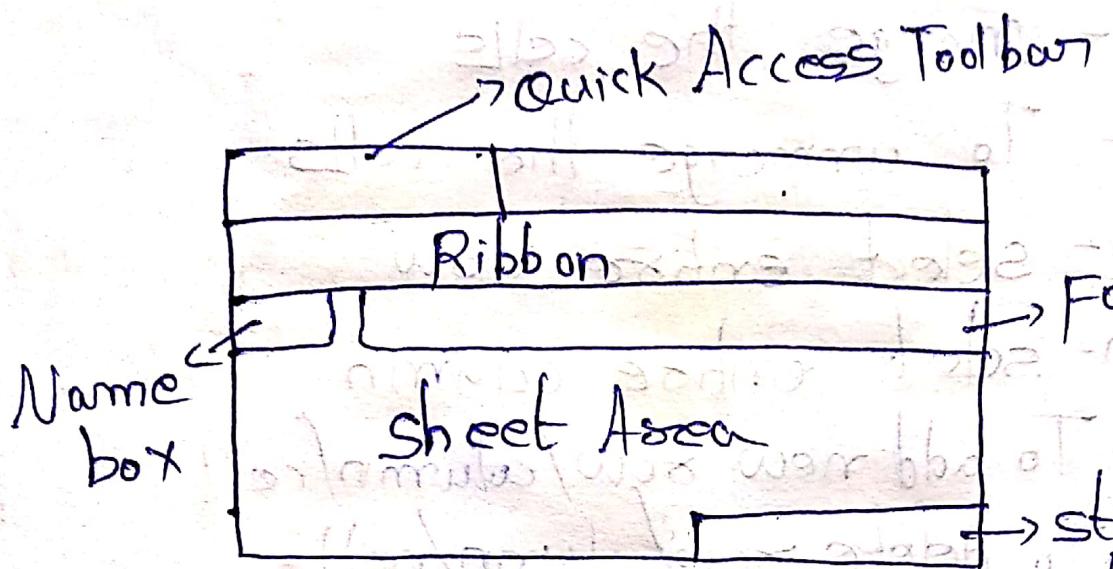


Microsoft Excel :-



shortcuts in Excel :-

- $ctrl + N$ - To add the new workbook
- $ctrl + W$ - To close the work book
- $ctrl + F1$ - To hide/unhide the ribbon
- $ctrl + shift + f4$ - To expand/collapse Formula bar
- shift + F11 - To add the worksheet.
- Alt, h, d, s - To [delete] the worksheet.
- Alt, e, s - paste special.
- = $ctrl + B$ - Bolding
- = $ctrl + I$ - Italic
- = $ctrl + u$ - underline

- **ctrl+1** - format the cells
- **Alt,h,m,c** - merge the cells
- **Alt, h,m,u** - To unmerge the cells
- **shift+spacebar** - select entire row
- **ctrl+spacebar** - select entire column
- **ctrl+t,f** - To add new row/column/cell
- **ctrl+del** - To delete row/column/cell
- **ctrl+F** - To find number/data
- **ctrl+h** - To replace
- **ctrl+g** - Go to (And Go to special (For formulae etc))
- **ctrl+k** - hyperlink
- **shift+↓** → select all the column data without empty cells.
- **ctrl+z** → undo [100 by default]
- **ctrl+A** → All selection

Quickly get the sum of highlighted cells

→ **Alt+=** : Total sum

→ **counta** : For non-numeric count without blank

→ count : For numeric count

=ctrl+Y : redo

shortcuts :-

ctrl+backspace = Active Cells selection

ctrl+.

- Moves to 4 corners of sheet

ctrl+bab

- current open workbooks

ctrl+s

- saves active workbook

ctrl+f2

- Save as

(ctrl, shift, alt) → pivot table

ctrl+shift+L → For filter

ctrl+f3 → Named Management

alt, A, E → Tent for columns

alt, A, tm → Remove duplicates

alt + f5 → To Refresh Pivot Table

ctrl + R → To drag the formula to right

alt + ' → Order sum

Excel shortcuts :-

ctrl + pg up - moving left

ctrl + pg dn - moving right

ctrl + z - undo

ctrl + alt + v - paste special

ctrl + ; - current date

ctrl + shift + : - current time

F2 - edit the cell

ctrl + Home - first cell reaching (Go Back)

ctrl + end - go back to last cell

shift + spacebar - select entire row

ctrl + spacebar - select entire column

ctrl + shift + ' - insert

ctrl + ' - delete

Alt + '=' - sum function

ctrl + D - Drag the formulae for below

ctrl + enter - formula is copied to all cells selected.

