



## Il Metodo OnTime ed esempi sui colori

Esempi di codice per modificare il colore del carattere in celle aventi "formule", "formule e numeri" e "costanti".

```
Sub colore_formule()  
Dim coloreF As Long  
Dim coloreN As Long  
Dim coloreC As Long  
Dim cell As Range  
coloreF = RGB(Red:=0, Green:=255, Blue:=0)  
coloreN = RGB(Red:=0, Green:=0, Blue:=0)  
coloreC = RGB(Red:=0, Green:=0, Blue:=255)  
For Each cell In ActiveSheet.UsedRange.SpecialCells(xlCellTypeFormulas)  
'celle con formule  
cell.Font.Color = coloreF  
Next cell  
For Each cell In ActiveSheet.UsedRange.SpecialCells(xlCellTypeFormulas, xlNumbers)  
'celle con numeri  
cell.Font.Color = coloreN  
Next cell  
For Each cell In ActiveSheet.UsedRange.SpecialCells(xlCellTypeConstants)  
'celle con costanti (non formule)  
cell.Font.Color = coloreC  
Next cell  
End Sub
```

```
Sub colore_formule2()  
Dim coloreF As Long  
Dim coloreN As Long  
Dim coloreC As Long  
Dim cell As Range  
coloreF = RGB(Red:=0, Green:=255, Blue:=0)  
coloreN = RGB(Red:=0, Green:=0, Blue:=0)  
coloreC = RGB(Red:=0, Green:=0, Blue:=255)  
For Each cell In ActiveSheet.UsedRange  
If cell.HasFormula = True Then  
'celle con formule  
cell.Font.Color = coloreF  
If IsNumeric(cell) = True Then  
'celle con numeri  
cell.Font.Color = coloreN  
End If  
Else  
cell.Font.Color = coloreC  
'cellule con costanti  
End If  
Next cell  
End Sub
```



Il codice sotto riportato cambia il colore di una cella quando vengono immessi i valori delle celle sfruttando l'evento Change del foglio di lavoro

```
Private Sub Worksheet_Change(ByVal Target As Range)
Dim coloreF As Long
Dim coloreN As Long
Dim coloreC As Long
coloreF = RGB(Red:=0, Green:=255, Blue:=0)
coloreN = RGB(Red:=0, Green:=0, Blue:=0)
coloreC = RGB(Red:=0, Green:=0, Blue:=255)
With Target
If .HasFormula Then
'celle con formule
.Font.Color = coloreF
If IsNumeric(Target) Then
'celle con numeri
.Font.Color = coloreN
End If
Else
.Font.Color = coloreC
'celle con costanti (non formule)
End If
End With
End Sub
```

## Eseguire le macro a intervalli periodici o a un tempo determinato

### ■ Il Metodo **Application.OnTime**

Si può utilizzare il metodo *Application.OnTime* per eseguire una procedura a intervalli specifici o in un momento specifico della giornata. *Sintassi: ApplicationObject OnTime (earliestTime, ProcedureName, LatestTime, Schedule).*

Usando questo metodo è possibile pianificare l'esecuzione di una procedura in futuro in specifici intervalli di tempo, a partire da oggi, oppure è possibile fissare un momento specifico della giornata. L'oggetto Application rappresenta l'intera applicazione Excel, ed è l'oggetto più in alto nella gerarchia degli oggetti di Excel. Gli argomenti *earliestTime* e *ProcedureName* devono essere specificati, mentre gli altri argomenti sono facoltativi.

*earliestTime*: Questo argomento specifica il momento in cui la procedura viene eseguita

*ProcedureName*: Questo argomento specifica il nome della procedura che si desidera eseguire.

*LatestTime*: Con questo argomento è possibile impostare il limite di tempo per l'esecuzione della procedura vale a dire che se si imposta *LatestTime* a *earliestTime* + 20 e se nel frattempo un'altra procedura viene eseguita ed Excel non è "pronto" entro 20 secondi, questa procedura non verrà eseguita, tralasciando l'argomento *LatestTime* Excel eseguirà la procedura. Tralasciando l'argomento *Schedule* sarà impostato di default su True, che stabilisce una nuova procedura OnTime. Per annullare una procedura OnTime esistente impostata in precedenza, si deve specificare OnTime su False.

Per eseguire una procedura in specifici intervalli a partire da oggi, si deve utilizzare "Now + TimeValue (ora)", se per esempio si esprime il metodo con una espressione come: TimeValue ("20:30:00"), verrà



eseguita una procedura alle 20:30, mentre invece per eseguire una procedura a intervalli specifici di tempo (ad esempio, da oggi), si deve utilizzare *Now + TimeValue (ora)* vale a dire, *Ora + TimeValue ("00:00:05")* imposta l'intervallo di tempo a 5 secondi, al quale intervallo la procedura verrà eseguita.

