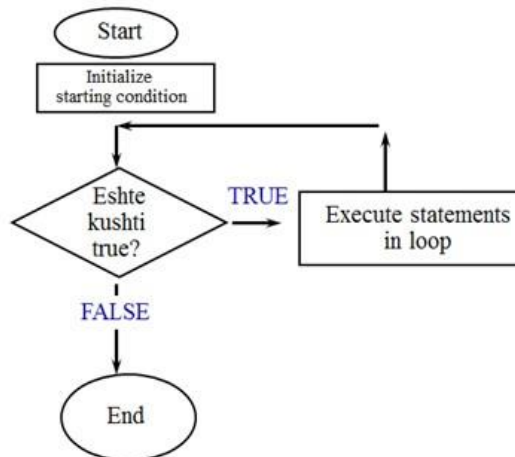


Ciklet - Instruksionet përsëritëse në VBA

Ciklet ose instruksionet përsëritëse në VBA ju lejojnë të përdorni një ose më shumë rreshta të kodit në mënyrë të përsëritur. Ju mund të përsërisni deklaratat në cikël derisa një kusht është i vërtetë, derisa një kusht është i gabuar, për një numër të caktuar herësh ose të paktën një herë. Bllokskema e mëposhtme tregon një cikël që kryen një bllok veprimesh deri sa një kusht të bëhet i vërtetë.



Nëse, në programin tuaj, ju duhet të kryeni të njëjtin detyrë (ose përsërisni të njëjtën pjesë të kodit) shumë herë, kjo mund të bëhet duke përdorur një nga ciklet VBA.

1- Cikli Do... Loop përsërit një bllok të veprimeve, përderisa një kusht logjik është i vërtetë ose gjersa të bëhet e vërtetë. Ndërtimi i Do...Loop ju lejon të provoni një kusht në fillim ose në fund të një cikli loop. Përsëritja e bllokut të veprimeve në ciklin përderisa (while) bëhet për sa kohë kushtet logjike janë të vërteta ose në ciklin gjersa (until) derisa kushtet logjike nuk janë të vërteta. Testimi i kushteve logjike i çdo cikli kryhet ose në fillim ose në fund të tij.

Sintaksa e Do...Loop është dhënë si më poshtë:

Do While <kushti logjik>

[veprime]

[Continue Do]

[veprime]

[Exit Do]

[veprime]

Loop

ose

Do Until <kushti logjik>

[veprime]

[Continue Do]

[veprime]

[Exit Do]

[veprime]

Loop

ose

Do

[veprime]
 [Continue Do]
 [veprime]
 [Exit Do]
 [veprime]

Loop While <kushti

logjik>

ose

Do

[veprime]
 [Continue Do]
 [veprime]
 [Exit Do]
 [veprime]

Loop Until <kushti logjik>

ku:

Do E domosdoshme, fjale çelës, Pjesë e ciklit

While E domosdoshme, përsërit veprim derisa kushti është I vërtetë dhe kur bëhet FALSE ndërpriten veprimet e ciklit

Until E domosdoshme, përsërit veprim derisa kushti është False dhe kur bëhet TRUE ndërpriten veprimet e ciklit

Condition E domosdoshme, kushti logjik, i cili ka dy vlera TRUE dhe FALSE

Veprime E domosdoshme, veprime

Continue Do **Fakultative**, transferon kontrollin në përsëritjen e radhës të ciklit.

Exit Do **Fakultative**. Transferon kontrollin nga cikli në veprimin pasardhës

Loop E domosdoshme, përfundon ciklin!