ctrl + o - hide columns

ctrl + g - hide rows

ctrl + shift + o - unhide columns

ctrl + shift + g - unhide rows

ctrl + t - create a new table

F4 - Repeat action

Alt + enter - new line inside a cell

ctrl + ; - gives current date

ctrl + : - gives current time

=F2 → over ~~write~~

=shift + F2 = Comments \Rightarrow To add delete comment

~~=~~ shift + F2 + Esc + Delete \Rightarrow To delete comment

Excel cal always starts from 01-01-1900,

02-01-1900 → 02

03-01-1900 → 03

31-12-1900 → 365

01-01-1901 → 366

$c81 + j \rightarrow$ current date

$c81 + i \rightarrow$ Time

1/1/2000

9/6/2018

=DATEDIF(G10, I11, "Y") & "years" &

DATEDIF(G10, I11, "YM") & "months" &

DATEDIF(G10, I11, "MD") & "Days"

18 years 8 months 5 days

Formulas → Evaluate Formula

↓
To execute step by step.

Alt + F9 → auto format

~~Alt, F1, B~~

Ctrl, Shift, → ↗ To select
or ↘

Alt + F11 → VBA editor

Comments → ↗ { } 2 ways
RBM ↘

Alt + F8 → Macros

Shift + F2 → Comment

Shift + F7 → UserForm from module code.

both ways ↗ Creating object → properties

→ class or object first

→ class name

→ class name with break at

Excel

Basic Reporting Tool

To Automate we use "VBA"

Visual Basic for Application
(Macros)

Reporting is Nothing but a saving or Data with you.

ETL => Extract Transform Load (Get & Transform Data).

16384 (=columns)

1048576 (=rows)

Data-validation → whole numbers →

to specify cell to validate

on specific range

settings (whole number) | Error Alert

Text length.

To create own msg.

To restrict phone numbers etc. - - -

Gold seek

Ctrl + F1 → Pivot field → Locked

From Pivot sheet to skip particular cells.

Pivot Tables

Analy3 & (Tabs) Design.

Field list

Values (only numeric)

Calcuations

Pivot chart

Enable select → entire sheet →

two ways to issue req - p1

two ways to contain answer

private window into c-

two ways - vba &

offset(A1, 0, 0, countA(A1:A7), countA(1:1))

↓
offset internal formula:

F10 → options → customized Ribbon →

Developer Tab

Ctrl+F7 → Immediate window debug print

locals
watch

|| local variables
|| Add watch

Project Explorer

Properties

VBA → VBA Project → Excel Project
sheet (mywish)

VBA :-

Alt + V + S -> ->

Excel

Alt + - - -

F5 - single Execution

F8 - step by step

F9 - prog stops at Break Point

naming convention of subroutines

→ only Alphabets starting

→ only - allows after 1st char - ,

VBA (Visual Basic for Applications) :-

programming language
from Microsoft.

Excel
Word
PowerPoint
Outlook.

Excel - Reporting
Word - Documental
PPT → presentation
Outlook → Mailing

To Automate these, we
use VBA
=> To avoid Repetitive works.

VBA is a Backend Programming Interface
for Microsoft Office Applications.

→ To open VBA ^{editor} (ALT + F11)

To get Macro window (ALT + F8)

There are 5 windows available in VBA

- 1) Project Explorer (CTRL + R)
- 2) Properties window (F4)
- 3) Immediate window (CTRL + G)
- 4) Locals window
- 5) Watch window

