VBA में ऑपरेटर और ऑपरेटर प्राथमिकता (Operators in VBA and operator precedence)

उद्देश्य : इस पाठ के अन्त में आप निम्नलिखित कार्य करने योग्य होंगे :

- **VBA में विभिन्न ऑपरेटरों और उनकी प्राथमिकता की व्याख्या करना।**

VBA में ऑपरेटर (Operators in VBA)

एक ऑपरेटर को सरल अभिव्यक्ति का उपयोग करके परिभाषित किया जा सकता है - 4 + 5, 9 के बराबर है। यहाँ, 4 और 5 को ऑपरेंड कहा जाता है और + को ऑपरेटर कहा जाता है। VBA निम्नलिखित प्रकार के ऑपरेटरों का समर्थन करता है :

- अरिथमेटिक ऑपरेटर (Arithmetic Operators)
- तुलनात्मक ऑपरेटर (Comparison Operators)
- लॉजिकल (या रिलेशनल) ऑपरेटर (Logical (or Relational) Operators)
- कांसटेनेशन ऑपरेटर (Concatenation Operators)

अंकगणितीय ऑपरेटर (The Arithmetic Operators)

निम्नलिखित, अंकगणितीय ऑपरेटरों को VBA द्वारा समर्थित किया जाता है।

मान लें वेरिएबल A5 धारणा करता है और वेरिएबल B10, धारणा करता है, टेबल 1 में दिखाए गए विभिन्न ऑपरेटरों को परिणाम।

टेबल 1

ऑपरेटर	विवरण	उदाहरण
+	दो ऑपरेड जोड़ता है	A + B 15 देंगे
-	पहले से दूसरे ऑपरेड को घटाता है	A - B -5 देंगे
*	दोनों ऑपरेटरों को गुणा करता है	A * B 50 देंगे
/	denominator द्वारा संख्यात्मक विभाजित करता है	B / A 2 देंगे
%	एक पूर्णांक विभाजन के बाद शेष देता है।	B % A 0 देंगे
^	एक्सपोनेंटिएशन ऑपरेटर	B ^ A 100000 देंगे

तुलनात्मक ऑपरेटर (The Comparison Operators)

निम्नलिखित तुलनात्मक ऑपरेटर परिणाम (VBA) के साथ दिखाए गए है।

मान लें वेरिएबल A10 धारण करता है और वेरिएबल B20, धारण करता है, तालिका 2 में दिखाए गए विभिन्न ऑपरेटरों के परिणाम।

टेबल 2

ऑपरेटर	विवरण	उदाहरण
=	जांचता है कि क्या दो ओपरेंडों का मान बराबर है या नही। यदि हां, तो स्थिति सच है।	(A = B) is False
<>	जांचता है कि क्या दो आपरेंड का मान बरारब है या नहीं। यदि मान बराबर नहीं हैं, तो स्थिति सत्य है।	(A <> B) is True
>	चेक करता है कि बाएं आपरेंड का मान दाएं आपरेंड के मान से अधिक है। यदि हां, तो स्थिति सच है।	(A > B) is False
<	चेक करता है कि बाएं आपरेंड का मान दाएं आपरेंड के मान से कम है। यदि हां, तो स्थिति सच है।	(A < B) is True
>=	चेक करता है कि बाएं आपरेंड का मान दाएं आपरेंड के मान से अधिक या बराबर है या नहीं। यदि हां, तो स्थिति सच है।	(A >= B) is False
<=	चेक करता है कि बाएं आपरेंड का मान सही आपरेंड के मान से कम है। यदि हां, तो स्थिति सच है।	(A <= B) is True

लॉजिकल ऑपरेटर्स (The Logical Operators)

VBA द्वारा निम्न लॉजिकल या तार्किक ऑपरेटर का समर्थन किया जाता है।

मान लें कि वेरिएबल A का मान 10 है और B का मान 0 है विभिन्न तार्किक ऑपरेटर पर परिणाम टेबल 3 में दिखाया गया है।

टेबल 3

ऑपरेटर	**विवरण**	**उदाहरण**
AND	तार्किक AND ऑपरेटर कहा जाता है। यदि दोनों स्थितियां सत्य हैं, तो अभिव्यक्ति सच है।	a<>0 AND b<>0 is False
OR	लॉजिकलOR ऑपरेटर कहा जाता है। यदि दो स्थितियों में से कोई भी सत्य है, तो स्थिति सत्य है।	a<>0 OR b<>0 is true
NOT	तार्किक NOT ऑपरेटर कहा जाता है। अपने ऑपरेंड की तार्किक स्थिति को उलट करने के लिए प्रयुक्त होता है। यदि कोई शर्त सत्य है, तो लॉजिकल NOT ऑपरेटर इसे गलत बना देगा।	NOT(a<>0 OR b<>0) is false
XOR	लॉजिकल बहिष्करण कहा जाता है। यह NOT और OR ऑपरेटर का संयोजन है। यदि अभिव्यक्तियों में से एक और केवल एक ही सत्य होने का मूल्यांकन करता है, तो परिणाम सही होता है।	(a<>0 XOR b<>0) is true

Concatenation ऑपरेटर (The Concatenation Operators)

