

VBA

□

Sisällys

- [1 Johdanto](#)
- [2 Käyttöönotto](#)
- [3 Tallentaminen](#)
- [4 Virheenetsintä](#)
- [5 Objekti](#)
- [6 Excel](#)
 - [6.1 Oma objekti \(Dim\)](#)
 - [6.1.1 Muuttuja](#)
 - [6.2 Funktio \(Sub\)](#)
 - [6.3 Jos \(If\)](#)
 - [6.4 Silmukka \(For..Next\)](#)
 - [6.5 With](#)
 - [6.6 Painike](#)
 - [6.7 Haku](#)
 - [6.7.1 Google-haku, Chrome-selain](#)
 - [6.8 Kopiointi](#)
 - [6.9 Noutaminen](#)
 - [6.10 Numerot kirjaimiksi](#)
 - [6.11 PDF](#)
 - [6.12 Ponnahdusikkuna](#)
 - [6.13 Pudotusvalikko](#)
 - [6.14 Rivi](#)
 - [6.15 Solun arvon muuttaminen automaattisesti muutettaessa toista solua](#)
 - [6.16 Solun tyjentäminen](#)
 - [6.17 Solun värjääminen](#)
 - [6.18 URL-osoite linkistä](#)
 - [6.19 Vie solu lainausmerkeissä tekstitiedostoon](#)
 - [6.20 JSON -tuonti Exceliin](#)
 - [6.20.1 JSON-esimerkki 1, YTJ](#)
- [7 Lähteitä](#)

Johdanto

Visual Basic for Applications (VBA)

Katso lisää funktioita sivulta [Microsoft 365](#)

[Office VBA Reference](#)

Käyttöönotto

1. Tiedosto > Asetukset > Valintanauhan mukauttaminen > valitse Kehitystyökalut > OK
2. Kehitystyökalut-valikko ilmestyy näkyville ikkunan yläosaan.
3. Visual Basic-painike avaa uuden makrojen muokkausikkunan.
4. Insert > Module
5. Lisää makron koodi ruutuun
6. Aja makro play-painikkeesta tai tee makrolle sitä kutsuva painike välilehdelle.

TAI Jos välilehdistä ei löydy kohtaa 'Kehitystyökalut': Klikkaa ikonitoimintorivillä hiiren oikeaa > Mukauta valintanauhaa... > Valitse Kehitystyökalut

Tallentaminen

Tiedostopääte on m-loppuinen.

Virheenetsintä

Paina F1 kun virhekohta on valittuna tai joku muu kohta on valittuna.

Objekti

Wordissa tiedosto-objekti (document), joka sisältää kappale-objekteja (paragraph). Excelissä esim. väilehti. Objektin määritelmää kutsutaan luokaksi (class), jolla objekti luodaan ja ilmennetään. Objektia voidaan muokata asettamalla sen ominaisuuksia (property) ja kutsumalla sitä metodeilla (method). Metodi saa objektin toimimaan jollain tavoin.

- Application.ActiveDocument.Save - tallentaa aktiivisena olevan tiedoston. Document on objekti. Save on metodi.
- Application.ActiveDocument.SaveAs ("tiedostonimi.docx") - SaveAs-metodi edellyttää parametrin (parameter), joka annetaan metodin jälkeen.
- Application.ActiveSheet.Range("A1").Select ja Application.Selection.Value = "Teksti" - Ominaisuuden asettaminen ja lukeminen tapahtuu samalla tavalla. Application on objekti, ActiveSheet on objekti. Select on metodi. Selectionin value-ominaisuuteen tallennetaan teksti, joka ilmenee A1-solussa.

Excel

- &: Jos VBA-koodissa netissä on esim. & ja tulee virheilmoitus ajettaessa makroa. Poista amp; ja jätä jäljelle &

Oma objekti (Dim)

Kutsutaan muuttujaksi (variable). Objektin tyyppi määritellään ensin Dim-statementilla. Sen jälkeen objektille voidaan määritellä arvot ja käyttää muutoin.

```
Dim OmaMerkkijonoMuuttuja As String
OmaMerkkijonoMuuttuja = "tekstiä"
Worksheets(1).Range("A1").Value = OmaMerkkijonoMuuttuja
```

```
Dim Muuttuja As Worksheet
Dim Muuttuja1 As Worksheet, Muuttuja2 As Long
Dim Muuttuja As Variant
Dim Muuttuja As String
Dim Muuttuja As Range
Dim Muuttuja As Integer
Dim Muuttuja As Long
jne.
```

Muuttuja

- Huomaa muuttujan käyttötarkoitus ja sen mukaan määrittele object tai string (ks. otsikko Dim) ja käytä set tai let.
- Jos käytössä on useita välilehtiä, kannattaa välilehdet nimetä esim. seuraavasti:

```
Dim Src As Worksheet <--Src voi olla muukin sana
Dim Trg As Worksheet
Set Src = ActiveWorkbook.Worksheets("välilehdennimi1")
Set Trg = ActiveWorkbook.Worksheets("välilehdennimi2")
```

- Alueen voi nimetä esim. seuraavasti:

```
Dim aluenimi As Range
Set aluenimi = Src.Range("C$2:G$100") 'Huomaa, tässä on käytetty yllä
esitettyä Src, joka ilmaisee mistä välilehdestä on kyse.
```

- Välilehden nimi ja solu muuttujassa, objekti, set

```
Dim strCell As Object
Set strCell = Worksheets("Välilehdennimi").Range("A10")
strCell.Value = "Teksti" 'Asetetaan solulle arvo
strLoadingCell = "" 'Tyhjennetään solun arvo
```

- Välilehden nimi ja solu muuttujassa, string, let

```
Dim strCell As String
Let strCell = Worksheets("Välilehdennimi").Range("A10")
strURL = "http://osoite/" & strCell
```

- Solu muuttujassa, string

```
Dim strCell As String
Let strCell = "A1"
strURL = "https://osoite/" & Sheets("Välilehdennimi").Range(strCell)
```

- Välilehden nimi ja solu muuttujassa, ???, ???
 - Puuttuu ohje miten määritellään kun halutaisiin korvata seuraava muuttujalla:
Sheets("Välilehdennimi").Cells.ClearContents

Funktio (Sub)

Suomenkielisessä Excelissä on suomenkieliset funktioiden nimet. Googlen ohjeet viittaavat esim. englannin kielisiin funktionimiin. Käännöstalulukoita:

<http://excelohjeet.com/excel-funktiot-suomeksi-ja-englanniksi>
<https://www.excel-function-translation.com/index.php?page=english-finnish.html>
<http://www.tekstiviestit.fi/sihteeriopas/sihteeriopas-materiaali-funktiot.html>