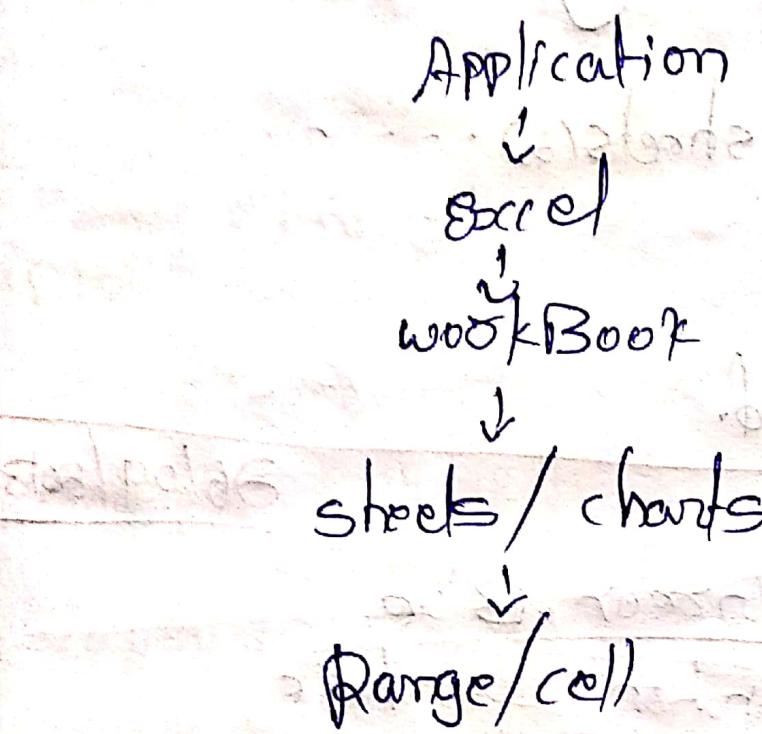
Basically our program starts with

Sub [Name]
ends with
End sub

Naming convention of Subroutine

- ⇒ It should not start with any Special characters or numbers.
- ⇒ It should not contain any special characters. Except underscore (_)
- ⇒ Should not contain more than 256 characters.
- ⇒ It should not have the another subroutine Name (Ambiguity errors)

Excel Object hierarchy model :-



These all are
Objects

Property - Name
= color
height
width

methods

Are Action

- copying
- Deleting
- Adding

Range("A1:A10") \leftrightarrow object

{> naming is property

{> copying is method.

Debugging ways :-

F8 = step by step Execution

F9 = Break point

F5 = Direct Execution

Rem } single line commenting *

Indexing :- sheets(1), sheets(2) ...

~~XLSX~~ → Excel
~~XLM~~ → Macro enabled.
XLSM

26/09/2018

Immediate window, Red, Green, Blue

?RGB(0,0,0)
0

combination of these
3 colors.

?RGB(255,255,255)

16777215

sub proc()

Range("C2").Interior.Color = 16777215

end sub.

DIM is a VBA keyword used to declare the
variable.

Dim book1 = pt

Module1 book1 = pt

1) byte }
2) int/long } number
3) long

4) single } Decimal
5) double }

6) string → characters

7) Boolean → true/false

8) currency → \$, €, £,

9) Date → #14/02/2018#

Variant \Rightarrow 16 Bytes (Default)

Sub spst()

a = 10

b = 20

c = a + b

End Sub

16 Bytes

Sub > psc()

Dim a as Byte

Dim b as Byte

Dim c as Byte

a = 10

b = 20

c = a + b

End Sub

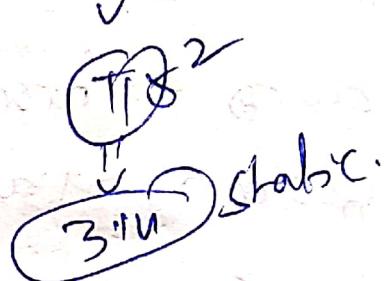
3 Bytes

scope of variable

Dim :- can be used inside sub proc

private :- can't be used inside sub proc

Dim (within module)
public (All modules)
private (within module)
static (constant value)



⇒ Variable :-

Variable is nothing but a Name in a memory

fixed location ~~in~~ a value

⇒ Option Explicit :- which is used to declare
the variables Mandatory.

⇒ By using Option Explicit, we can reduce
the memory consumption and can reduce
increase the program efficiency & also
can minimise program run time.

Datatypes available in VBA

Byte

Integer

Long

Single

Double

String

Boolean

Currency

Date

→ Variant (default)

Scope of Variables :-

If we declare a variable as

public = will have Access to All the modules.

private = will have Access to that particular module only.

static =

Cut :-

sub cut()

sheet1.Range("A1:A3").Cut sheet1.Range("D1")

End sub

Object Variables :-

Range
workbook
worksheet

} objects .

Dim ws as worksheet
Set ws = Sheets("sheet1")
Dim sng as Range
Set sng = Range("A1:A10")

To Avoid Object Scoping we use with...end with

Conditions :-

1) IF - EndIF

2) Select Case - End Select

Loops

1) For - Next

2) For Each next

3) DoWhile Loop

4) Do Until Loop

If condition① Then

stmt②

ElseIf condition③ Then

stmt②

ElseIf condition④ Then

stmt③

else

End If Stmt④

IFF

Range("B1").Value = IFF(a > 90, "Excellent",

IFF(a < 90 and a >= 75, "Super",)

End IFF (or < 75 and a >= 50, "Average", "Worst"))

Select-Case

Sub P50-1()

Dim a As Integer

a = Range("A1").Value

Select Case a

Case ~~for~~ To 90 To 100

Range("B1").Value = "Excellent"

case ~~89~~ To 75 To 89

Range("B1").Value = "Super"

Case ~~74~~ To 50 To 74

Range("B1").Value = "Average"

case Else : Range("B1").Value = "Worst"

End Select

End Sub.