निम्नलिखित, कॉन्सटेनेशन ऑपरेटरों को VBA द्वारा समर्थि किया जाता है।

मान लें, वेरिएबल A रखता है 5 और वेरिएबल B रखता है 10, टेबल 4 में दिखाए गए विभिन्न **Concatenation** ऑपरेटरों का परिणाम।

टेबल 4

ऑपरेटर	**विवरण**	**उदाहरण**
+	वेरिएबल के रूप में दो नाम जोड़ता है। मूल्य संख्यात्मक हैं।	A + B 15 देंगे
&	दो मानों को जोड़ता है।	A & B 510 देंगे

मान लें वेरिएबल A = “Microsoft” और वेरिएबल B = “VBScript”,
विभिन्न concatenation का परिणाम टेबल 5 में दिखाई देगा।

टेबल 5

ऑपरेटर	**विवरण**	**उदाहरण**
+	दो मानों को जोडता है।	A + B MicrosoftVBScript देंगे
&	दो मानों को जोडता है।	A & B MicrosoftVBScript देंगे

नोट : कॉन्सटेनेशन ऑपरेटर दोनों संख्याओं और स्ट्रिंग के लिए उपयोग किया जा सकता है। आउटपुट संदर्भ पर निर्भर करता है, यदि चर संख्यात्मक मान या स्ट्रिंग मान रखते हैं।

प्राथमिकता (Precedence)

जब अभिव्यक्ति में कई ऑपरेशन होते हैं, तो प्रत्येक भाग का मूल्यांकन पूर्व निधार्रित क्रम में किया जाता है, जिसे ऑपरेटर प्राथमिकता कहा जाता है। जब ऑपरेटरों की वही प्राथमिकता होती है तो उनका मूल्यांकन बाएं से दाएं से किया जाता है। पैराथेसिस का उपयोग ऑर्डर ओवरराइड करने और अभिव्यक्ति के कुछ हिस्सों का मूल्यांकन करने के लिए किया जा सकता है। कोष्ठक के अंदर संचालन हमेशा उन लोगों के सामने किया जाता है।

जब ऑपरेटरों की एक श्रृंखला एक ही अभिव्यक्ति में दिखाई देती है तो एक सख्त आदेश होता है जिसमें उनका मूल्यांकन किया जाएगा। प्राथमिकता के नियम कंम्प्यूटर को बताते हैं, जो ऑपरेटरों को पहले मूल्यांकन करना है। प्राथमिकता के नियम कम्प्यूटर को बताते हैं, किस ऑपरेटरों को पहले मूल्यांकन करना है।

जब ऑपरेटरों की एक श्रृँखला एक ही एक्सप्रेशन में आती है तब उनकी गणना का एक सख्त प्रक्रम होता है।

प्राथमिकता के नियम कोम्पाइलर को बताते हैं कि पहले कौन-से ऑपरेटर की गणना करनी है।

कोष्टक ऑपरेटर की वरीयता को क्रम देने के लिए उपयोग किए जाते है। ऑपरेटरों का इस क्रम में मूल्यांकन किया जाता है। गणितीय, कांकेतेनेशन, रिलेशनल और लॉजिकल।

टेबल 6 ऑपरेटर के वरीयता क्रम को दिखाता है।

टेबल 6

क्रम	ऑपरेटर	चिन्ह
1	घातांक	^
2	नेगेशन	-
3	गुणा	*
3	भाग	/
4	पूरांक परिणाम के साथ भाग	\
5	शेषफल	Mod
6	जोड़	+
6	घटाना	-
7	स्ट्रिंग जोड़ना	&
8	असाइंमेंट या बराबर	=
8	के बराबर नहीं	<>
8	इससे कम	<
8	इससे ज्यादा	>
8	इससे कम या बराबर	<=
8	इससे ज्यादा या बराबर	>=
9	Not	NOT
10	And	AND
11	Or	OR
12	एक्सक्लूसिव OR	XOR
13	इक्विवालेइस	EQV
14	इम्पिलकिशेन	IMP

टेबल 7 एक्सप्रेशन को एवं के चरण मूल्यांकन और परिणाम को दर्शा रहा है।

एक्सप्रेशन	पहला चरण	दूसरा चरण	तीसरा चरण	परिणाम
3^(15/5)*2-5	3^3*2-5	27*2-5	54-5	49
3^((15/5)*2-5)	3^(3*2-5)	3^(6-5)	3^1	3
3^(15/(5*2-5))	3^(15/(10-5))	3^(15/5)	3^3	27

VBA संदेश बॉक्स और इनपुट बॉक्स (VBA Message boxes and Input boxes)

उद्देश्य : इस पाठ के अन्त में आप निम्नलिखित कार्य करने योग्य होंगे :
- **VBA में संदेश बॉक्स और इनपुट बॉक्स के उपयोग को बताना**
- **Msgbox विधि और msgbox फंक्शन की व्याख्या करना**
- **Inputbox विधि और inputbox फंक्शन की व्याख्या करना।**

परिचय (Introduction)