- Ju përdorni ciklin Do...Loop kur doni të përsërisni një sërë veprimesh me një numër të pafund herësh, derisa plotësohet kusht logjik. Nëse doni të përsërisni veprimet për një numër të caktuar herësh përdorni ciklin For...Next pasi është një zgjedhje më e mirë.
- Ju mund të përdorni ose While ose Until për të specifikuar kushtin logjik, por jo të dyja.
- Ju mund ta testoni kushtin logjik vetëm një here në fillim ose në fund të ciklit. Nëse testoni kushtin logjik në fillim të ciklit (në ciklin DO...<kushti logjik>), cikli mund të mos zgjasë edhe një herë. Nëse testoni kushtin logjik në fund të ciklit (në ciklin Loop...<kushti logjik>), cikli ekzekutohen të paktën një herë.
- Kushti logjik zakonisht vjen si rezultat i krahasimit të dy vlerave, por mund të jetë çdo shprehje që vlerëson vlerën logjike të të dhënave (True ose False). Kjo përfshin vlerat e llojeve të të dhënave të tjera, të tilla si llojet numerike, të cilat janë konvertuar në Boolean.

Shembull (shih lab-u4)

'Ndërtoni një program që përdor ciklin **Do While...loop** për të gjetur dhe afishuar shumën e 10 numrave të parë natyrorë.

```

Sub shuma()
Dim i As Integer
Dim s As Integer
s = 0
i = 1
Do While i <= 10
s = s +
i
i = i +
1
Loop
MsgBox "Shuma=" & s
End Sub

```

(lab-u7)

'Ndërtoni një program që përdor ciklin **Do Until...loop** për të gjetur dhe afishuar shumën e 10 numrave të parë natyrorë.

```

Sub shuma()
Dim i As Integer
Dim s As Integer
s = 0
i = 1
Do Until i > 10
s = s + i
i = i + 1
Loop
MsgBox "Shuma=" & s
End Sub

```

(lab-u8)

'Ndërtoni një program që përdor ciklin **Do ...loop while** për të gjetur dhe afishuar shumën e 10 numrave të parë natyrorë.

```

Sub shuma()
Dim i As Integer
Dim s As Integer
s = 0
i = 1

```

```

Do
s = s + i
i = i + 1
Loop While i <= 10
MsgBox "Shuma=" & s
End Sub

```

'Ndërtoni një program që përdor ciklin Do ...loop until për të gjetur dhe afishuar shumën e 10 numrave të parë natyrorë.

```

Sub shuma()
Dim i As Integer
Dim s As Integer
s = 0
i = 1
Do
s = s + i
i = i + 1
Loop Until i > 10
MsgBox "Shuma=" & s
End Sub

```

Instrukcioni Exit Do

Instrukcioni Exit Do mund të ofrojë një mënyrë alternative për të dalë nga cikli Do ... Loop.

Instrukcioni Exit Do transferon kontrollin menjëherë në deklaratën që ndjek fjalën çelës Loop.

Ai përdoret shpesh pasi vlerësohet një kusht, për shembull në një cikël If...Then...Else.

Mund të dëshironi të dilni nga një cikël nëse zbuloni një kusht që e bën të panevojshëm ose të pamundur të vazhdoni përsëritjen, si për shembull një vlerë e gabuar ose një kërkesë për ndërprerje.

Një përdorim i instrukcionit Exit Do është të provosh një kusht që mund të shkaktojë një cikël të pafundem, i cili mund të ekzekutojë një numër të madh apo edhe pafund herësh bllokun e veprimeve.

Ju mund të përfshini një numër të caktuar të instrukcioneve Exit Do kudo në një cikël Do ... Loop.

Shembulli i mëposhtëm përdor ciklin while dhe kushti logjik testohet në fillim të ciklit.

```

Sub shembull()
Dim i As Integer
i = 0
Do While i <= 10
MsgBox ("Vlera e i-se:" & i)
i = i + 1
Loop
End Sub

```

' Output: 0 1 2 3 4 5 6 7 8 9 10

Në shembullin e mëposhtëm, kushti logjik ndalet variabli i është më i madh se 100. Ndërsa If në cikël nëpërmjet instruksionit Exit Do ndalon ciklin kur variabli i është më i madh se 10.

```
Sub shembull()
  Dim i As Integer
  i = 0
  Do While i <= 100
    If i > 10 Then
      Exit Do
    End If
    MsgBox ("Vlera e i-se:" & i)
    i = i + 1
  Loop
End Sub
```

' Output: 0 1 2 3 4 5 6 7 8 9 10

2- Cikli While ... Wend kryen një bllok veprimesh për aq kohë sa kushti logjik i specifikuar pas fjalës While është i vërtetë. Përdorni ciklin While..End While kur dëshironi të përsërisni një sërë deklaratash një numër të pafund herësh, për aq kohë sa një kusht të mbetet i vërtetë.