Do While : - Executes until other condition is True

Do Until : - Executes until its condition is False

Errors handling :-

- 1) On Error Resume Next, ("A") = 0
- 2) On Error GoTo "Label"
- 3) On Error GoTo 0

immediate window
?Range("A1") • currentRegion.select
True. linked Region.

// connects through links, diagonally.

? Activatesheet. UsedRange, Select

// selects All.

CurrentRegion :- CurrentRegion is a linked cells Range where we need to have a connection between cell to cell.
→ CurrentRegion is the Range Property.

(60) UsedRange - UsedRange is the Range from the cell used in a sheet to last cell used in a Range.
→ UsedRange is the sheet property.

`chslf \` \Rightarrow last used row

`chslf \` \Rightarrow last used column

FSO's :- File system objects :-

To Access, modify, Delete and create we use file system object.

\Rightarrow Microsoft Scripting Runtime Library

\Rightarrow Create New Instance.

Array :- It is a collection of elements of same kind of data types
(or)

Group of variables of same data type.

`Dim a(5) as Integer` `Dim b(3) as Variant`

`a(0)=10`

`b(0)=16`

`a(1)=20`

`b(1)="SSH"`

`a(2)=30`

`b(2)=#11/05/2018#`

`a(3)=40` `a(5)=60`

`b(3)= 10,000`

`a(4)=50`

Option Base 0 - is used to set the lower bound of an array to 0 (or 1)

option Base : 0 or 1

Dim myarr(5) as Integer
 ↓
 Index

Dim myarr(0 to 5) as Integer
 ↑
 Lower Bound Upper Bound
 "lbound" "ubound"

option Base 1

sub Array-1()

Dim myarr(3) As Integer

myarr(0) = 10

myarr(1) = 20

myarr(2) = 30

myarr(3) = 40

end sub.

Sub A\$say_2()

Dim myarr(3) As Variant

myarr(0)=10

myarr(1)="SSH"

myarr(2)=#5/4/2018#

myarr(3)=100000

End Sub

Sub A\$ay_3()

Dim myarr() To 3) As Integer

Dim myarr() As Integer

18=Range("A11").End(xlDown).Row

~~Re~~ Redim myarr(18)

For i=1 To 18

myarr(i)=Range("A11 & " & i & "").Value

Next i

End Sub

Multi Dimension Assays :-

⇒ Redim is used to increase the size of an Assay.

⇒ When we use Redim, the previously stored Elements will go off.

⇒ Redim preserve will also used to increase the size of an Assay, but when we use Redim preserve, the previous elements will reside in an Assay and new elements will append in the Assay.

⇒ Diff b/w sheet change & sheet selection change?
sheet change :- If we change the cell content, the event will trigger.

sheet selection change :- If we change the cursor position, the event will trigger.

~~Alt + F + M~~ => To create Module

User Forms

Tab order

FSO -

ADOS -

outlook -

Powerpoint -

word -

~~Object~~ Application

~~Excel~~

Excel

workbook

worksheets / charts

Range / cell

Word Application

↓
Word

↓
Document

↓
Page

↓
Shape

Powerpoint

Application

↓
Powerpoint

↓
Presentation

↓
Slide

Early Binding :- is a Technique to write a program, by taking the reference of object library.

Ex:- Microsoft scripting Runtime for FSO's.

Late Binding:- is a Technique to write a program, by creating the object on our own.

Ex:- Dim FSO As Object
Set FSO = CreateObject("Scripting.FileSystemObject")

Note:- It is always a Good practice to write a Program in Early Binding and Convert it to Late Binding so that we will not have compatibility issues.

System defined functions - User Defined functions

→ Insts
→ ~~InstsOver~~
→ ~~Stg diverse~~
→ Strcomp
→ Date
→ Datediff
→ Envioron
→ Lcase, Ucase
→ split
→ join

→ Area of circle
→ Perimeter of circle
→ vlookup
→ Repetitive fash

vbBinary ---

Strcomp ("swaga", "sw", vbTextCompare)

1 → substring

-1 → different

0 → ~~off~~ Same