- Funktion käyttö

```
Public Function Funktionnimi(tuotavamuuttuja1, tuotavamuuttuja2)
    tuotuja muuttujia voi käyttää samoilla nimillä eikä niitä tarvitse
    uudestaan Dim...
End Function
```

```
Sub Nimi1()
    jotain...
End Sub
```

```
Sub Nimi2()
    Funktionnimi <-- kutsutaan suoraan funktion nimellä
    Funktionnimi tuotavamuuttuja1, tuotavamuuttuja2 <-- tai lähetetään funktiolle
    muuttujissa tietoa
    Nimi1 <-- kutsutaan suoraan Subia
End Sub
```

Jos (If)

- And ja useamman Or-operaattorin yhdistäminen. Toisen vaihtoehdon useampi vaihtoehto niputetaan sulkeilla yhdeksi eli X And (Y1 Or Y2 Or Y3) Esimerkissä Src on tällä sivulla käytetty viittaus, joka kertoo mistä Excelin välilehdestä on kyse.

```
=If Src.Cells(i, "C") = muuttuja1 And (Src.Cells(i, "D") = muuttuja2 Or
Src.Cells(i, "E") = muuttuja2) Then
```

- Esimerkki 1

```
If muuttuja1 = "" Then
    Exit Sub
ElseIf muuttuja2 = "" And muuttuja3 <> "" Then
    jotain...
End If
```

- Esimerkkinä testataan onko välilehdellä 1 oleva vuosiluku alle kolme vuotta nykyisestä vuodesta ja jos on, haetaan välilehdeltä 3 arvo solusta B1 välilehdelle 2. Ja jos yli 3 vuotta, haetaan välilehdeltä arvo solusta B2.

```
If (Year(Date) - Worksheets("Välilehdennimi1").Range("A1")) > 3 Then
    Worksheets("Välilehdennimi2").Range("A2") =
Worksheets("Välilehdennimi3").Range("B1")
    Else: Worksheets("Välilehdennimi2").Range("A2") =
Worksheets("Välilehdennimi3").Range("B2")
End If
```

- Select, ponnahdusikkuna Kyllä, Ei, Peruuta

```
Select Case MsgBox("Lisätään valmis teksti, mutta tyhjennetäänkö samalla
solu A1?", vbYesNoCancel)
```

```

Case vbCancel
    Cancel = True 'peruutetaan toimenpide
Case vbNo
    Worksheets("Välilehdennimi1").Range("A2").Value =
Worksheets("Välilehdennimi2").Range("B10")
Case vbYes
    Worksheets("Välilehdennimi1").Range("A1").ClearContents
    Worksheets("Välilehdennimi1").Range("A2").Value =
Worksheets("Välilehdennimi2").Range("B10")
End Select

```

Silmukka (For..Next)

```

Dim i As Long
For i = 2 To 10
    jotain...
Next i

Sub Makro1()
    If Worksheets(1).Range("A1").Value = "valittu" Then
        Dim i As Integer
        For i = 2 To 10
            Worksheets(1).Range("A" & i).Value = "Teksti" & i
        Next i
    Else
        MsgBox "Aseta teksti valittu soluun A1"
    End If
End Sub

```

With

- Käyttötarkoitus 1: Arvojen syöttäminen useisiin saman objektin ominaisuuksiin.

```

With JokuObjekti
    .Koko = 1
    .Korkeus =2
End With

```

Painike

Usein painikkeita käytetään yhdessä makron kanssa, joka ohjelmoidaan Visual Basicilla (VBA). Painettaessa, tapahtuu jokin toiminto.

- Painikkeita lisätään a) ohjausobjektipainikkeena tai b) ActiveX ohjausobjektina.
- Peruskuvio on, että lisätään painike ja liitetään siihen makro.

- ActiveX ohjausobjekti (komentopainike) Windows Wordissa

1. Kehitystyökalut > Ohjausobjektit-ryhmässä ActiveX-komponentit valikosta komentopainike.
 1. Suunnittelutila Ohjausobjektit-ryhmässä on päällä, jolloin painiketta voidaan muokata. Kun se on poissa, painiketta voidaan käyttää.
 2. Valitse Ominaisuudet painikkeen päällä hiiren oikealla.

1. Caption kohdassa aseta painikkeessa näkyvä nimi.
2. Hiirellä värjätyn tekstin vieminen VBA-muuttujaan: valitse ylävalikosta painikkeen nimi, esim. CommandButton1 ja aseta TakeFocusOnClick arvoksi False. Tällöin sivulla oleva teksti voidaan värjätä ja kun painetaan painiketta, teksti siirtyy muuttujaan joka on ohjelmoitu VBA:ssa.
3. Valitse Näytä koodi painikkeen päällä hiiren oikealla.
 1. Lisää VBA-koodi
4. Ota suunnittelutila pois päältä ja painike toimii.

- Ohjausobjektipainike, lomakeohjaus ohjausobjekti (painike): Tämän voi valita useissa tapauksissa. Tätä ei välttämättä löydy Wordista.

1. Kehitystyökalut > Lisää > Lomakeohjausobjektit > Painike
2. Hiiren nuoli muuttuu ristikoksi, vedä painike haluamaasi kohtaan. Myöhemmin makroa voi siirrellä vapaasti ja muuttaa tekstiä.
3. Avautuu makro-ikkuna > Valitse olemassa oleva jo aiemmin tehty makro ja paina 'OK' TAI kirjoita uusi nimi makrolle ja paina 'Uusi', jolloin avautuu Microsoft Visual Basic for Applications-ikkuna.
 1. Makrot kannattaa heti alusta alkaen nimetä kuvaavasti ja johdonmukaisesti.
4. Makroja voi kirjoittaa useita samaan moduuliin, vaikka aina aukeaisi uusi moduuli. <-- tarkenna
 1. Makro alkaa: Sub Makronnimi()
 2. Makro päättyy: End Sub
 3. Edellisten väliin tulee makron suorittama koodi, esim.
 1. Range("A1") = Worksheets("Välilehdennimi").Range("B1") hakee välilehdeltä 'Välilehdennimi' solusta B1 arvon/tekstin ja lisää sen soluun A1.
 1. Solu voidaan siis osoittaa samalla välilehdellä Range("A1") ja toisella välilehdellä Worksheets("Välilehdennimi").Range("A1") ja alue Range("A1:C100")
 2. Worksheets("Välilehdennimi").Range("A1").ClearContents tyhjentää solun arvon, mutta säilyttää muotoilun.
 3. Worksheets("Välilehdennimi").Range("A1:A10").Copy kopioi leikepöydälle alueella A1:A10 olevat arvot ja muotoilun.
 1. Huomaa, että jos soluissa on rivinvaihtoja (alt + enter), ne laitetaan sitaateihin. Jokainen tekstikappale kannattaa täten laittaa omaan soluun. Tyhjä rivinvaihto muodostetaan tyhjällä solulla.
5. Paininike pitäisi toimia suoraan sitä klikatessa. Jos näin ei tapahdu, paina painikkeen päällä hiiren oikealla ja valitse Assign Macro... ja yhdistä haluttu makro painikkeeseen.