### ■ Arrestare o annullare una procedura utilizzando il metodo OnTime

Se si tenta di chiudere la cartella di lavoro, mentre una procedura è in esecuzione utilizzando *Application.OnTime*, Excel riaprirà la cartella di lavoro per completare la procedura, quindi, sarà necessario annullare la procedura. Per annullare una procedura in esecuzione utilizzando il metodo *OnTime*, è richiesto il tempo preciso della sua esecuzione pianificata. Si noti che se non si passa il tempo a una variabile, Excel non sa quale metodo *OnTime* deve utilizzare per annullare l'operazione. Per esempio se viene usata la sintassi *Now + TimeValue ("00:00:03")*, non è un valore statico, ma lo può diventare quando viene passato a una variabile. Ciò significa che il momento in cui la procedura è da eseguire (argomento *earliestTime*) deve essere assegnato a una variabile, utilizzando una variabile pubblica per rendere la variabile a disposizione di tutte le procedure in tutti i moduli e poi utilizzarlo per annullare il metodo *OnTime*.

*Esempio:* Questa procedura utilizza il metodo *OnTime* il cui valore di incremento automatico viene reperito in una cella e viene eseguita a intervalli di tempo specifici. Il procedimento si Arresta al superamento di un valore in una cella specifica. La procedura deve essere inserita in un modulo standard

```
Public tempoR As Date
Sub incremento1()
'impone l'intervallo di tempo a 3 secondi
tempoR = Now + TimeValue("00:00:03")
'la procedura verrà eseguita in automatico nell'intervallo di tempo fissato
Application.OnTime EarliestTime:=tempoR, Procedure:="incremento1", schedule:=True
'incrementare il valore nella cella A1 di 5 ogni volta che si esegue la macro
Cells(1, 1).Value = Cells(1, 1).Value + 5
'la procedura si ferma quando trova il valore 25 nella cella A1
If Cells(1, 1).Value > 25 Then
'annullare la procedura impostando l'argomento schedule a False
Application.OnTime tempoR, "incremento1", , False
End If
End Sub
```

*Esempio:* Questa procedura utilizza il metodo *OnTime* il cui valore di incremento automatico viene reperito in una cella e viene eseguita a intervalli di tempo specifici. Il procedimento si Arresta dopo essere stata eseguita un determinato numero di volte. La procedura deve essere inserita in un modulo standard

```
Public tempoI As Date
Dim conta As Integer
Sub incremento2()
tempoI = Now + TimeValue("00:00:03")
Application.OnTime tempoI, "incremento2", , True
Cells(1, 1).Value = Cells(1, 1).Value + 5
conta = conta + 1
'Interrompere la procedura dopo averla eseguita per 5 volte
If conta = 5 Then
Application.OnTime tempoI, "incremento2", , False
conta = 0
End If
End Sub
```



*Esempio:* Avviare la procedura OnTime automaticamente quando la cartella di lavoro viene aperta e interromperla automaticamente alla chiusura della cartella di lavoro. Questa procedura imposta i promemoria in orari specifici e chiude automaticamente la cartella di lavoro in un momento specifico. Aggiungere il codice sotto riportato al modulo [ThisWorkbook](#)

```
Private Sub Workbook_Open()  
ricordami  
End Sub  
  
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
On Error Resume Next  
'fermo il promemoria  
ricordamiST  
'salva cartella di lavoro prima della chiusura  
ThisWorkbook.Save  
End Sub
```

Le procedure sotto riportate devono essere inserite in un modulo standard:

```
Public dTime As Date  
Sub ricordami()  
On Error Resume Next  
dTime = Now + TimeValue("00:00:01")  
Application.OnTime dTime, "ricordami"  
'Chiudere la cartella di lavoro nel momento specificato  
  
If Time = TimeSerial(18, 30, 0) Then  
chiudiCAR  
End If  
  
'Impostare promemoria di chiusura ufficio  
If Time = TimeSerial(17, 30, 0) Then  
Application.OnTime dTime, "chiudiOF"  
End If  
  
'impostare promemoria pausa pranzo  
If Time = TimeSerial(13, 0, 0) Then  
Application.OnTime dTime, "pranzo"  
End If  
  
'impostare il promemoria per la pausa caffè  
If Time = TimeSerial(11, 15, 0) Then  
Application.OnTime dTime, "Pcaffe"  
End If  
End Sub
```



```
Sub ricordamiST()  
    'fermare la procedura ricordami  
    Application.OnTime dTime, "ricordami", , False  
End Sub  
  
Sub chiudiCAR()  
    On Error Resume Next  
    ricordamiST  
    'salva cartella di lavoro prima della chiusura  
    ThisWorkbook.Save  
    'chiude la cartella di lavoro  
    ThisWorkbook.Close  
End Sub  
  
Sub Pcaffe()  
    Dim obj As Object  
    Dim strMsg As String  
    Set obj = CreateObject("WScript.Shell")  
    'riproduco un segnale acustico di avviso  
    Beep  
    'Popup Message Box si chiuderà automaticamente da solo  
    strMsg = obj.Popup("Pausa Caffè!", vbOKCancel)  
End Sub  
  
Sub pranzo()  
    Dim obj As Object  
    Dim strMsg As String  
    Set obj = CreateObject("WScript.Shell")  
    Beep  
    strMsg = obj.Popup("Pausa Pranzo!", vbOKCancel)  
End Sub  
  
Sub chiudiOF()  
    Dim obj As Object  
    Dim strMsg As String  
    Set obj = CreateObject("WScript.Shell")  
    Beep  
    strMsg = obj.Popup("Chiudi Ufficio!", vbOKCancel)  
End Sub
```