कई एप्लिकेशन उपयोगकर्ताओं से आवश्यक कार्रवाई करने के लिए डेटा इनपुट पर निर्भर करते हैं। एक्सेल VBA में बहुत उपयोगी फंक्शन हैं जो आपको अपने अनुप्रयोगों के लिए इनपुट इकट्ठा करने की अनुमति देते हैं। VBA आपको अपयोगकर्ता इनपुट प्राप्त करने के लिए संदेश बॉक्स, उपयोगकर्ता इनपुट फॉर्म और इनपुटबॉक्स बनाने की अनुमति देता है। VBA संदेश बॉक्स उपयोगकर्ता को जानकारी देने और प्रोग्राम चलाने के दौरान उपयोगकर्ता से जानकारी प्राप्त करने का एक तरीका प्रदान करते हैं। इनपुटबॉक्स फ़ंक्शन का उपयोग उपयोगकर्ता को मान दर्ज करने के लिए संकेत देने के लिए किया जा सकता है।

मेसेज बॉक्स (Message Box)

VBA में, मेसेज बॉक्स दो मूल श्रेणियों में आते हैं। MsgBox विधि और MsgBox फंक्शन।

मेसेज बॉक्स मेथड (The MsgBox Method)

मेसेज बॉक्स मेथड का प्रयोग उपयोगकर्ता को पूर्व-परिभाषित संदेश प्रदर्शित करने के लिए किया जाता है। इसमें उपयोगकर्ता को संदेश को खारिज करने की अनुमति देने के लिए एक ही कमांड बटन "OK" होता है और उन्हें प्रोग्राम में काम करना जारी रखने से पहले ऐसा करना होता है।

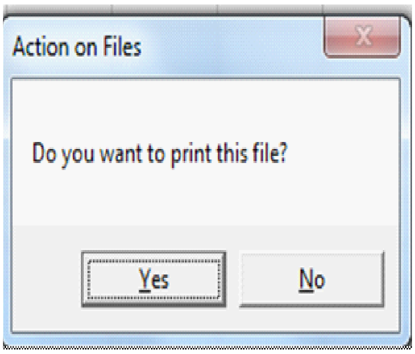
VBA में मेसेज बॉक्स (msgbox) का मूल फॉर्म यह हैं Msgbox("message")

उदाहरण :

```
Sub result()
Msgbox("congratulations")
End sub
```

यह ऐसा मेसेज बॉक्स प्रदर्शित करेगा जैसा Fig 1 में दिखायेगा।

Fig 1



VBA मेसेज बॉक्स में बटन को कस्टमाइज करना (Customize the buttons in a VBA message box)

Msgbox() को उस पर रखे बटन और आइकन को बदलकर अनुकूलित किया जा सकता है।

VBA मेसेज बॉक्स में इस्तेमाल किए जा सकने वाले विभिन्न बटन और आइकनों की एक सूची टेबल 1 में दिखाया गया है।

उदाहरण के लिए, Msgbox() में एक आइकन और शीर्षक जोड़ने के लिए हम निम्नलिखित कोड लिख सकते हैं।

```
Sub test()
Dim n As Integer
n = MsgBox("Congratulations", vbExclamation, "result")
End Sub
```

यह Fig 2 में निम्नलिखित परिणाम देगा।

Fig 2

संदेश बॉक्स फंक्शन (The MsgBox Function)

MsgBox फ़ंक्शन एक डायलॉग बॉक्स में एक संदेश प्रदर्शित करता है, उपयोगकर्ता को बटन पर क्लिक करने की प्रतीक्षा करता है, और फिर एक पूर्णांक देता है जो दर्शाता है कि उपयोगकर्ता द्वारा कौन सा बटन क्लिक किया गया था।

Msgbox() फंक्शन का सिंटेक्स इस प्रकार है -

Return value = MsgBox(Prompt, Button and Icon types, Title, Help File, Help File Context)

टेबल 1

कांस्टेंट	विवरण
vbOKOnly	यह एक एकल OK बटन प्रदर्शित करता है।
vbOKCancel	यह दो बटन OK और Cancel प्रदर्शित करता है।
vbAbortRetryIgnore	यह तीन बटन Abort, Retry, और Ignore प्रदर्शित करता है।
vbYesNoCancel	यह तीन बटन Yes, No, और Cancel प्रदर्शित करता है।
vbYesNo	यह दो बटन Yes और No प्रदर्शित करता है।
vbRetryCancel	यह दो बटन Retry and Cancel प्रदर्शित करता है।
vbCritical	यह दो बटन Critical Message icon प्रदर्शित करता है।
vbQuestion	यह एक Query icon प्रदर्शित करता है।
vbExclamation	यह एक Warning Message icon प्रदर्शित करता है।
vbInformation	यह एक Information Message icon प्रदर्शि करता है।
vbDefaultButton1	First button डिफ़ॉल्ट के रूप में माना जाता है।
vbDefaultButton2	Second button डिफ़ॉल्ट के रूप में माना जाता है।
vbDefaultButton3	Third button डिफ़ॉल्ट के रूप में माना जाता है।
vbDefaultButton4	Fourth button डिफ़ॉल्ट के रूप में माना जाता है।
vbApplicationModal	जब तक उपयोगकर्ता संदेश बॉक्स का जवाब नहीं देता तब
vbSystemModal	यह तब तक सभी एप्लिकेशन को निलंबित करता है जब तक कि उपयोगकर्ता संदेश बॉक्स का जबाव नहीं देता।
vbMsgBoxHelpButton	यह संदेश बॉक्स में एक सहायक बटन जोड़ता है।
VbMsgBoxSetForeground	यह सुनिश्चित करता है कि संदेश बॉक्स विंडो अग्रभूमि है।
vbMsgBoxRight	यह टेक्स्ट को दाएं संरेखित करने के लिए सेट करता है।
vbMsgBoxRtlReading	यह विकल्प निर्दिष्ट करता है कि टेक्स्ट दाएं से बाएं रूप में दिखनी चाहिए।