- Makroja sisältävä tiedosto tallennetaan xlsx -muodossa.

HUOM! Jos makrot ovat välilehtiin liittyvissä ikkunoissa, välilehden poistaminen TUHOAA myös sen makrot!!!

- Jos makro tallentaa esimerkiksi tilapäisen tiedoston samaan hakemistoon, jossa varsinainen taulukkotiedosto sijaitsee ja tulee path-virheilmoitus. Jos taulukkotiedosto on OneDrivessä, siirrä tiedosto suoraan tietokoneen C-hakemistoon. Muutoin säädä polkua.
- Module-ikkunassa tehdyt muutokset ovat välittömästi käytettävissä.
- Jos makrot eivät muutosten jälkeen toimi, ks. ikoniriviltä Visual Basic-editorissa onko "joku" päällä ja paina 'Reset'-ikonina. Yleensä ilmenee kun makroissa on virheitä.
- Painikkeita ja makroja voi yhdistellä myöhemmin, esim. muutettaessa nimiä painamalla hiiren oikeaa painikkeen päällä ja valitsemalla 'Liitä makro...'

- Makroiin pääsee myös: Tools > Macro > Visual Basic Editor
- Uusi makro: Insert > Module
- Makroa voi käyttää myös solussa viittaamalla makrofunktion nimeen, esim. =FunktionNimi(A1)
- Hävinnyt Makron salasana

1. Työkalut > Makro > Visual Basic Editor
2. Avautuu ikkuna, jonka vasemmasta palkista avaa hakemistopuu näkyviin
3. Etsi sieltä makro

Haku

- Ohje puuttuu seuraavalta, tarkenna tähän.
 - =JOS(ONLUKU(KÄY.LÄPI("Haettava sana";A1:A100));B1:B100;"")
- Haku soluista määrätystä välilehdeltä kun ehto täyttyy. Kopioidaan koko löydetty rivi toiselle välilehdelle.
 - Esimerkki 1

```
Dim Src As Worksheet, Trg As Worksheet
Dim i As Long, j As Integer
Dim muuttujal As String, muuttuja2 As String
Set Src = ActiveWorkbook.Worksheets("välilehdennimi1") 'välilehti jossa tehdään haku
Set Trg = ActiveWorkbook.Worksheets("välilehdennimi2") 'välilehti johon tulostetaan haun tulokset
j = 5 'aloitetaan hakutulosten tulostus kohdevälilehden riviltä 5
If muuttujal <> "" And muuttuja2 = "sana" Then
For i = 2 To 10 ' aloitetaan riviltä 2 ja lopetetaan riville 10
    If Src.Cells(i, "D") = muuttuja2 Or Source.Cells(i, "E") = muuttuja2 Then
'etstiään sarakkeista D ja E
        Src.Rows(i).EntireRow.Copy Trg.Rows(j) 'kopioidaan sanan sisältävä
koko rivi toiselle välilehdelle
        j = j + 1 'jos useita erilaisia hakuja, tämä säilyttää tulossivun
seuraavan tyhjän rivin
    End If
Next i
End If
```

- Esimerkki 2. Tyhjän muuttujan hakemin saattaa kaataa Excelin!

```
Dim c As Range, j As Integer
For Each c In Src.Range("C2:F10") 'Src ks. esimerkki 1
    If c = muuttuja Then
        Src.Rows(c.Row).Copy Trg.Rows(j) 'Trg ks. esimerkki 1
        j = j + 1
    End If
Next c
```

- Vapaasana-haku
 - Hakee sanan, sanan osan tai peräkkäisten sanojen yhdistelmän
 - Kutsutaan esim. toisesta Sub:sta seuraavasti: Vapaasanahaku vapaasana, j, Src, Trg, vapaahakualue
 - vapaasana-muuttujassa on haettava sana, esim. ruusu

- j-muuttuja kertoo toisen välilehden seuraavan vapaan rivin, jolle löydetyn sanan koko rivi kopioidaan
- vapaahakualue on määritelty: Set vapaahakualue = Src.Range("B\$2:G\$100")
- Src ja Trg viittaavat välilehden nimeen, ks. toisaalla tällä sivulla.

```
Public Function Vapaasanahaku(vapaasana, j, Src, Trg, vapaahakualue)
    Dim cl As Range
    Dim FirstFound As String
    Dim sh As Worksheet
    Dim rivi As Variant
    Dim b As Range
    Application.FindFormat.Clear
    For Each b In vapaahakualue.Rows
        rivi = b.Row 'tallennetaan
        Set cl = b.Cells.Find(What:=vapaasana, _
            After:=b.Cells(1, 1), _
            LookIn:=xlValues, _
            LookAt:=xlPart, _
            SearchOrder:=xlByRows, _
            SearchDirection:=xlNext, _
            MatchCase:=False, _
            SearchFormat:=False)
        If Not cl Is Nothing Then
            FirstFound = cl.Address
            Do
                Src.Rows(rivi).EntireRow.Copy Trg.Rows(j)
                j = j + 1
                Set cl = b.Cells.FindNext(After:=cl)
            Loop Until FirstFound = cl.Address
        End If
    Next
End Function
```

Lähde

<https://stackoverflow.com/questions/11813720/search-for-a-string-in-a-worksheet-using-vba>

- Haku toiselta välilehdeltä ja löydetyn viereinen solu
 - Käytä hakusanana koko solun riviä, esim. >results>name eikä pelkästään name

```
Sub HakuSivulta()
    Dim VL1 As Worksheet: Set VL1 = Worksheets("Välilehdennimi jossa haettava
sana ja johon tulos")
    Dim VL2 As Worksheet: Set VL2 = Worksheets("Välilehdennimi josta
haetaan")
    Dim FoundCell As Range: Set FoundCell =
VL2.Range("A:A").Find(VL1.Range("A1").Value, LookIn:=xlValues,
LookAt:=xlWhole)
    'Yllä haetaan toisen välilehden sarakkeesta A sanaa joka on annettu
toisella välilehdellä solussa A1
    If FoundCell Is Nothing Then
        Set FoundCell = Nothing
    End If
End Sub
```



```

        Set VL1 = Nothing
        Set VL2 = Nothing
    Exit Sub
Else
    VL1.Range("A2") = FoundCell.Offset(0, 1) 'Tulostetaan soluun A2
löydetystä tiedosta oikealla olevasta solusta arvo
End If
Set FoundCell = Nothing
Set VL1 = Nothing
Set VL2 = Nothing
End Sub

```

- Lähde, Source: ?

