Calcul de la date de Pâques julienne[

Calcul de la date de Pâques *julienne* (326-) en calendrier *julien* (Algorithme de Delambre) [8](https://fr.wikipedia.org/wiki/Calcul_de_la_date_de_P%C3%A2ques_selon_la_m%C3%A9thode_de_Meeus#cite_note-11)

*Si* Année ≥ 326 [Note 4](https://fr.wikipedia.org/wiki/Calcul_de_la_date_de_P%C3%A2ques_selon_la_m%C3%A9thode_de_Meeus#cite_note-12) *alors* :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date de Pâques julienne (algorithme de Delambre)** | | | | |
| **Dividende** | **Diviseur** | **Quotient** | **Reste** | **Expression** |
| *Année* | 19 |  | *A* |  |
| *Année* | 7 |  | *B* |  |
| *Année* | 4 |  | *C* |  |
| 19 *A* + 15 | 30 |  | *D* |  |
| 2 *C* + 4 *B* - *D* + 34 | 7 |  | *E* |  |
| D + E + 114 | 31 | F | G |  |

*F* est le mois de Pâques (3 = mars, 4 = avril) ;

*G + 1* est le quantième, dans le mois ci-dessus, du dimanche de Pâques.

*Exemple pour l'année****1492***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date de Pâques julienne en calendrier julien pour 1492 (algorithme de Delambre)** | | | | | | | |
| **Dividende** | **Valeur Dividende** | **Diviseur** | **Valeur Diviseur** | **Quotient** | **Valeur Quotient** | **Reste** | **Valeur Reste** |
| *Année* | 1492 | 19 | 19 |  |  | *A* | 10 |
| *Année* | 1492 | 7 | 7 |  |  | *B* | 1 |
| *Année* | 1492 | 4 | 4 |  |  | *C* | 0 |
| 19 *A* + 15 | 205 | 30 | 30 |  |  | *D* | 25 |
| 2 *C* + 4 *B* - *D* + 34 | 13 | 7 | 7 |  |  | *E* | 6 |
| *D* + *E* + 114 | 145 | 31 | 31 | *F* | 4 | *G* | 21 |

*F* = 4

Donc : Mois = avril

G = 21 ; G + 1 =22

Donc : Quantième = 22

**Pâques est le 22 avril 1492**.

# Algo ajout des données dans la table « tblCalendrier »

* Récupérer la date du jour :
  + - Fonction « Date »
    - DtDate =Date
* Ouvrir un recordset de type Dynaset sur « tblCalendrier »
  + - Set oDb = CurrentDb
    - Set oRstCalendrier = oDb.OpenRecordset(‘’tblCalendrier’’,dbOpenTable)
* Passer en mode Ajout sur le recordset
  + - oRstCalendrier.AddNew
* Ajouter 360 lignes à la table « tblCalendrier » … les lignes seront constituées du jour courant au 360 -ème jour en partant du jour courant :
  + - For intI = 1 To 360 Step 1
      * If intI = 1 then
        + oRstCalendrier.Fields(‘’idDate ‘’).value =DtDate
      * Else
        + DtDate = DtDate +1
        + oRstCalendrier.Fields(‘’idDate ‘’).value =DtDate
      * Select case weekDay(DtDate)
        + Case vbMonday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Lundi’’

* + - * + Case vbTuesday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Mardi’’

* + - * + Case vbWednesday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Mercredi’’

* + - * + Case vbThursday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Jeudi’’

* + - * + Case vbFriday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Vendredi’’

* + - * + Case vbSaturday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Samedi’’

* + - * + Case vbSunday

oRstCalendrier.Fields(‘’idDay ‘’).value = ‘’Dimanche’’

* + - * Select case Month(DtDate)
        + Case 1

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Janvier’’

* + - * + Case 2

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Février’’

* + - * + Case 3

oRstCalendrier.Fields(‘’idMonth‘’).value =‘’Mars’’

* + - * + Case 4

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Avril’’