Sintaksa e saj është dhënë si më poshtë:

While <kushti logjik>

[veprime]

[**Continue While**]

[veprime]

[**Exit While**]

[veprime]

Wend ku:

While E domosdoshme, përsërit veprim derisa kushti të bëhet FALSE

Condition E domosdoshme, kushti logjik, i cili ka dy vlera TRUE dhe FALSE

Veprime E domosdoshme, Veprime

Continue While **Fakultative**, transferon kontrollin në përsëritjen e radhës të ciklit.

Exit While **Fakultative**. Transferon kontrollin nga cikli në veprimin pasardhës

Wend E domosdoshme, përfundon ciklin!

'Ndërtoni një program që përdor ciklin While...Wend për të gjetur dhe afishuar shumën e 10

'numrave të parë natyrorë.

```
Sub shuma()
```

```
  Dim i, s As Integer
```

```
  s = 0
```

```
  i = 1
```

```
  While i <= 10
```

```
    s = s + i
```

```
i = i + 1
Wend
MsgBox "shuma=" & s
End Sub
```

3- Cikli FOR përsërit një bllok veprimesh për një numër të caktuar herësh. **Cikli For ... Next** Sintaksa e saj është si me poshtë:

For counter [As datatype] = start To end [Step step]

[veprime]

[Continue For]

[veprime] [Exit For]

[veprime]

Next [counter]

ku:

For E domosdoshme, fjale çelës, Pjese e ciklit

Counter E domosdoshme, ka vlere numerike dhe është variabël kontrolli për ciklin.

Datatype Fakultative është deklarimi i tipit të dhënës për counter

Start E domosdoshme, vlere numerike, inicializimi i variablit të kontrollit

End E domosdoshme, vlere numerike, vlere përfundimtare e variablit të kontrollit

Step Fakultative. Vlere numerike. Shuma me të cilën counter rritet çdo herë përmes ciklit

veprime E domosdoshme, Veprime

Continue For Fakultative, transferon kontrollin në përsëritjen e radhës të ciklit.

Exit For Fakultative. Transferon kontrollin nga cikli në veprimin pasardhës

Next E domosdoshme, përfundon ciklin.

Vlera e hapit mund të jetë pozitive ose negative. Vlera default për Step-in është 1. Ky parametër përcakton përpunimin e cikleve FOR sipas tabelës së mëposhtme:

Vlera e step-it	Kushti logjik
Pozitive/ zero	Variabël kontrolli <=vlere përfundimtare
Negative	Variabël kontrolli >=vlere përfundimtare

'Ndërtoni një program që përdor ciklin for next për të gjetur dhe afishuar shumën e 10 'numrave të parë natyrorë.

```
Sub shuma()
```

```
Dim i, s As Integer
```

```
s = 0
```

```
For i = 1 To 10
```

```

s = s + i
Next
'by default nëse është rritës 1...10 step +1
'nese është zbritës 10...1 step -1
MsgBox "shuma=" & s
End Sub

```

Ndryshimi i stepit

'Ndërtoni një program që përdor ciklin for next për të gjetur dhe afishuar numrat tek ne segmentin nga 1 ne 10 numra të plote.

```

Sub nrtek()
Dim i As Integer
For i = 1 To 10 Step 2
MsgBox "nr=" & i
Next
End Sub

```

Ushtrime me Instruksionet Përsëritëse

Shembull 1. (shih lab-u1) Ndërtoni një program i cili mblesh numrat nga 1 ne 10 në variablin total dhe e afishon ne ekran.

```

Sub shembull1()
Dim i, total As Integer
For i = 1 To 10
total = total + i
Next i
MsgBox "shuma =" & total
End Sub