Google-haku, Chrome-selain

- Lisää painike ja painikkeelle seuraava VBA-koodi. Huomaa, chromePolun sijainti on eri 32 ja 64-bittisissä Windowseissa. Toimii Windows 10.

```

Sub GoogleHaku()
    Dim chromePolku As String
    Dim haku_ehto As String
    haku_ehto = Worksheets("Välilehdennimi").Range("A10") 'Missä solussa
hakusanat
    'haku_ehto = "Haettava teksti" jos teksti määritetään suoraan tässä
koodissa
    haku_ehto = Replace(haku_ehto, " ", "+")
    chromePolku = "C:\Program Files
(x86)\Google\Chrome\Application\chrome.exe" '64-bittinen Windows
Shell (chromePolku & " -url http://google.ca/#q=" & haku_ehto)
End Sub

```

Kopiointi

- Rivin kopiointi välilehdeltä toiselle. Katso tarkemmin esimerkit Haku-otsikon alta, jossa for-silmukka (i) ja tulostussivun seuraavasta rivistä (j)

```

Src.Rows(i).EntireRow.Copy Trg.Rows(j) 'Src ja Trg viittaavat välilehden
nimii, ks. toisaalla täällä Wikikko-sivulla.
Src.Rows(c.Row).Copy Trg.Rows(j)

```

Noutaminen

- VBA: Solun arvojen noutaminen välilehdeltä makroon:

```

Dim Src As Worksheet
Dim muuttuja1 As Variant
Dim muuttuja2 As String
Set Src= ActiveWorkbook.Worksheets("välilehdennimi")

muuttuja1 = Worksheets("välilehdennimi").Range("D5")
muuttuja1 = Src.Range("D5")

```

```
With Src
    muuttuja1 = .Range("D5")
    muuttuja2 = .Range("S6")
End With
```

Numerot kirjaimiksi

Excelissä ei ole valmista funktiota, jolla voisi esimerkiksi rahasummat muuttaa kirjaimiksi, joten se on kirjoitettava VBA:lla. Toimiva löytyy 8/2022 esim.

<https://www.extendoffice.com/documents/excel/1446-excel-convert-change-numbers-to-words.html>

PDF

```
Sub LuoPDF()
    Dim Src As Worksheet
    Dim alue As Range
    Dim tiedostonimi As String

    Set Src = ActiveWorkbook.Worksheets("välilehdennimi")
    Set alue = Src.Range("A2:F100") 'tulostettava alue
    'Muotoillaan sivua ennen vientiä. Voi asettaa myös välilehden
tulostusasetuksissa
    Src.PageSetup.Zoom = False
    Src.PageSetup.FitToPagesWide = 1
    Src.PageSetup.Orientation = xlLandscape
    tiedostonimi = "jokunimitähän" & "_" & Format(Now(), "yymmdd_hhmm") &
".pdf"
    tiedostonimi = Environ("USERPROFILE") & "\Desktop\" & tiedostonimi
'tallennetaan Windowsin työpöydälle
    alue.ExportAsFixedFormat Type:=xlTypePDF, _
    Filename:=tiedostonimi, _
    Quality:=xlQualityStandard, _
    IncludeDocProperties:=True, _
    IgnorePrintAreas:=True, _
    OpenAfterPublish:=True 'jos ei haluta aukeavan tallennuksen jälkeen,
asetta False
End Sub
```

Lähde: <http://www.jtrive.com/convertng-excel-worksheets-to-pdf-with-vba.html>

Ponnahdusikkuna

- Lisää painike ja yhdistä se seuraavaan makroon:

```
Sub Makron_nimi()
    MsgBox Worksheets("Välilehdennimi").Range("B10"), , "Ponnahdusikkunan
otsikko" 'hakee tekstin annetulta välilehdeltä.
End Sub
```

- Taulukosta haettavassa tekstissä voi tehdä rivinvaihtoja Alt + Enter
- Vaihtoehtoja:
 - MsgBox "Teksti tähän"

- MsgBox "Ensimmäinen rivi" & vbNewLine & "Toinen rivi"
- MsgBox muuttujatähän
- MsgBox "Kysymys?", vbYesNoCancel, "Ponnahdusikkunan otsikko" 'Painikkeet Yes, No, Cancel
- If MsgBox("Kysymys?") _ = vbYes Then MsgBox "Vastasit Kyllä.", , "Ponnahdusikkunan otsikko" Else MsgBox "Vastasit Ei.", , "Ponnahdusikkunan otsikko" End If
- Ks. kohta 'Jos', jossa on esimerkki Kyllä, Ei, Peruuta toteutettuna Select-komennolla.

- Ilmoitus jos (If)-ehdosta:

```
If muuttuja1 = "" And muuttuja2 = "" Then
MsgBox "Hakuehto puuttuu.", vbExclamation
Exit Sub
End If
```

Pudotusvalikko

- Avaa Visual Basic-editorssa 'TämäTyökirja'-ikkuna
- Valitse Workbook ja Open
- Kirjoita seuraava:

```
Private Sub Workbook_Open()
Worksheets("Välilehdennimi").Range("A1").Validation.Add
Type:=xlValidateList, AlertStyle:=xlValidAlertStop, _
Formula:="valinta1,valinta2,jne"
End Sub
```

- Tallenna tiedosto > Sulje tiedosto > Avaa tiedosto > Pudotusvalikko ilmestyy soluun A1, kun sen valitsee (pieni väkänä oikeassa alakulmassa ulkopuolella solua).
- Huomaa, että tähän tehdyt muutokset vaativat aina tiedoston sulkemisen ja uudelleenavaamisen!

Rivi

```
For Each b In vapaahakuaalue.Rows
rivi = b.Row 'tallennetaan silmukasta meneillään olevan rivin numero
...
```

Solun arvon muuttaminen automaattisesti muutettaessa toista solua

1. Klikkaa hiiren oikealla sen välilehden nimeä ikkunan alareunassa, jossa on solu tai alue, johon tehtäessä muutos saadua aikaan muutos jossain muualla.
2. Valitse Näytä koodi
3. Avautuu VBA ja ikkuna, jossa lukee Worksheet - Change
4. Lisää seuraava koodi. Esimerkissä solun A1 arvon muuttaminen saa aikaiseksi, että Välilehdennimi1 solun A1 arvoksi asetetaan arvo B2 Välilehdennimi2:sta. Alue määritetään esim Range("A1:B10")

```
Private Sub Worksheet_Change(ByVal Target As Range)
Dim KeyCells As Range
Set KeyCells = Range("A1")
If Not Application.Intersect(KeyCells, Range(Target.Address)) _
Is Nothing Then
Sheets("Välilehdennimi1").Range("A1") =
```

```
Worksheets("Välilehdennimi2").Range("B2")
End If
End Sub
```

