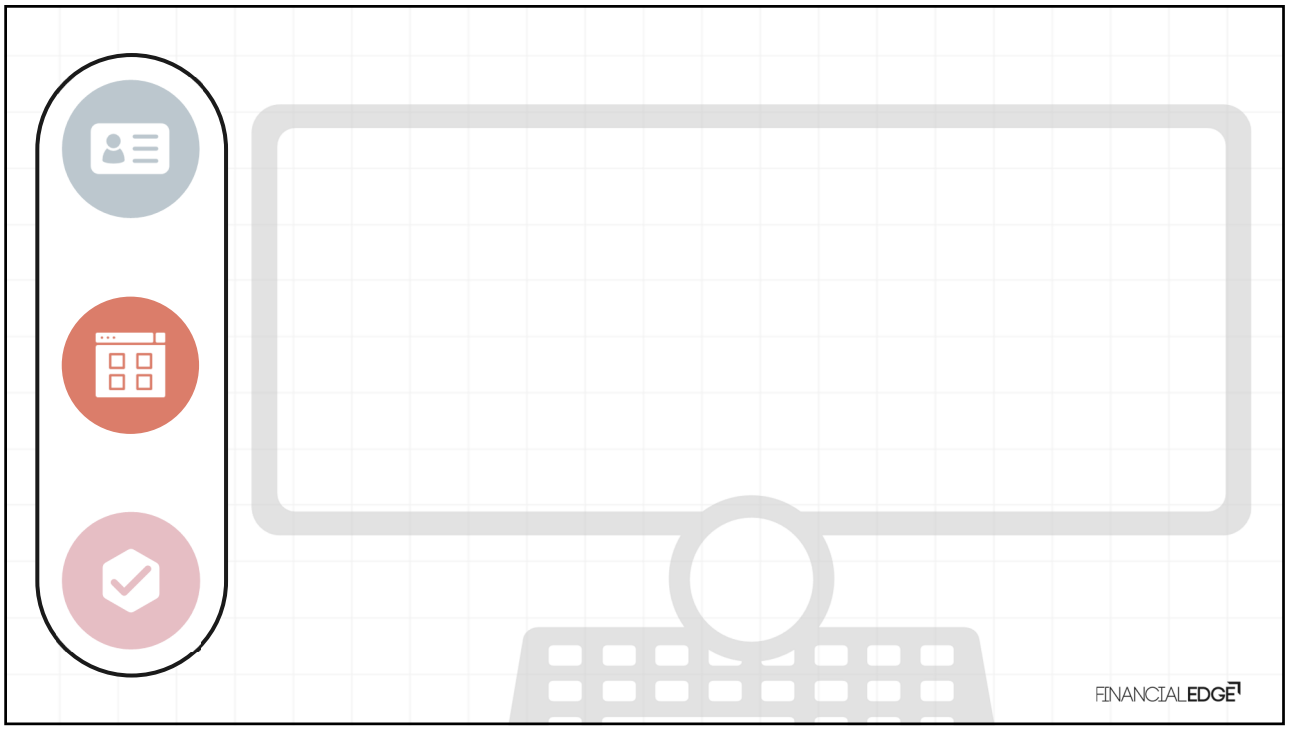
The Worksheets Collection



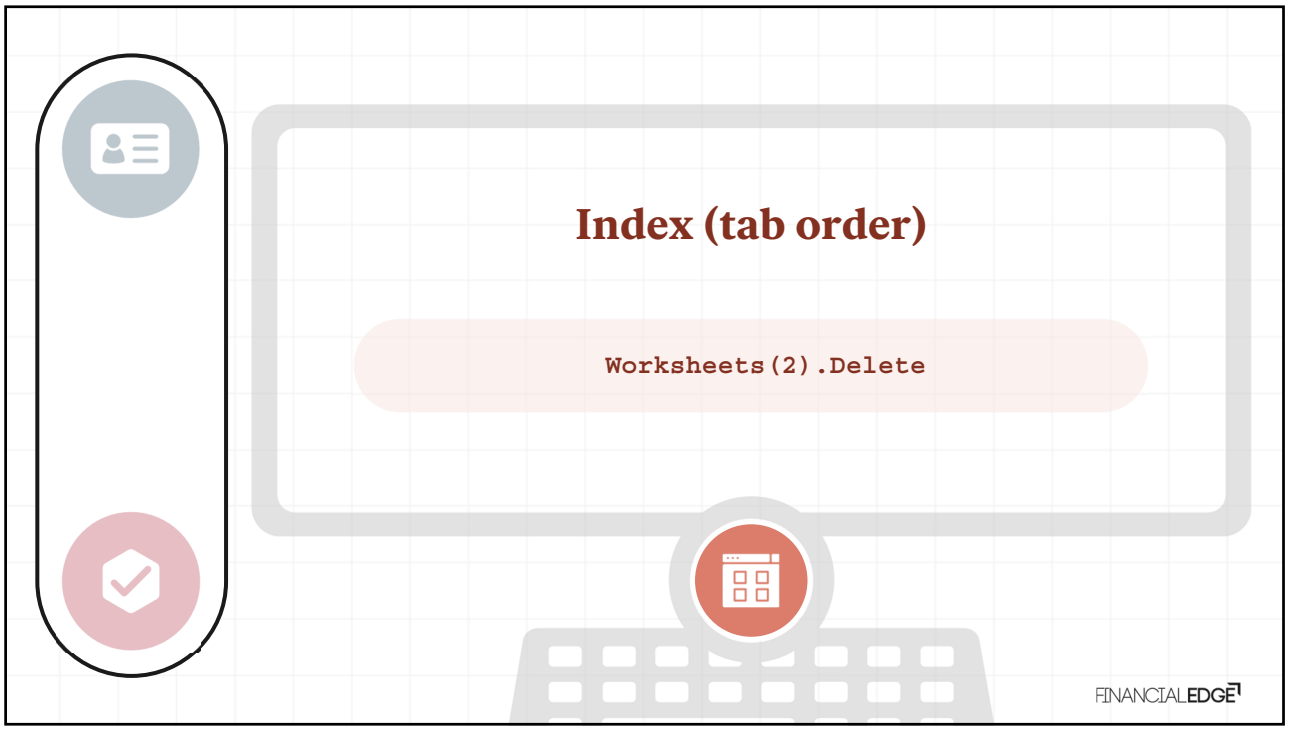


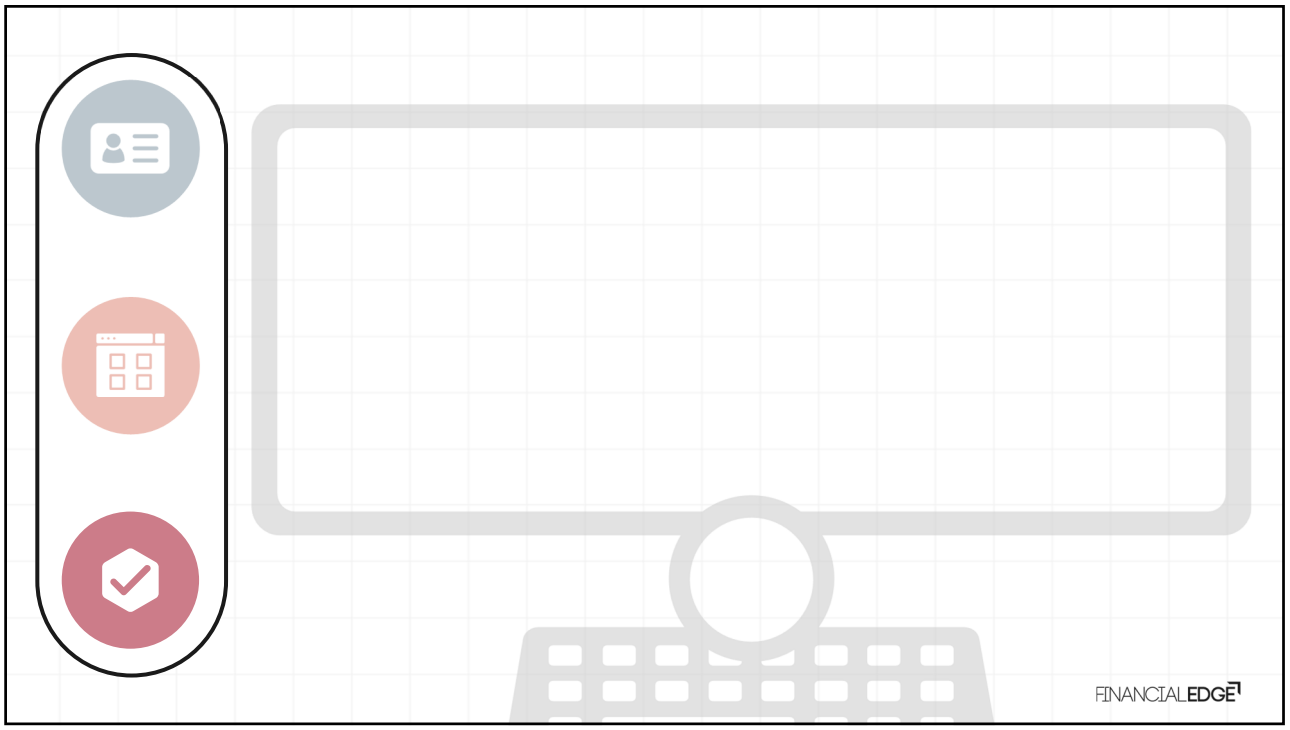
© 2024 Financial Edge Training



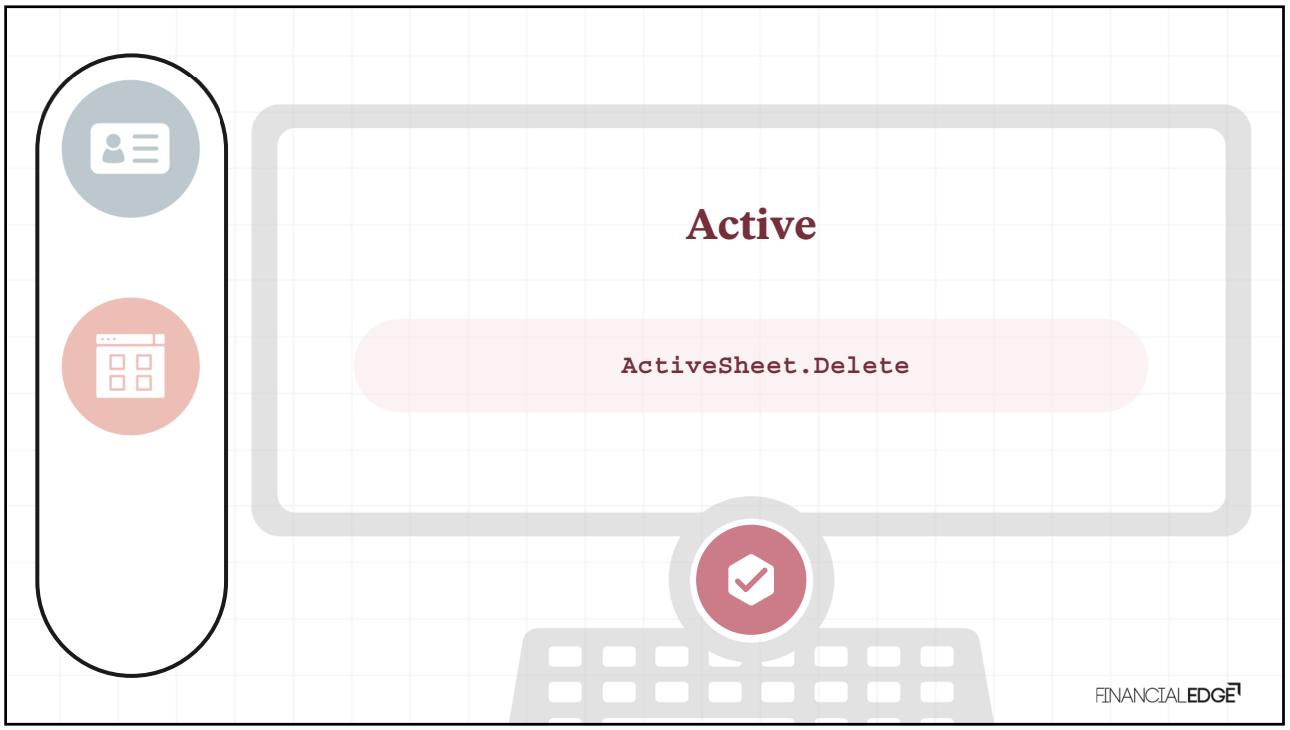


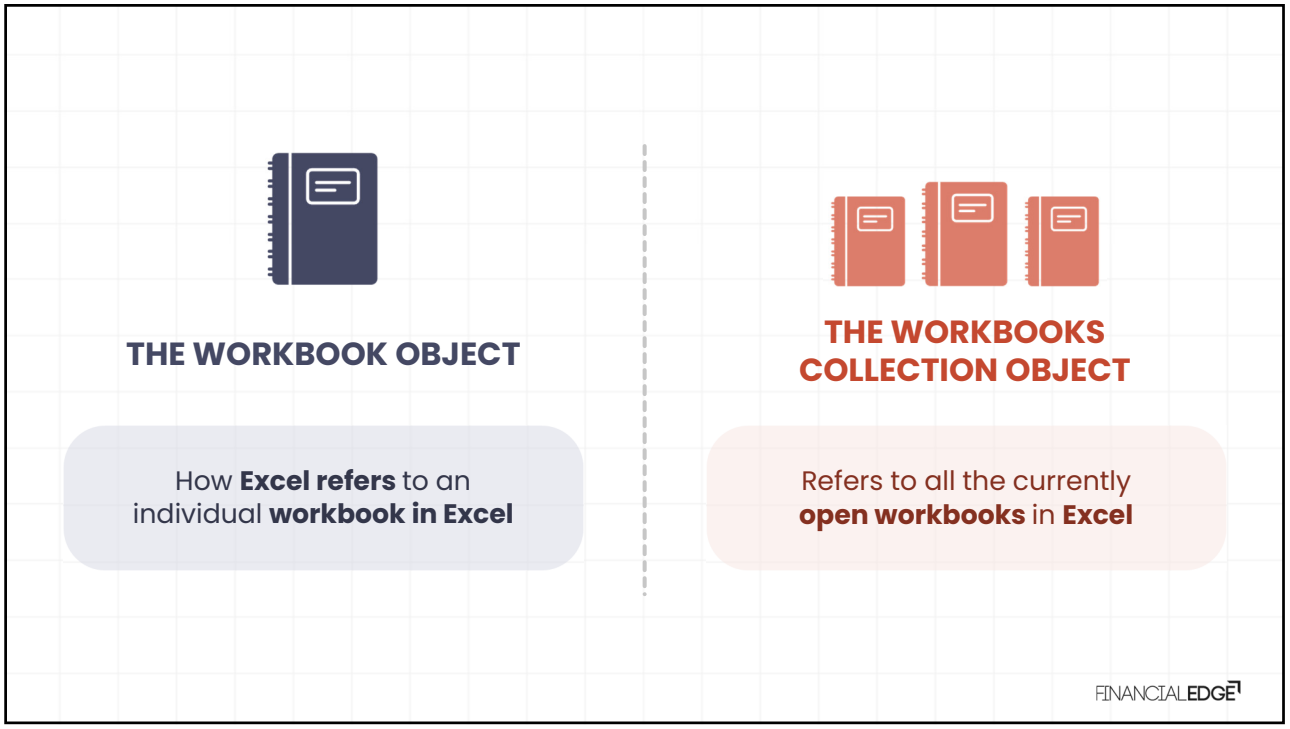
© 2024 Financial Edge Training





© 2024 Financial Edge Training

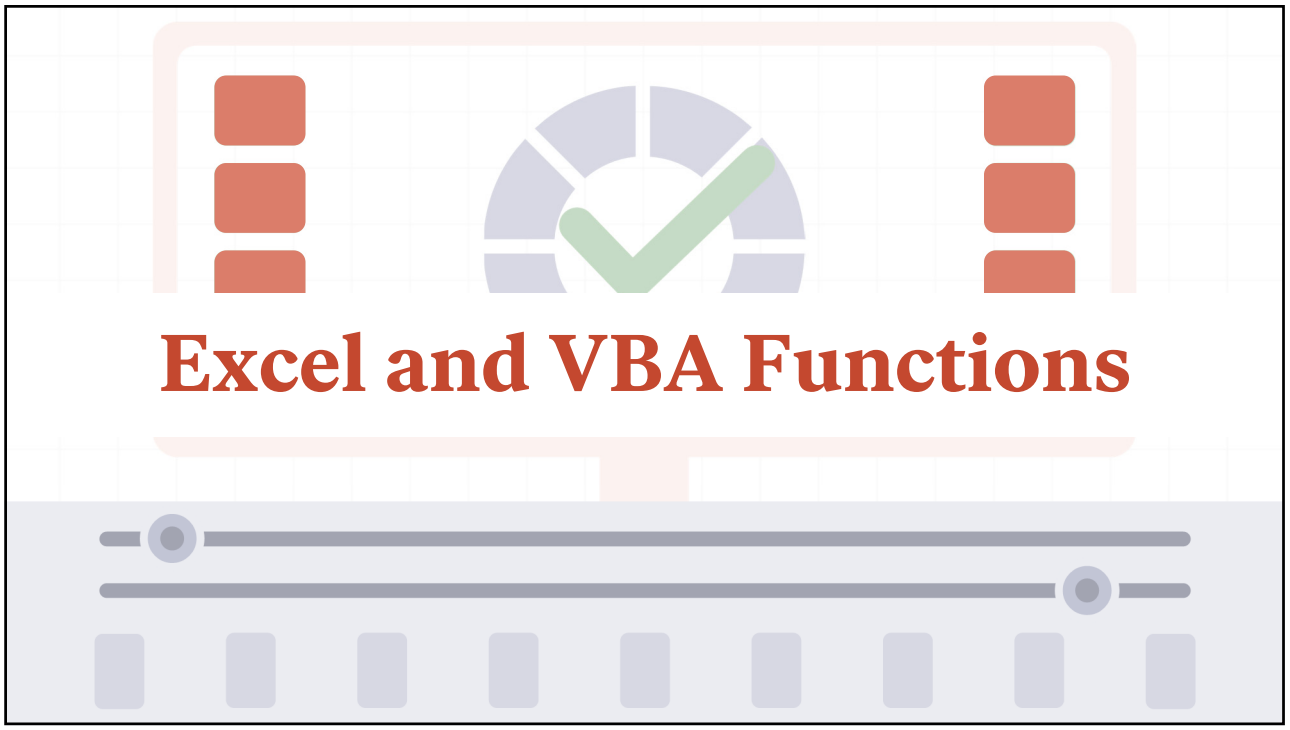




© 2024 Financial Edge Training

Name	`Workbooks("My Workbook.xlsx").Close`