जहाँ :

रिटर्न मान (Return Value): उस एक्शन को इंगित करता है, उपयोगकर्ता ने लिख, जब संदेश बॉक्स उसे दिखाया गया था।

प्रोम्प्ट (Prompt): यह संदेश है, संदेश बॉक्स के मुख्य भाग में निहित संदेश है।

बटन और आइकन के प्रकार (Button and Icon Types): यह बटन और आइकन और उनके प्लेसमेंट के सेट को निर्दिष्ट करता है क्योंकि वे उपयोगकर्ता को दिखाई देंगे।

सहायता फाइल (Help File): यह एक सहायता फाइल का पथ है जिसे उपयोगकर्ता इस विषय पर संदर्भित कर सकता है।

सहायता फाइल संदर्भ (Help File Context): यह सहायता फाइल के उस भाग का सूचक है जो विशेष रूप से इस संदेश से संबंधित है।

MsgBox फंक्शन द्वारा लौटाया गया मान (Values returned by MsgBox Function):

VBA MsgBox फंक्शन उपयोगकर्ता इनपुट के आधार पर एक मान देता है। ये मान टेबल 2 में दिखाए गए कोई भी हो सकते हैं।

एक Msgbox फंक्शन उदाहरण नीचे उल्लिखित कोड में दिखया गया

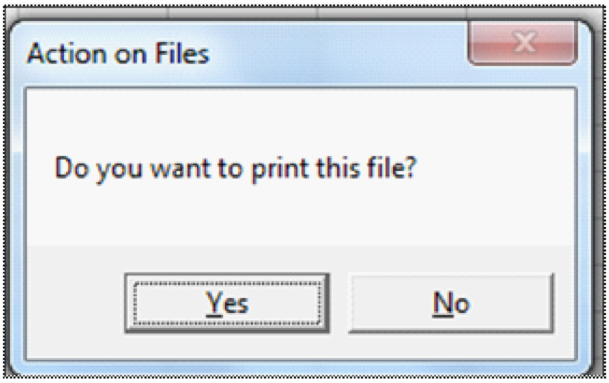
```
Sub test()

Dim n As Integer

 n = MsgBox("Do you want to print this file?", vbYesNo,
"Action on Files")

End Sub
```

टेबल 2

मान	विवरण
1	निर्दिष्ट करता है कि OK बटन क्लिक किया गया है।
2	निर्दिष्ट करता है कि Cancel करें बटन क्लिक किया गया है।
3	निर्दिष्ट करता है कि Abort button डिक है।
4	निर्दिष्ट करता है कि Retry button डिस्क किया गया है।
5	निर्दिष्ट करता है कि Ignore button क्लिक किया गया है।
6	निर्दिष्ट करता है कि Yes बटन डिस्क किया गया है।
7	निर्दिष्ट करता है कि No बटन क्लिक किया गया है।

यह Fig 3 द्वारा दर्शाया गया परिणाम देगा।

Fig 3

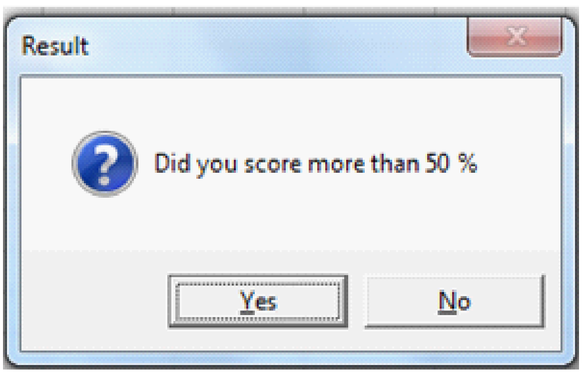


Msgbox() रिटर्न मानों को पढ़ना (Reading the Msgbox() return values)

MsgBox() द्वारा दिए गए मान के आधार पर, निर्णय किए जा सकते हैं।

उदाहरण के लिए, यहाँ उल्लिखित कोड मेसेज बॉक्स प्रदर्शित करेगा, और जब उपयोगकर्ता, "Yes" पर क्लिक करेगा, तो यह एक बधाई संदेश प्रदर्शित करेगा। यदि उपयोगकर्ता "No" पर क्लिक करता है, जो Fig 4 में दिखाए गए अनुसार एक और संदेश "Better Luck Next time" दिखाई देता है।

Fig 4




```
Sub test()
Dim n As Integer
n = MsgBox("Did you score more than 50 % ", vbYesNo + vbQuestion, "Result")
If n = 6 Then
MsgBox ("Congratulations")
Else
MsgBox ("Better Luck Next Time")
 End If
End Sub
```