Lähde: <https://docs.microsoft.com/en-us/office/troubleshoot/excel/run-macro-cells-change>

Solun tyjentäminen

```
Sub Tyhjenna_alue()
    Dim Trg As Worksheet
    Set Trg = ActiveWorkbook.Worksheets("välilehdennimi")
    Trg.Range("A5:G10").ClearContents 'tyhjennetään sisältö
    Trg.Range("A5:G10").ClearFormats 'tyhjennetään muotoilu
End Sub
```

Solun värjääminen

Värjää valittu solu punaisella. Lisää painike ja painikkeelle seuraava VBA-koodi.

```
Sub Värjää_solu()
    ActiveCell.Select
    ActiveCell.Interior.ColorIndex = 3
End Sub
```

- Värivaihtoehdot: <https://www.excel-easy.com/vba/examples/background-colors.html>

URL-osoite linkistä

- Kun URL-osoite on solussa tekstin "takana"
- Kirjoita Visual Basicissa seuraava funktio ja kirjoita tyhjään soluun =TulostaURL(tähän linkin sisältävän solu)

```
Public Function TulostaURL(c As Range) As String
    On Error Resume Next
    GetURL = c.Hyperlinks(1).Address
End Function
```

Vie solu lainausmerkeissä tekstitiedostoon

- Esim. "teksti","teksti"

Kirjoita Visual Basicissa seuraava funktio

Lähde:

<https://support.microsoft.com/en-us/help/291296/procedure-to-export-a-text-file-with-both-comma-and-quote-delimiters-i>

```
Sub QuoteCommaExport()
    ' Dimension all variables.
    Dim DestFile As String
    Dim FileNum As Integer
    Dim ColumnCount As Integer
```

```

Dim RowCount As Integer

' Prompt user for destination file name.
DestFile = InputBox("Enter the destination filename" _
    & Chr(10) & "(with complete path):", "Quote-Comma Exporter")

' Obtain next free file handle number.
FileNum = FreeFile()

' Turn error checking off.
On Error Resume Next

' Attempt to open destination file for output.
Open DestFile For Output As #FileNum

' If an error occurs report it and end.
If Err <> 0 Then
    MsgBox "Cannot open filename " & DestFile
    End
End If

' Turn error checking on.
On Error GoTo 0

' Loop for each row in selection.
For RowCount = 1 To Selection.Rows.Count

    ' Loop for each column in selection.
    For ColumnCount = 1 To Selection.Columns.Count

        ' Write current cell's text to file with quotation marks.
        Print #FileNum, """" & Selection.Cells(RowCount, _
            ColumnCount).Text & """";

        ' Check if cell is in last column.
        If ColumnCount = Selection.Columns.Count Then
            ' If so, then write a blank line.
            Print #FileNum,
        Else
            ' Otherwise, write a comma.
            Print #FileNum, ",";
        End If
        ' Start next iteration of ColumnCount loop.
    Next ColumnCount
' Start next iteration of RowCount loop.
Next RowCount

' Close destination file.
Close #FileNum
End Sub

```

JSON -tuonti Exceliin

JSON-tiedostojen tuonti Exceliin ei ole ihan yksinkertainen.

JSON-esimerkki 1, YTJ

Seuraavassa eräs ohje, esimerkkinä yrityksen tietojen haku YTJ:n avoimesta datasta. Windows 10. Käytetty 8/2020.

- Lähde, source: <https://excelrocks.com/how-to-import-json-to-excel-using-excel-vba>
- Siirrä xlsx-tilukkotiedosto OneDrivestä tietokoneen C-hakemiston juureen (toimii ainakin siellä).
- Lisää taulukkoon välilehti, jossa käsitellään haettava hakusana ja johon tulos, tässä *Main*, välilehden nimen voi muuttaa koodin alussa.
- Lisää taulukkoon välilehti, johon JSON-tieto tuodaan, tässä YTJ-tulokset, välilehden nimen voi muuttaa koodin alussa.
- Lisää VBA-moduuli nimeltä *modjson* ja siihen seuraava koodi.
 - Moduulin nimen muuttaminen jälkeenpäin: VBA-editorissa paina Ctrl-R > Paina F4 > Valitse moduuli ylläruudussa ja muuta nimi alaruutuun välilehdellä Alphabetic.

Option Explicit

```
'*****  
'* Code to exercise the clsJson class: *  
'* Read data from a URL and store to a file, *  
'* read the file and parse out JSON data *  
'* Sanjay Gour *  
'* https://excelrocks.com/how-to-import-json-to-excel-using-excel-vba *  
'*****  
  
'Get data from a URL and store to "\\YTJ-haku_tilapäinen_poistettava.txt" in  
the same  
'directory as the excel spreadsheet.  
Public Sub StoretHTTPToTextFile()  
    Dim xmlHttp As Object  
    Dim strReturn As String  
    Dim strURL As String  
    Dim strYtunnusCell As String  
    Let strYtunnusCell = Worksheets("Main").Range("A1") 'Missä solussa  
annetaan y-tunnus muodossa 1234567-8  
  
    Dim strLoadingCell As Object  
    Set strLoadingCell = Worksheets("Main").Range("A2") 'Mihin soluun  
ilmestyy tilapäinen "Noudetaan..."  
    Dim strResultSheet As Object  
    Set strResultSheet = Worksheets("YTJ-tulokset") 'Hakutulosten välilehden  
nimi, lisää välilehti etukäteen  
    'HUOM! Muuta koodiin välilehden nimi kohtiin YTJ-tulokset, mitkä eivät  
muutu yllä olevalla objektilla  
    strURL = "http://avoindata.prh.fi/bis/v1/" & strYtunnusCell 'URL-osoite  
ilman hakusanaa
```

```

    strLoadingCell.Value = "Noudetaan tietoja YTJ-palvelusta..."
'osoitteesta" & strURL
    'Open URL and get JSON data
    Set xmlHttp = CreateObject("MSXML2.ServerXMLHTTP.6.0")
    xmlHttp.Open "GET", strURL
    xmlHttp.setRequestHeader "Content-Type", "text/xml"
    xmlHttp.send
    'Save the response to a string
    strReturn = xmlHttp.responseText
    'save JSON text to file
    ExportToTextFile ActiveWorkbook.Path & "\YTJ-
haku_tilapäinen_poistettava.txt", strReturn
    Call LoadTheTextFileAndParse
    strLoadingCell = ""
    'Go to 'YTJ-tulokset' sheet and select the first cell
    strResultSheet.Activate
    strResultSheet.Range("A1").Select
End Sub
'Import a set of JSON data (from JSON_temp.txt located
'in the same directory as this worksheet)
'and move JSON data keys to column A and JSON
'values for the keys in Column B
'of the worksheet
Private Sub LoadTheTextFileAndParse()
    Dim intIndex As Integer
    Dim strVar As String
    Dim strStatus As String
    Dim strErrStatus As String
    Dim intDebugLevel As Integer
    Dim clsJson As clsJson
    Dim strJSONTempFile As String
    '0 = no info
    '1 = debug.print status info
    intDebugLevel = 1
    'Set the temporary JSON temp file path in the current folder
    strJSONTempFile = ActiveWorkbook.Path & "\YTJ-
haku_tilapäinen_poistettava.txt"
    'Get JSON text string from file
    strVar = ImportTextFile(strJSONTempFile)

    'Clear old contents in spreadsheet
    Sheets("YTJ-tulokset").Cells.ClearContents
    Sheets("YTJ-tulokset").Cells.ClearFormats
    Sheets("YTJ-tulokset").Columns("B:B").HorizontalAlignment = xlLeft
    'Set the class connection
    Set clsJson = New clsJson
    If intDebugLevel > 0 Then
        Debug.Print "Tila ennen noutamista: " & clsJson.err
    End If
    'The first step is to load the JSON information
    'The loaded information will then be available