* + - * + Case 5

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Mai’’

* + - * + Case 6

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Juin’’

* + - * + Case 7

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Juillet’’

* + - * + Case 8

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Août’’

* + - * + Case 9

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Septembre’’

* + - * + Case 10

oRstCalendrier.Fields(‘‘idMonth‘’).value = ‘’Octobre’’

* + - * + Case 11

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Novembre’’

* + - * + Case 12

oRstCalendrier.Fields(‘’idMonth‘’).value = ‘’Décembre’’

* + - * oRstCalendrier.Fields(‘’idYear‘’).value = Year(DtDate)
      * Mettre à jour les données
        + oRstCalendrier.Update

# Attribution NotWorkDay Weekend« tblCalendrier »

* Après l’exécution de l’algorithme 1 exécuter cet algo pour attribuer les cases WorkDay et AlgoDay en fonction des weekends :
* Dans un premier temps créer un script sql « strQryWeekend » qui va sélectionner toutes les lignes de la table « tblCalendrier » :
  + - * strQryWeekend = ‘’SELECT tblCalendrier.\* & vbCrLf & \_

’’From tblCalendrier & vbCrLf & \_

‘’Where tblCalendrier.idDay = ‘’Samedi ‘’ or tblCalendrier.idDay = ‘’Dimanche‘’ ‘’

* Ouvrir un recordset de type Dynaset basé sur la requête « strQryWeekend »
  + - * Set oDb = CurrentdB
      * Set oRstCalendrier = oDb.openRecordset( strQryWeekend , dbOpenDynaset)
* On vérifie ensuite l’existence d’enregistrement contenu dans le recordset pour attribuer la valeur true à NotWorkDay et AlgoDay pour les weekends
  + - * If Not oRstCalendrier.EOF Then
        + While not oRstCalendrier.EOF

oRstCalendrier.Edit

oRstCalendrier.Fields(« NotWorkDay »). Value = True

oRstCalendrier.Fields(« AlgoDay »). Value = True

oRstCalendrier.Update

oRstCalendrier.MoveNext

* Wend
  + - * Else
        + Msgbox ‘’ il n’y a pas d’enregistrement ‘’
      * End if
* Libération des données
  + - * oRstCalendrier.Close
      * oDb.Close
      * Set oRstCalendrier = Nothing
      * Set oDb = Nothing

# Attribution NotWorkDay Férié« tblCalendrier »

* Après l’exécution de l’algorithme 2 exécuter cet algo pour attribuer les cases WorkDay et AlgoDay en fonction des jours Fériés:
* Dans un premier temps créer un script sql « strQryFerie» qui va sélectionner toutes les lignes de la table « tblCalendrier » :
  + - * strQryFerie = ‘’SELECT tblCalendrier.\* & vbCrLf & \_

’’From tblCalendrier & vbCrLf & \_

* Ouvrir un recordset de type Dynaset basé sur la requête « strQryFerie»
  + - * Set oDb = CurrentdB
      * Set oRstCalendrier = oDb.openRecordset( strQryFerie, dbOpenDynaset)
* On vérifie ensuite l’existence d’enregistrement contenu dans le recordset pour attribuer la valeur true à NotWorkDay et AlgoDay pour les weekends
  + - * If Not oRstCalendrier.EOF Then
        + While not oRstCalendrier.EOF

oRstCalendrier.Edit

if EstFerie(oRstCalendrier.(« idDate »).value) Then

oRstCalendrier.Fields(« NotWorkDay »). Value = True

oRstCalendrier.Fields(« AlgoDay »). Value = True

oRstCalendrier.Update

* End if

oRstCalendrier.MoveNext

* Wend
  + - * Else
        + Msgbox ‘’ il n’y a pas d’enregistrement ‘’
      * End if
* Libération des données
  + - * oRstCalendrier.Close
      * oDb.Close
      * Set oRstCalendrier = Nothing
      * Set oDb = Nothing