

Option Explicit

```
Public Type tParametres
    HeuresHebdo As Double
    JoursSemaine As Long
    DureeJour As Double
    DureeEcole As Double
End Type
```

```
Public Params As tParametres
```

```
Public Sub ChargerParametres()
```

```
    Dim wsP As Worksheet
```

```
    Dim wsPar As Worksheet
```

```
    Set wsP = ThisWorkbook.Worksheets("Planning")
```

```
    Set wsPar = ThisWorkbook.Worksheets("Parametres")
```

```
    ' Lecture sécurisée : 0 ne pose plus aucun problème
```

```
    Params.HeuresHebdo = Val(wsP.Range("B1").Value)
```

```
    Params.JoursSemaine = Val(wsP.Range("B2").Value)
```

```
    Params.DureeEcole = Val(wsPar.Range("B5").Value)
```

```
    ' Calcul automatique
```

```
    If Params.JoursSemaine > 0 Then
```

```
        Params.DureeJour = Params.HeuresHebdo / Params.JoursSemaine
```

```
    Else
```

```
        Params.DureeJour = 0
```

```
    End If
```

```
    ' Écriture sécurisée dans Parametres!B4
```

```
    With wsPar
```

```
        .Unprotect Password:=PWD
```

```
        .Range("B4").Value = Params.DureeJour
```

```
        .Protect Password:=PWD, UserInterfaceOnly:=True
```

```
    End With
```

```
End Sub
```

## Option Explicit

```

Public Sub GenererPlanning()
    Dim ws As Worksheet, wsParam As Worksheet
    Dim dt As Date
    Dim baseRow As Long, semaineRow As Long
    Dim i As Long, R As Long
    Dim lastRow As Long
    Dim Annee As Long
    Dim reponse As Variant
    Dim h As Long, ligne As Long
    Dim rngHeures As Range

    Application.ScreenUpdating = False
    Application.EnableEvents = False

    ' Charger les paramètres (même si B1 ou B2 = 0)
    Call ChargerParametres

    ' Choix année
    reponse = InputBox("Quelle année veux-tu générer ?", "Planning", Year(Date))
    If Not IsNumeric(reponse) Then GoTo Fin
    Annee = CLng(reponse)

    ' Feuilles
    Set ws = ThisWorkbook.Worksheets("Planning")
    Set wsParam = ThisWorkbook.Worksheets("Parametres")

    ' RESET COMPLET DE LA FEUILLE
    With ws
        .Unprotect PWD
        .Cells.Clear
        .Cells.Validation.Delete
        .Cells.Borders.LineStyle = xlNone
        .Cells.Interior.ColorIndex = xlNone
        .Cells.Font.ColorIndex = xlAutomatic
        .Cells.NumberFormat = "General"
        .Rows.Hidden = False
        .Columns.Hidden = False
        .Cells.Locked = False
    End With

    ' EN-TÊTES
    ws.Range("A1").Value = "Planning " & Annee

    ws.Range("B1").Value = "Contrat hebdo (h)"
    ws.Range("C1").Value = Params.HeuresHebdo
    ws.Range("B2").Value = "Jours travaillés"
    ws.Range("C2").Value = Params.JoursSemaine

    ws.Range("A4:L4").Value = Array( _
        "Date", "Arrivée", "Début pause", "Fin pause", _
        "Début pause 2", "Fin pause 2", "Départ", _
        "Total jour", "Commentaire", "Heures payées", _
        "Repas matin", "Repas soir")

    With ws.Range("A4:L4")
        .Font.Bold = True
        .Interior.Color = RGB(220, 220, 220)
    End With

    ' Premier lundi
    dt = DateSerial(Annee, 1, 1)
    Do While Weekday(dt, vbMonday) <> 1
        dt = dt + 1
    Loop

    baseRow = 6

    ' GÉNÉRATION DES SEMAINES
    Do While dt <= DateSerial(Annee, 12, 31)

        semaineRow = baseRow

        ws.Range("A" & semaineRow).Value = _
            "Semaine du " & Format(dt, "dd/mm/yyyy") & _
            " au " & Format(dt + 6, "dd/mm/yyyy")
    
```

```

With ws.Range("A" & semaineRow & ":L" & semaineRow)
    .Font.Bold = True
    .Interior.Color = RGB(235, 235, 235)
End With

For i = 1 To 7
    R = semaineRow + i

    ' Date
    ws.Cells(R, "A").Value = dt + (i - 1)
    ws.Cells(R, "A").NumberFormat = "dddd dd mmm"

    ' Total jour (H)
    ws.Cells(R, "H").FormulaLocal = _
        "=SI(OU(B" & R & "="""";G" & R & "=""")&"";" & _
        "G" & R & "-B" & R & "-(D" & R & "-C" & R & ")&-" & _
        "SI(OU(E" & R & "="""";F" & R & "=""")&0;F" & R & "-E" & R & ")&)"

    ' Heures payées (J)
    ws.Cells(R, "J").FormulaLocal = _
        "=SI(I" & R & "="""";H" & R & "&";" & _
        "SI(I" & R & "=""Ecole"";Parametres!B5/24;" & _
        "SI(OU(" & _
        "I" & R & "=""RTT"";" & _
        "I" & R & "=""Congés payés"";" & _
        "I" & R & "=""Arrêt de travail"";" & _
        "I" & R & "=""Conges sans solde"";" & _
        "I" & R & "=""Jour de repos"";" & _
        "I" & R & "=""Formation"";" & _
        "I" & R & "=""Férié"";" & _
        "Parametres!B4/24;H" & R & ")&))"

    ' Formats
    ws.Cells(R, "B").Resize(1, 6).NumberFormat = "hh:mm"
    ws.Cells(R, "H").NumberFormat = "[h]:mm"
    ws.Cells(R, "J").NumberFormat = "[h]:mm"
Next i

' Totaux semaine
ws.Range("A" & (semaineRow + 8)).Value = "Total heures semaine"
ws.Range("J" & (semaineRow + 8)).FormulaLocal = _
    "=SOMME(J" & (semaineRow + 1) & ":J" & (semaineRow + 7) & ")"

ws.Range("A" & (semaineRow + 9)).Value = "Écart vs contrat"
ws.Range("J" & (semaineRow + 9)).FormulaLocal = _
    "=J" & (semaineRow + 8) & "*24-Planning!C1"

ws.Range("A" & (semaineRow + 10)).Value = "Total repas semaine"
ws.Range("K" & (semaineRow + 10)).FormulaLocal = _
    "=SOMME(K" & (semaineRow + 1) & ":K" & (semaineRow + 7) & ")"
ws.Range("L" & (semaineRow + 10)).FormulaLocal = _
    "=SOMME(L" & (semaineRow + 1) & ":L" & (semaineRow + 7) & ")"

baseRow = baseRow + 11
dt = dt + 7
Loop

lastRow = baseRow - 1

' LISTES DÉROULANTES
' Heures en Parametres!Z
ligne = 1
For h = 0 To 23
    wsParam.Cells(ligne, "Z").Value = Format(TimeSerial(h, 0, 0), "hh:mm")
    wsParam.Cells(ligne + 1, "Z").Value = Format(TimeSerial(h, 30, 0), "hh:mm")
    ligne = ligne + 2
Next h

Set rngHeures = wsParam.Range("Z1:Z" & ligne - 1)

With ws.Range("B6:G" & lastRow).Validation
    .Delete
    .Add Type:=xlValidateList, Formula1:="" & rngHeures.Address(True, True, xlA1, True)
End With

' Commentaires