```

```

'through the clsJSON.Key and clsJSON.Value data pairs
clsJson.LoadTheJSONString strVar

If intDebugLevel > 0 Then
    Debug.Print "Tila noutamisen jälkeen: " & clsJson.err
    Debug.Print "Noudettu " & clsJson.NumElements & " elementtiä."
End If
strStatus = "Noudettu " & clsJson.NumElements & " elementtiä."
'Convert status data to text
Select Case clsJson.err
    Case -1
        strErrStatus = "JSON-jonoa ei voitu noutaa."
    Case -2
        strErrStatus = "JSON-jonoa ei voitu täydellisesti jäsentää."
    Case 1
        strErrStatus = "JSON-jono jäsennetty."
End Select
'Display status data
strStatus = strStatus & Chr(10) & strErrStatus & Chr(10) & Chr(10) & _
            "YTJ-tulokset on noudettu."
MsgBox strStatus, vbInformation, "TIETOJEN HAKU"
'Get all the elements of the parsed JSON text and put the data in cells
(in Sheets("YTJ-tulokset"))
For intindex = 1 To clsJson.NumElements
    Sheets("YTJ-tulokset").Cells(intindex, 1).Value =
clsJson.Key(intindex)
    Sheets("YTJ-tulokset").Cells(intindex, 2).Value =
clsJson.Value(intindex)
Next
'Lastly, delete the JSON_temp.txt file
If Len(Dir$(strJSONTempFile)) > 0 Then Kill strJSONTempFile
End Sub

'Load a text file to a string
Private Function ImportTextFile(strFile As String) As String
    Open strFile For Input As #1
    ImportTextFile = Input$(LOF(1), 1)
    Close #1
End Function
'Save a text string to a file
Private Function ExportToTextFile(strFile As String, strText As String) As
String
    Dim fso As Object
    Dim oFile As Object
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set oFile = fso.CreateTextFile(strFile)
    oFile.WriteLine strText
    oFile.Close
    Set fso = Nothing
    Set oFile = Nothing
End Function

```


- Lisää 'Main'-välilehdelle painike ja liitä siihen edellinen makro eli StoreHTTPToTextFile
- Lisää VBA-Class moduuli nimeltä *clsjson* Lisäys tapahtuu Insert > Class Module ja nimeä tarvittaessa uudestaan kuten yllä ohjeistettu.

Option Explicit

```
Private strKey As Variant
Private strVal As Variant
Private intHMax As Integer
Private lngStatus As Long
```

```
Private Sub Class_Initialize()
    lngStatus = -1
End Sub
```

```
Public Property Get err() As Long
```

```
'Status:
' 1 = JSON string has been successfully parsed
' -1 = JSON string has not been loaded, no results are available
' -2 = JSON string cannot be fully parsed (JSON text not fully or
incorrectly formed)
    err = lngStatus
End Property
```

```
Public Property Get NumElements() As Integer
```

```
    NumElements = intHMax
End Property
```

```
Public Property Get Key(Index As Integer) As Variant
```

```
    If Index > UBound(strKey) Or Index < LBound(strKey) Then
        Key = ""
    Else
        Key = strKey(Index)
    End If
End Property
```

```
Public Property Get Value(Index As Integer) As Variant
```

```
    If Index > UBound(strVal) Or Index < LBound(strVal) Then
        Value = ""
    Else
        Value = strVal(Index)
    End If
End Property
```

```
'This method Loads the JSON text into an array
```

```
Public Sub LoadTheJSONString(JSONText As String)
    'Maximum Value for Long type
    Const cLongMax = (2 ^ 31) - 1
    Dim lngIndex As Long
    Dim lngContLoc As Long
    Dim lngLoc As Long
```

```

Dim lngDelimitOffset As Long
Dim lngASize As Long
Dim intNoOfChecks As Integer 'Number of different control characters in
JSON
Dim intCheck As Integer
Dim intCtrlChr As Integer
Dim intObjLvl As Integer
Dim intAryElement As Integer
Dim intLvl As Integer
Dim strID As String
Dim strChr As String
Dim strKeyValue As String
Dim strValue As String
Dim strPChar As String
Dim strFoundVal As String
Dim strTempString As String
Dim strAKey() As String
Dim strAVal() As String
Dim strALvlKey(100) As String
Dim blArray As Boolean 'Flag to indicate that an array has been
found
Dim blStringArray As Boolean 'Flag to indicate that the element in the
array is a string
Dim BlArrayEnd As Boolean 'Flag to indicate that the end of an
array is found
Dim blValue As Boolean 'Flag to indicate that a value has been
found
Dim blKeyAndValue As Boolean 'Found a key and value pair
Dim blDebug As Boolean
'Set the flag to true if you want to see
'debug information during the loading process
blDebug = True
On Error GoTo ErrHandler:
lngASize = 10
ReDim strAKey(lngASize)
ReDim strAVal(lngASize)
'intArrayElement = 1 'initialize value
'initialize values
blArray = False
BlArrayEnd = False
blStringArray = False
'Generate a string of control characters
'String is {[: , ]}'
strID = ""
strID = strID & Chr(123) 'The '{' character
strID = strID & Chr(91) 'The '[' character
strID = strID & Chr(58) 'The ':' character
strID = strID & Chr(44) 'The ',' character
strID = strID & Chr(93) 'The ']' character
strID = strID & Chr(125) 'The '}' character
strID = strID & Chr(34) 'The '"' character