इनपुट बॉक्स (Input box)

उपयोगकर्ता से इनपुट स्वीकार करने के लिए इनपुट बॉक्स का उपयोग दो तरीकों से किया जाता है-इनपुट बॉक्स फंक्शन और इनपुट बॉक्स विधि। इनपुट बॉक्स विधि इनपुट बॉक्स फंक्शन से अलग होती है। जिसमें यह उपयोगकर्ता के इनपुट के चुनिंदा सत्यापन की अनुमति देता है, और यह हो सकता है माइक्रोसॉफ्ट एक्सेल ऑब्जेक्ट्स, त्रुटि मान, और सूत्रों के साथ प्रयोग किया जाता है।

ध्यान दें कि **Application.Input Box Input Box method;** को कॉल को रति है। बिना क्वालीफ़ायर का **Input Box InputBox** फंक्शन को कॉल करता है।

इनपुट बॉक्स फंक्शन (Input Box Function)

इनपुट बॉक्स फंक्शन उपयोगकर्ता इनपुट के लिए एक संवाद बॉक्स प्रदर्शित करता है। यह डायलॉग बॉक्स में दर्ज की गई जानकारी को संग्रहित करता है। इनपुट बॉक्स फंक्शन के लिए सिंटेक्स :

InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])

सरल रूप में इनपुट बॉक्स फंक्शन ऐसा दिखेगा

like:n = Inputbox("Enter your Age")

इनपुट बॉक्स मेथड (The InputBox Method)

जब हम इनपुट बॉक्स फंक्शन के सामने "Application" लगते हैं तो हमें इनपुट बॉक्स मेथड मिलता है जो हमें यह निर्दिष्ट करने की अनुमति प्रदान करता है हम किस तरह का मान संगृहित करना चाहेंगे यानि **Application.InputBox**

इसका सिंटेक्स है **: Input Box(Prompt, Title, Default, Left, Top, HelpFile, HelpContextId, Type)**

प्रोम्प्ट, टाइटल और डिफॉल्ट इनपुट बॉक्स फंक्शन जैसे ही है। हाँलाकि, आखरी आर्ग्युमेंट **"Type"** है जो यह बताती है कि हमें किस तरह का डाटा संगृहित करना है। यह नीचे दर्शाया गया है :

टाइप:=0 एक फॉर्मूला

टाइप:=1 एक संख्या

टाइप:=2 टेक्स्ट (एक स्ट्रिंग)

टाइप: = 4 A logical value (True or False)

टाइप: = 8 A cell reference, as a Range object

टाइप: = 16 An error value, such as #N/A

टाइप := 64 An array of values

The following is an example of an InputBox method

```
Sub test()

Dim n As Integer

n = Application.InputBox("Enter you age", "Personal Details", , , , , , 1)

 'Exit sub if Cancel button used

If n > 60 Then

MsgBox "You are eligible for senior citizen's concession"

Else

MsgBox ("No concession")

End If

End Sub
```

VBA में निर्णय लेने वाली स्टेटमेंट (Decision making statements in VBA)

उद्देश्य : इस पाठ के अन्त में आप निम्नलिखित कार्य करने योग्य होंगे :
- **"if... Then" कथन का उपयोग करके निर्णय लेने कि प्रक्रिया का वर्णन करना**
- **"ladder off" और "nested if" कथन के उपयोग का वर्णन करना**
- **"select ....case" कथन के उपयोग की व्याख्या करना।**

परिचय (Introduction)

एक प्रोग्राम में, स्टेटमेंट का एक सेट क्रमशः में क्रम निष्पादित किया जाता है, जिसमें वे प्रकार होती हैं। ऐसा तब होता है जब कोई निर्णय लेने या पुनरावृत्ति वाली स्टेटमेंट शामिल नहीं होती है लेकिन वास्तविकता में, ऐसी कई स्थितियाँ हो सकती हैं जहाँ हमें कुछ शर्तों को सही या गलत होने के आधार पर स्टेटमेंट के निष्पादन के आदेश को बदलना पड़ सकता है। कुछ उदाहरण हो सकते हैं :

a यह तय करने के लिए कि क्या प्रशिक्षु को "उत्तीर्ण" किया जाना है या नहीं।

b एक छात्र द्वारा प्राप्त ग्रेड प्रदर्शित करने के लिए।

c संख्याओं जैसे किसी विशेष डेटा प्रकार के इनपुट को स्वीकार करने के लिए।

d यह तय करने के लिए कि कोई संख्या प्रमुख है या नहीं।

e यह तय करने के लिए कि कोई स्ट्रिंग पैलिंड्रोम है अविभाज्य है या नहीं।

f कुछ शर्तों के आधार पर वेतन, कर, कमीशन आदि की गणना करने के लिए आदि।

g किसी क्रिया को निश्चित समय तक दोहराने के लिए, जब तक निश्चित सीमा तक ना पहुँच जाये।

निर्णय लेने की प्रकिया बुद्धिमानी से व्यावहारिक समस्याओं को हल कर सकती हे और उपयोगकर्ता को उपयोगी आउटपुट या फीडबैक प्रदान कर सकती है। प्रोग्राम प्रवाह को नियंत्रित करने और निर्णय लेने के लिए, हमें रिलेशनल ऑपरेटरों और तार्किक ऑपरेटरों को **if control** संरचना के साथ एक साथ उपयोग करने की आवश्यकता है।