```

```
With ws.Range("I6:I" & lastRow).Validation
    .Delete
    .Add Type:=xlValidateList, Formula1:="=Parametres!$A$11:$A$19"
End With
```

```
' Repas (0 ou 1)
With ws.Range("K6:L" & lastRow).Validation
    .Delete
    .Add Type:=xlValidateList, Formula1:="0;1"
End With
```

```
Fin:
Application.EnableEvents = True
Application.ScreenUpdating = True
MsgBox "Planning " & Annee & " généré avec succès.", vbInformation
End Sub
```

## Option Explicit

```
Private Sub Worksheet_Change(ByVal Target As Range)
    On Error GoTo Fin

    If Application.EnableEvents = False Then Exit Sub
    Application.EnableEvents = False

    ' Coloration uniquement colonne I à partir de la ligne 6
    If Not Intersect(Target, Me.Range("I6:I1048576")) Is Nothing Then
        Dim wsParam As Worksheet
        Dim lastRow As Long, i As Long
        Dim com As String, comClean As String
        Dim parts() As String
        Dim R As Long, G As Long, B As Long

        Set wsParam = ThisWorkbook.Worksheets("Parametres")

        com = CStr(Target.Value)
        comClean = LCase(Trim(com))

        ' Si vide ? ligne blanche
        If comClean = "" Then
            Me.Range("A" & Target.Row & ":L" & Target.Row).Interior.Color = RGB(255, 255, 255)
            GoTo Fin
        End If

        lastRow = wsParam.Cells(wsParam.Rows.Count, "A").End(xlUp).Row

        For i = 11 To lastRow
            If LCase(Trim(wsParam.Cells(i, "A").Value)) = comClean Then
                parts = Split(wsParam.Cells(i, "B").Value, ",")
                If UBound(parts) = 2 Then
                    R = CLng(Trim(parts(0)))
                    G = CLng(Trim(parts(1)))
                    B = CLng(Trim(parts(2)))

                    Me.Range("A" & Target.Row & ":L" & Target.Row).Interior.Color = RGB(R, G, B)
                End If
            End If
        Next i
    End If

Fin:
    Application.EnableEvents = True
End Sub
```