```

```

intNoOfChecks = Len(strID)
intObjLvl = 0
'First element in the array will be strKey(1) and strVal(1)
lngIndex = 1
'As we process the JSON string it becomes shorter and
'shorter, until its all been processed
Do While Len(JSONText) > 0
    'Set to maximum value as default
    lngContLoc = cLongMax
    'Find Next control character:
    'Scan the text for the closest control character
    'to the beginning of the remaining JSON text
    For intCheck = 1 To intNoOfChecks
        strChr = Mid(strID, intCheck, 1)
        lngLoc = InStr(1, JSONText, strChr, vbBinaryCompare)
        If (lngLoc > 0) And (lngLoc < lngContLoc) Then
            lngContLoc = lngLoc
            intCtrlChr = intCheck
            strPChar = strChr
        End If
    Next intCheck
    'When the above For Next Loop ends we will have found
    'the closest control character
    'stored in intCtrlChr - an index (1 to 8) to the
    'found character in strChr
    'stored in lngContLoc - position of the next
    'control character
    'stored in strPChar - the closest next control character
    If blDebug = True Then
        Debug.Print "Parse Character: " & strPChar
    End If
    'A control character has been found, figure out what to do by the
found character
    If lngContLoc < cLongMax Then
        'Capture the information before the control character
        strValue = Mid(JSONText, 1, lngContLoc - 1)
        'Capture everything after the control character (the remaining
JSON string)
        JSONText = Mid(JSONText, lngContLoc + 1, Len(JSONText))
    Else
        'We found the end of the JSON string
        Exit Do
    End If
    'Found a number or boolean value or key (the comma)
    'Handles number types in array (process value as
'string or number; not both)
    If (intCtrlChr = 4) Then
        If ((blValue = True) Or (blArray = True)) And (blStringArray =
False) Then
            'Found a value, and we already have key
            strFoundVal = ConvertStringToValue(strValue)

```

```

        blKeyAndValue = True 'Set the "Key and value found" flag
    End If
    'Finding a comma resets the string found in the array
    blStringArray = False
End If
'Start of object (The "{" character)
If intCtrlChr = 1 Then
    intObjLvl = intObjLvl + 1
    blArray = False 'An object, not an array
    blValue = False 'Need to find a key first
    If blDebug = True Then
        Debug.Print "Start of Object, Moved up to level" & intObjLvl
    End If
End If
'End of of object (The "}" character)
If intCtrlChr = 6 Then
    'Numbers preceded by the "}" character
    If blValue = True Then
        'Get the found value and set a flag
        strFoundVal = ConvertStringToValue(strValue)
        blKeyAndValue = True 'Set the "Key and value found" flag
        'Add back a "}" character to the string so that the level can
be decremented properly
        JSONText = "}" & JSONText
    Else
        'No value was found, the "}" character indicates the end of
this level
        intObjLvl = intObjLvl - 1
        blValue = False 'Need to find a key first
    End If
    If blDebug = True Then
        Debug.Print "End of Object, Moved down to level" & intObjLvl
    End If
End If
'Start of array (The "[" character)
If intCtrlChr = 2 Then
    blArray = True
    blValue = True 'Next thing should be a value
    intAryElement = 1
    If blDebug = True Then
        Debug.Print "Start of Array, Moved up to level" & intObjLvl
    End If
End If
'End of of array (The "]" character)
If intCtrlChr = 5 Then
    'Parse last numeric or boolean value of an array
    If (blArray = True) And (blStringArray = False) Then
        'Get the found value and set a flag
        strFoundVal = ConvertStringToValue(strValue)
        blKeyAndValue = True 'Set the "Key and value found" flag
    End If

```

```

BlArrayEnd = True 'Mark that the end of the array is found
blArray = False
blValue = False 'Need to find a key first
If blDebug = True Then
    Debug.Print "End of Array, Moved down to level" & intObjLvl
End If
End If
'Object Value start is found (The ":" character)
If intCtrlChr = 3 Then
    blValue = True
    'Start of an object value is not the end of an array
    BlArrayEnd = False
    If blDebug = True Then
        Debug.Print "ready to get value"
    End If
End If
'Start of a string (the quote " character)
'Can be a key or value
If intCtrlChr = 7 Then
    'The start of the key or value has been found
    'The next quote will end the key or value
    '(unless the quote has an escape character in front of it "\")
    lngDelimitOffset = 1
    Do
        'Look for the next quote character
        lngLoc = InStr(lngDelimitOffset, JSONText, Chr(34),
vbBinaryCompare)
        'If the string is zero length "" then exit the loop
        If lngLoc = 1 Then
            Exit Do
        End If
        'Check to see if there is a delimiter just before the quote
        'if there is then quote is part of the string and not the end
of
        'the string.
        If Mid(JSONText, lngLoc - 1, 1) = Chr(92) Then
            ' The quote character has an escape character in front of
it
            'so this quote doesn't count. Remove the escape
character.
            JSONText = Mid(JSONText, 1, lngLoc - 2) & Mid(JSONText,
lngLoc, Len(JSONText))
            'and move the start of the check past the delimited quote
            lngDelimitOffset = lngLoc
            'If we have a messed up JSON string where there
            'is no valid closing quotes the above "If" will
            'cause an error (the "Mid" statement will attempt
            'to check the string starting at a position of
            '-1) and the code will jump to the error handling
            'section. If this error didn't occur the
            'Do..Loop would get stuck.

```

```

        Else
            Exit Do
        End If
Loop
'We now have a string, find any other delimiters
'(any delimited " characters have already been fixed)
strTempString = FixTheJSONString(Mid(JSOText, 1, lngLoc - 1))
If (blValue = True) Or (blArray = True) Then
    'The key has been previously found and this is
    'the value for the key
    strFoundVal = strTempString
    'Set the "Key and value found" flag
    blKeyAndValue = True
    If blArray = True Then
        'The value is a string
        blStringArray = True
    End If
Else
    If lngLoc > 0 Then
        'We've found a key
        strALvlKey(intObjJLvl) = strTempString
        If blDebug = True Then
            Debug.Print "Found Key:" & strALvlKey(intObjJLvl) & _
                " for Level: " & intObjJLvl
        End If
    End If
End If
JSOText = Mid(JSOText, lngLoc + 1, Len(JSOText))
End If
'Found a key and value, move it to the array
If blKeyAndValue = True Then
    If lngIndex > lngASize Then
        lngASize = lngASize + 100
        ReDim Preserve strAKey(lngASize)
        ReDim Preserve strAVal(lngASize)
    End If
    strAKey(lngIndex) = ""
    For intLvl = 1 To intObjJLvl
        strAKey(lngIndex) = strAKey(lngIndex) & ">" &
strALvlKey(intLvl)
    Next intLvl
    'Save last element of an array
    If (blArray = True) Or (BlArrayEnd = True) Then
        'add the array element to the key
        strAKey(lngIndex) = strAKey(lngIndex) & ">" &
Trim(Str(intAryElement))
        'increment the array element
        intAryElement = intAryElement + 1
        'Reset end of array flag (Reset, when array end is found)
        BlArrayEnd = False
    End If
End If