डिसिशन मेकिंग स्टेटमेंट के लिए आवश्यक है कि प्रोग्रामर एक या अधिक स्थितियों को प्रोग्राम द्वारा मूल्यांकन या परीक्षण करने के लिए निर्दिष्ट करें, यदि कोई कंडीशन स्टेटमेंट या कथन के साथ निष्पादित किया गया हो, तो कंडीशन को सत्य माना जाता है, और यदि कंडीशन गलत साबित होती है, अन्य बयान निष्पादित किया जाएगा। टेबल 1 VBA में आमतौर पर उपयोग किए जाने वाले निर्णय लेने के बयान दिखाता है।

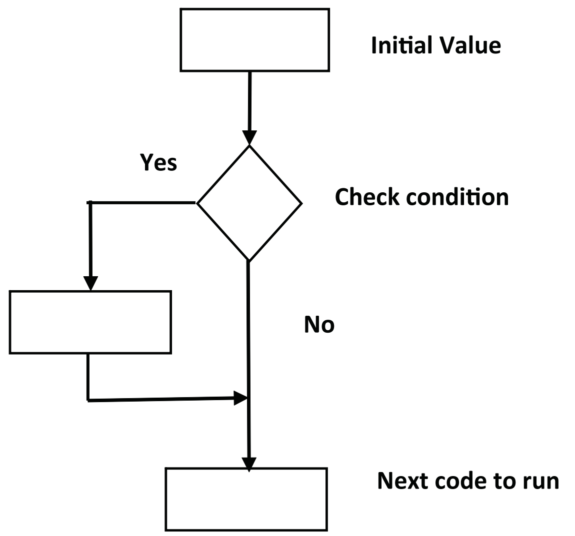
If … Then स्टेटमेंट (The If … Then Statement)

यह नियंत्रण कथन का सबसे सरल रूप है। अक्सर निर्णय निष्पादन के नियंत्रण प्रवाह को बदलने और बदलने में उपयोग किया जाता है। if-then कथन के लिए सिंटेक्स है :

```
If CONDITION Then
' code if the condition is met
End If
```

एक समान्य If स्टेटमेंट के लिए फ्लो चार्ट Fig 1 में दिखाया गया है।

Fig 1



यहाँ कंडीशन एक एक्सप्रेशन संदर्भित करती है जो बूलियन प्रकार के परिणाम के माध्यम से परिणाम देती है, यानी। सही या गलत। उदाहरण के लिए "अगर आयु <18" स्टेटमेंट का परीक्षण करेगा, यदि वेरिएबल "आयु" का मान 18 से कम है या नहीं। यदि कंडीशन सही साबित होती है, तो If स्टेटमेंट के अंदर का कोड निष्पादित किया जाएगा। उदाहरण के लिए :

```
If (age < 18) Then
debug.print "Not Eligible"
End If
```

निम्न उदाहरण टेक्स्ट बॉक्स में संख्या के मान का परीक्षण करता है। और निर्णय लेता है।

```
Private Sub Button1_Click()
Dim n As Integer
'Enter the number of items sold by the agent
n = val(TextBox1.Text)
If n> 100 Then
Label1.Caption = " You are entitled for a commission of Rs. 10000"
End If
End Sub
```

If Then…. Else स्टेटमेंट (The If Then…. Else Statements)

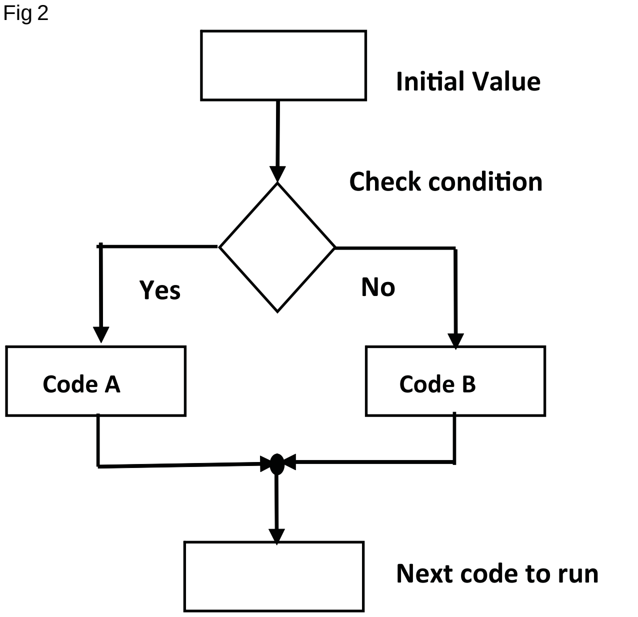
जब एक कंडीशन के सही होने पर एक एक्शन लेना हो और कंडीशन के गलत होने पर दूसरा एक्शन लेना हो तो If Then…. Else स्टेटमेंट उपयोग में लायी जाती है।

The syntax for the If Then … Else statements is as follows

```
If CONDITION Then

' code if the condition is met

Else

' code if the condition is not met

End If
```