Index (order of opening)	`Workbooks(2).Close`
Active	`ActiveWorkbook.Close`
This	`ThisWorkbook.Close`

FINANCIALEDGE



© 2024 Financial Edge Training



VBA has a **variety of built-in functions** that simplify **calculations** and **operations**



Many **VBA functions** are similar (or identical) to **Excel Worksheet functions**

```
Sub CalcRoot ()

Range ("A1").Value = Sqr(36)

End Sub
```

The **VBA** Sqr function is equivalent to the **Excel SQRT** worksheet function

Depending on the function, arguments can be **variables, constants or expressions**

FINANCIALEDGE



If a VBA function that's equivalent to one you use in Excel **isn't available,** you can use **Excel's worksheet functions** directly in **your code**



```
'Note cells used in these functions must be identified with the Range object and the cell reference placed inside " "

'The WorksheetFunction.Sum is used to replicate the Excel Sum function
Range("J18").Value = Application.WorksheetFunction.Sum(Range("J8:J17"))

'.Vlookup works similarly
Range("C20").Value = Application.WorksheetFunction.VLookup(Range("C24"), Range("B27:C29"), 2, 0)
```

Quick Info provides **guidance** on the **arguments required**

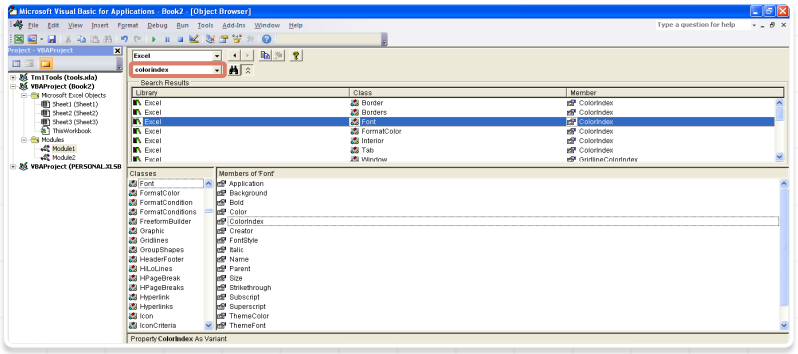
FINANCIALEDGE

© 2024 Financial Edge Training

The Object Browser

Excel VBA (Visual Basic Editor) > View > Object Browser

The Object Browser lists every **property** and **method** for **every object**



The Object Browser provides a really useful **overview** of all of **the properties** and **methods** relating to an **object**

F2 **Object Browser F2**

F1 Type the **object, property or method** in the **Code window** and **press F1**

FINANCIALEDGE

© 2024 Financial Edge Training



© 2024 Financial Edge Training