Option Explicit

```
Public Const PWD As String = "240194"
```

```
Public Sub ToggleProtection()
```

```
    Dim ws As Worksheet  
    Dim wsP As Worksheet  
    Dim lastRow As Long  
    Dim code As String
```

```
    code = InputBox("Entrez le code de sécurité :", "Sécurité")
```

```
    If code <> PWD Then
```

```
        MsgBox "Code incorrect.", vbCritical
```

```
        Exit Sub
```

```
    End If
```

```
    ' Déjà protégé ?
```

```
    If ThisWorkbook.ProtectStructure = True Then
```

```
        ThisWorkbook.Unprotect Password:=PWD
```

```
        For Each ws In ThisWorkbook.Worksheets
```

```
            ws.Unprotect Password:=PWD
```

```
        Next ws
```

```
        MsgBox "Classeur et feuilles déprotégés.", vbInformation
```

```
        Exit Sub
```

```
    End If
```

```
    ' Sinon : protéger
```

```
    For Each ws In ThisWorkbook.Worksheets
```

```
        ws.Cells.Locked = True
```

```
    Next ws
```

```
    Set wsP = ThisWorkbook.Worksheets("Planning")
```

```
    lastRow = wsP.Cells(wsP.Rows.Count, "A").End(xlUp).Row
```

```
    ' Zones saisissables
```

```
    wsP.Range("B6:G" & lastRow).Locked = False
```

```
    wsP.Range("I6:I" & lastRow).Locked = False
```

```
    wsP.Range("K6:L" & lastRow).Locked = False
```

```
    ThisWorkbook.Protect Password:=PWD, Structure:=True
```

```
    For Each ws In ThisWorkbook.Worksheets
```

```
        ws.Protect Password:=PWD, UserInterfaceOnly:=True
```

```
    Next ws
```

```
    MsgBox "Classeur et feuilles protégés.", vbInformation
```

```
End Sub
```

## Option Explicit

```
Public Sub MettreAJourRecap()
```

```
    Dim wsP As Worksheet, wsR As Worksheet
    Dim lastRow As Long, i As Long
    Dim mois As Integer, ligne As Long
    Dim joursTrav As Long
    Dim repasMatin As Long, repasSoir As Long
    Dim heuresMois As Double
    Dim dictCom As Object
    Dim com As String
    Dim colCom As Long
    Dim heuresContratMois As Double
    Dim key As Variant
```

```
    Set wsP = ThisWorkbook.Worksheets("Planning")
    Set wsR = ThisWorkbook.Worksheets("Recap")
```

```
    ' S'assurer que Params est chargé
    Call ChargerParametres
```

```
    wsR.Cells.Clear
```

```
    wsR.Range("A1:G1").Value = Array("Mois", "Jours travaillés", "Repas pris", "Différence", "Heures sup", "Signature", "RTT")
    wsR.Range("A1:G1").Font.Bold = True
```

```
    ligne = 2
    lastRow = wsP.Cells(wsP.Rows.Count, "A").End(xlUp).Row
```

```
    For mois = 1 To 12
        Set dictCom = CreateObject("Scripting.Dictionary")
        joursTrav = 0
        repasMatin = 0
        repasSoir = 0
        heuresMois = 0
```

```
    For i = 6 To lastRow
        If IsDate(wsP.Cells(i, "A").Value) Then
            If Month(wsP.Cells(i, "A").Value) = mois Then
                ' Jours travaillés (sans commentaire)
                If wsP.Cells(i, "H").Value > 0 And wsP.Cells(i, "I").Value = "" Then
                    joursTrav = joursTrav + 1
                End If

                ' Repas
                repasMatin = repasMatin + Val(wsP.Cells(i, "K").Value)
                repasSoir = repasSoir + Val(wsP.Cells(i, "L").Value)

                ' Heures payées (en heures)
                heuresMois = heuresMois + (wsP.Cells(i, "J").Value * 24)

                ' Commentaires
                com = wsP.Cells(i, "I").Value
                If com <> "" Then
                    If Not dictCom.Exists(com) Then dictCom.Add com, 0
                    dictCom(com) = dictCom(com) + 1
                End If
            End If
        End If
    Next i
```

```
    ' Écriture recap
    wsR.Cells(ligne, "A").Value = Format(DateSerial(Year(Date), mois, 1), "mmmm")
    wsR.Cells(ligne, "B").Value = joursTrav
    wsR.Cells(ligne, "C").Value = repasMatin + repasSoir
```

```
    ' Différence vs contrat (heures)
    heuresContratMois = Params.HeuresHebdo * 4.33
    wsR.Cells(ligne, "D").Value = heuresMois - heuresContratMois
```

```
    ' Heures sup
    If heuresMois > heuresContratMois Then
        wsR.Cells(ligne, "E").Value = heuresMois - heuresContratMois
    Else
        wsR.Cells(ligne, "E").Value = 0
    End If
```

```
wsR.Cells(ligne, "E").NumberFormat = "[h]""h""mm""min""

' Signature
wsR.Cells(ligne, "F").Value = ""

' Colonnes dynamiques pour les commentaires
colCom = 8
For Each key In dictCom.Keys
    wsR.Cells(1, colCom).Value = key
    wsR.Cells(1, colCom).Font.Bold = True
    wsR.Cells(ligne, colCom).Value = dictCom(key)
    colCom = colCom + 1
Next key

    ligne = ligne + 1
Next mois

wsR.Columns.AutoFit
MsgBox "Récap mensuel mis à jour.", vbInformation
End Sub
```