एक प्रारूपि If Then … Else संरचना का फ्लो चार्ट इस प्रकार होगा जैसा Fig 2 में दिखाया गया है।

Fig 2



If Then … Else का उदाहरण निचे दिया हुआ है। यह उदाहरण जाँचेगा कि यदि यूजर द्वारा इंटर की गयी टैक्सेबल इनकम 250000 है या नहीं और यदि यह सही है तो आपके सामने एक संदेश प्रदर्शित होगा जो यह कहेगा कि यूजर को इनकम टैक्स देने की आवश्यकता नहीं है एवं यदि ऐसा नहीं है तो दूसरा मेसेज बॉक्स यह कहेगा कि यूजर को टैक्स देना पड़ेगा। यदि हाँ है तो एक मैसेज बॉक्स दिखाई देगा जिसमें कह गया होगा कि प्रयोगकर्ता को आयकर नहीं देना होगा। अन्यथा दूसरा बड़ा बॉक्स कहेगा उपयोगकर्ता को आयकर भरने के लिए (fig 1)

```
Sub test()

Dim income As Long

income = Application.InputBox("Enter you Taxable income")

If income< 250000 Then

MsgBox "You need not pay any income tax"

Else

Msg Box ("You must pay income tax")

End If

End Sub
```

मल्टीपल If स्टेटमेंट का उपयोग (Using Multiple If Statements)

कभी - कभी कंडीशन सिर्फ एक कंडीशन के आधार पर "True" या "False" लौटाने के लिए नहीं उयोग की जाती बल्कि एक से अधिक कंडीशन के लिए उपयोग की जाती है। ऐसे मामलों में If Then …. Else स्टेटमेंट उपयोग में लायी जा सकती है।

यह दो तरीकों से उपयोग की जा सकती है -

1 Ladder If and

2 Nested if

Ladder If स्टेटमेंट (Ladder If statements)

ladder if स्टेटमेंट यह जांच करती है कि कंडीशन 1, कंडीशन 2 … आदि मिल रही है एवं जो कंडीशन सही होती है उसके आधार पर निर्णय लिया जाता है। लैडर If का सिंटेक्स इस प्रकार है :

```
if(boolean_expression 1)

{

/* Executes when the boolean expression 1 is true */

}

else if( boolean_expression 2)

{

 /* Executes when the boolean expression 2 is true */

}

else if( boolean_expression 3)

{

 /* Executes when the boolean expression 3 is true */

}

else

{

 /* executes when the none of the above condition is true */

}
```

ladder if स्ट्रक्चर का निम्नलिखित उदाहरण है

```
Sub grades()

Dim marks As Integer

marks = InputBox("Enter you marks")
If marks >= 80 Then
MsgBox "Distinction"

ElseIf marks >= 70 Then

MsgBox "A Grade"

ElseIf marks >= 60 Then
MsgBox "B Grade"

ElseIf marks >= 40 Then

MsgBox "C Grade"

Else

MsgBox "Failed"

End If

End Sub
```

यह प्रोग्राम उपयोगकर्ता द्वारा दर्ज अंकों के आधार पर ग्रेड प्रदर्शित करेगा।

पेस्टेड If स्टेटमेंट्स (Nested If statements)

कभी - कभी केवल एक शर्त का मूल्यांकन करना आवश्यक होता है यदि पहले की कंडीशन पूरी हो जाती है तो ऐसे मामलों में If Then स्टेटमेंट बाहर की If Then स्टेटमेंट के अंदर उपयोग की जाती है। इस संरचना को नेस्टेड If स्ट्रक्चर भी कहते हैं। नेस्टेड If स्ट्रक्चर का सिंटेक्स इस प्रकार है :

```
If(Boolean_expression 1)
{
//Executes when the Boolean expression 1 is true
If(Boolean_expression 2)

{

//Executes when the Boolean expression 2 is true

}

}
```

उदाहरण के लिए, एक निश्चित भर्ती स्थिति में कहा गया है कि पात्र घोषित करने के लिए उम्मीदवार के पास कम से कम 5 साल का अनुभव होना चाहिए और परीक्षा में एलिस्ट 75% अंक भी बनाए होने चाहिए। ऐसी स्थिति में पहली कंडीशन **experience >= 5 years** के लिए जाँची जाएगी और यदि यह कंडीशन सच होती है तो दूसरी कंडीशन की जाँच होगी। यदि पहली कंडीशन गला निकलती है तो कण्ट्रोल End if स्टेटमेंट के बाद की स्टेटमेंट पर चला जायेगा। उल्लेखित किये उदाहरण का निम्नलिखित कॉड है।

```
Sub job_test()
Dim experience, marks As Integer
experience = InputBox("Enter your work experience in years")
If experience >= 5 Then
marks = InputBox("Enter you marks percentage")
If marks >= 75 Then
MsgBox (" You are eligible for the post")
Else
MsgBox (" You are NOT eligible for the post")
End If
Else
MsgBox (" You are NOT eligible for the post")
End If
End Sub
```

If स्ट्रक्चर में लॉजिकल ऑपरेटरों का उपयोग (Using Logical operators in If Structure)