```

Shembull 2. (shih lab-u12) Ndërtoni një program i cili vendos ne sheet1 fushe kuq e zi me dhjete rreshta dhe dhjete shtylla.

```

Sub shembull2()
Const NB_CELLS As Integer = 10
Dim offset_row As Integer, offset_col As Integer
offset_row = ActiveCell.Row - 1
offset_col = ActiveCell.Column - 1
For r = 1 To NB_CELLS
For c = 1 To NB_CELLS
If (r + c) Mod 2 = 0 Then
Cells(r + offset_row, c + offset_col).Interior.Color = RGB(200, 0, 0)
Else

```

```

Cells(r + offset_row, c + offset_col).Interior.Color = RGB(0, 0, 0)
End If
Next
Next
End Sub

```

Shembull 3. (shih lab-u2) Ndërtoni një program i cili afishon numrat nga 1 ne 5 ne ekran duke përdorur ciklin FOR.

```

Sub shembull3()
  For i = 1 To 5
    MsgBox i
  Next
End Sub

```

Shembull 4. Ndërtoni një program i cili merr një vlere nga qeliza A1 dhe me pas afishon numrin e përsëritjeve te tij.

```

Sub shembull4()
  Dim max_loops As Integer
  max_loops = Range("A1")
  For i = 1 To 7
    If i > max_loops Then
      Exit For
    End If
    MsgBox i
  Next
End Sub

```

Shembull 5. (shih lab-u15) Ndërtoni një program i cili gjen shumen 50 numrave te plote natyrore dhe me pas e afishon ate.

```

Sub shembull5()
  Dim i, S As Integer
  S = 0
  For i = 1 To 50
    S = S + i
  Next
  MsgBox "Shuma=" & S
End Sub

```

Shembull 6. Ndërtoni një program i cili vendos numrat nga 1 ne 12 në secilën prej qelizave në kolonën A (nga rreshti 1 deri në 12).

```

Sub shembull6()
  Dim num As Integer
  num = 1

```

```

While num <= 12
Cells(num, 1) = num
num = num + 1
Wend
End Sub

```

Shembull 7. Ndërtoni një program që gjen të gjithë numrat e vargut të Fibonacci në një varg natyror nga 1 deri në 1000 dhe i vendos në shtyllën e parë të sheet-it ku jeni duke punuar.

```

Sub shembull7()
Dim i As Integer
Dim iFib As Integer
Dim iFib_Next As Integer
Dim iStep As Integer
i = 1
iFib_Next = 0
Do While iFib_Next < 1000
If i = 1 Then
iStep = 1 iFib = 0
Else
iStep = iFib
iFib = iFib_Next
End If
Cells(i, 1).Value = iFib
iFib_Next = iFib + iStep
i = i + 1
Loop
End Sub

```

Shembull 8. Cikli i mëposhtëm vendos numri 5 në një qelizë, pastaj lëviz tek qeliza tjetër dhe bën të njëjtën gjë. Kjo gjë përsëritet 10 here.

```

Sub shembull8()
Dim i As Integer
i = 1
Do While i < 11
Cells(i, 1).Value = 5
i = i + 1
Loop
End Sub

```

Shembull 9. Ndërtoni një program i cili vendos numrat nga 1 në 12 në secilën prej qelizave në kolonën A (nga rreshti 1 deri në 12).

```

Sub shembull9()

```

```
Dim num As Integer
num = 1
Do While num <= 12
Cells(num, 1) = num
num = num + 1
Loop
End Sub
```

Shembull 10. (shih lab-u16) Ndërtoni një program që gjen shumën nga 1, 3, 5, 7, 9, 11, ..., 31 duke përdorur ciklin Do While.

```
Sub shembull10()
Dim i, S As Integer
i = 1
Do While i <= 31
S = S + i
i = i + 2
Loop
MsgBox "Shuma=" & S
End Sub
```

```
Sub shembull10()
Dim i, S As Integer
For i = 1 To 31 Step 2
S = S + i
Next
MsgBox "Shuma=" & S
End Sub
```