लॉजिकल ऑपरेटर And, Or और Not If स्ट्रक्चर में भी उपयोग किये जाते हैं और वेसा ही परिणाम देते हैं जैसा नेस्टेड If में दिया था।

उदाहरण के लिए कंडीशनल स्टेटमेंट में And ऑपरेटर का उपयोग करके ऊपर दी हुई कंडीशन को जांचा जा सकता है।

```
Sub job_test()

Dim experience, marks As Integer
experience = InputBox("Enter your work experience in years")

marks = InputBox("Enter you marks percentage")

If experience >= 5 And marks >= 75 Then
MsgBox (" You are eligible for the post")
Else

MsgBox (" You are NOT eligible for the post")

End If
End Sub
```

सलेक्ट... केस (Select...Case)

डिसिशन मेकिंग को लागू करने का अन्य तरीका है सेलेक्ट स्टेटमेंट का उपयोग करना। Select...Case स्टेटमेंट एक चार को एक से अधिक बार जाँचने के लिए उपयोग की जाती है और उस पर आधारित कोई निर्दिष्ट कार्यवाही या एक्शन लेती है।

सेलेक्ट केस स्टेटमेंट का उपयोग करना हमेश एक अच्छा अभ्यास होता है। जब कई If-Else स्थितियाँ शामिल होती हैं। जैसे-जैसे If-Else कोन्दितिओन्स बढ़ती है, वैसा ही डीबगिंग करना और कण्ट्रोल के प्रवाह को समझना एक कठिन काम हो जाता है।

Select...Case का सिंटेक्स है :

```
Select Case VARIABLE
Case VALUE1
' code to run if VARIABLE equals Value1
Case VALUE2
' code to run if VARIABLE equals Value2
Case Else
' code to run for remaining cases
```

ऐण्ड सेलेक्ट (End Select)

उदाहरण के लिए, यह प्रोग्राम यूजर को गेम का नाम टाइप करने के लिए बोलता है और गेम को खोलने वाले प्लेयर की संख्या प्रदर्शित करता है।

```
Sub players()
Dim game As String
game = InputBox("enter the name of the game")
game = LCase(game)
Select Case game
Case "tennis"
Debug.Print "2 Players."
Case "cricket"
Debug.Print "11 Players."
Case "volleyball"
Debug.Print "5 Players."
Case "baseball"
Debug.Print "9 Players."
Case Else
Debug.Print "I have no idea."
End Select
End Sub
```

IIF फंक्शन (IIF Function)

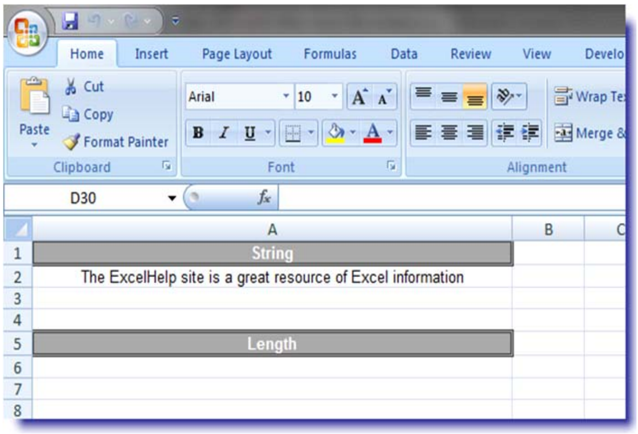
यह function कोई एक्सप्रेशन का मूल्यांकन करने और मूल्यांकन के परिणाम पर आधारित दो में से कई एक एक्शन निष्पादित करता है।

यदि (मान > 10 है, यह एक्शन निष्पादित होगा और यदि मान <= 10, तो यह एक्शन निष्पादित होगा।

यह फंक्शन VBA कोड के भीतर और एक्सेल फंक्शन के रूप में भी उपलब्ध है। आमतौर पर IIF फंक्शन का उपयोग त्वरित तार्किक आंकलन करने के लिए किया जाता है और इसे अधिक जटिल मूल्यांकन करनके के लिए दिया जा सकता है। हांलाकि यह याद रखना महत्त्वपूर्ण है कि नेस्टेड IF कथन बहुत जटिल और, सपोर्ट और बनाए रखने के मुश्किल हो सकता है।

अब चलिए एक उदाहरण देखें। आइए मान लीजिए कि कम केवल स्ट्रिंग की लंबाई की गणना करना चाहते हैं, यदि इसमें Excel Help और Excel (Fig 3) मान शामिल हैं।

Fig 3



यह ध्यान रखना महत्त्वपूर्ण है कि हम वर्कशीट पर सभी पंक्तियों के माध्यम से चलाने के लिए हमारे फॉर-नेक्स्ट लूप में से एक में IIF कथन का इस्तेमाल कर सकते थे।

कोड (Code)

```
Dim StringToProcess As String'Variable to hold the string to be processed
StringToProcess = ActiveSheet.Cells(2, 1).Value
ActiveSheet.Cells(6, 1).Value = IIf(InStr(StringToProcess, "ExcelHelp") > 0, IIf(InStr(StringToProcess, " Excel ") > 0, Len(StringToProcess), 0), 0)
```

परिणाम (Output) (Fig 4)

Fig 4