```

```

    strAVal(lngIndex) = strFoundVal
    If blDebug = True Then
        Debug.Print "Added Key:" & strAKey(lngIndex) & _
            " Value: " & strAVal(lngIndex) & " index: " & lngIndex
    End If
    lngIndex = lngIndex + 1 'Increment the array
    blKeyAndValue = False 'Reset the "found" flag
    blValue = False 'Reset the "Value Found" flag
End If
DoEvents
Loop
'Number of items found
intHMax = lngIndex - 1
strKey = strAKey
strVal = strAVal
lngStatus = 1 'JSON successfully parsed
Exit Sub

```

ErrorHandler:

```

'Error handling
lngStatus = -2 'JSON Parse error
'Uncomment the next 3 lines to figure out the
'cause of the issue:
'Debug.Print VBA.err.Number
'Debug.Print VBA.err.Description
'Resume

```

End Sub

'This function converts a string that contains formatting
'information into a string that only contains a value.
'Values can be text, integer, or floating point values.
'null is passed back as a zero length string: "".

Private Function ConvertStringToValue(strInStr As String) As String

```

    Dim intStrPos As Integer
    Dim strTemp As String
    Dim intChar As Integer
    'default value
    strTemp = ""
    'Make sure that the string does not have a zero length
    strInStr = " " & strInStr
    'Loop through each character in the string and remove anything
    'that is not alphanumeric.
    For intStrPos = 1 To Len(strInStr)
        intChar = Asc(Mid(strInStr, intStrPos, 1))
        If ((intChar >= Asc("a")) And (intChar <= Asc("z"))) Or _
            ((intChar >= Asc("A")) And (intChar <= Asc("Z"))) Or _
            ((intChar >= Asc("0")) And (intChar <= Asc("9"))) Or _
            (intChar = Asc(".")) Or (intChar = Asc("+")) Or _
                (intChar = Asc("-")) Then
            strTemp = strTemp & Chr(intChar)
        End If
    
```

```

Next intStrPos
'Values that are listed as 'null' are converted to a zero length string
If InStr(1, "null", strTemp, vbTextCompare) > 0 Then
    strTemp = ""
End If
ConvertStringToValue = strTemp
End Function

```

```

'This function goes through a JSON string and corrects
'delimited characters

```

```

Private Function FixTheJSONString(strInput As String) As String
    Dim blParseComplete As Boolean
    Dim lngStartPos As Long
    Dim lngCurrentPos As Long
    blParseComplete = False
    lngStartPos = 1
    Do While blParseComplete = False
        'If we don't find any escape sequences then allow the loop to end
        blParseComplete = True
        'Escaped sequence: replace \\ with \
        'look for the the specific escape sequence
        lngCurrentPos = InStr(lngStartPos, strInput, "\\ ", vbTextCompare)
        If lngCurrentPos > 0 Then
            strInput = Mid(strInput, 1, lngCurrentPos - 1) & "\" & _
                Mid(strInput, lngCurrentPos + 2, Len(strInput))
            'set the status to check for another escape
            blParseComplete = False
        End If
        'Escaped sequence: replace \/ with /
        'look for the the specific escape sequence
        lngCurrentPos = InStr(lngStartPos, strInput, "\/", vbTextCompare)
        If lngCurrentPos > 0 Then
            strInput = Mid(strInput, 1, lngCurrentPos - 1) & "/" & _
                Mid(strInput, lngCurrentPos + 2, Len(strInput))
            'set the status to check for another escape
            blParseComplete = False
        End If
        'Escaped sequence: replace \b with a backspace
        'look for the the specific escape sequence
        lngCurrentPos = InStr(lngStartPos, strInput, "\b", vbTextCompare)
        If lngCurrentPos > 0 Then
            strInput = Mid(strInput, 1, lngCurrentPos - 1) & Chr(8) & _
                Mid(strInput, lngCurrentPos + 2, Len(strInput))
            'set the status to check for another escape
            blParseComplete = False
        End If
        'Escaped sequence: replace \f with a formfeed
        'look for the the specific escape sequence
        lngCurrentPos = InStr(lngStartPos, strInput, "\f", vbTextCompare)
        If lngCurrentPos > 0 Then
            strInput = Mid(strInput, 1, lngCurrentPos - 1) & Chr(12) & _

```



```

        Mid(strInput, lngCurrentPos + 2, Len(strInput))
        'set the status to check for another escape
        blParseComplete = False
    End If
    'Escaped sequence: replace \n with a newline
    'look for the the specific escape sequence
    lngCurrentPos = InStr(lngStartPos, strInput, "\n", vbTextCompare)
    If lngCurrentPos > 0 Then
        strInput = Mid(strInput, 1, lngCurrentPos - 1) & Chr(10) & _
            Mid(strInput, lngCurrentPos + 2, Len(strInput))
        'set the status to check for another escape
        blParseComplete = False
    End If
    'Escaped sequence: replace \r with a carriage return
    'look for the the specific escape sequence
    lngCurrentPos = InStr(lngStartPos, strInput, "\r", vbTextCompare)
    If lngCurrentPos > 0 Then
        strInput = Mid(strInput, 1, lngCurrentPos - 1) & Chr(13) & _
            Mid(strInput, lngCurrentPos + 2, Len(strInput))
        'set the status to check for another escape
        blParseComplete = False
    End If
    'Escaped sequence: replace \t with a horizontal tab
    'look for the the specific escape sequence
    lngCurrentPos = InStr(lngStartPos, strInput, "\t", vbTextCompare)
    If lngCurrentPos > 0 Then
        strInput = Mid(strInput, 1, lngCurrentPos - 1) & Chr(9) & _
            Mid(strInput, lngCurrentPos + 2, Len(strInput))
        blParseComplete = False 'set the status to check for another
escape
    End If
    'Escaped sequence: replace \uXXXX with a unicode character
    'look for the the specific escape sequence
    lngCurrentPos = InStr(lngStartPos, strInput, "\u", vbTextCompare)
    If lngCurrentPos > 0 Then
        strInput = Mid(strInput, 1, lngCurrentPos - 1) & _
            ChrW$(CLng("&h" & Mid(strInput, lngCurrentPos + 2,
4))) & _
            Mid(strInput, lngCurrentPos + 6, Len(strInput))
        'set the status to check for another escape
        blParseComplete = False
    End If
    Loop
    FixTheJSONString = strInput
End Function

```

- Syötä Main-välilehden soluun A1 sellainen y-tunnus, joka löytyy YTJ-sivustolta > paina lisäämäsi painiketta > Ilmestyy soluun A2 teksti "Noudetaan..." > Ilmestyy ponnahdusikkuna > Noudettu JSON-tieto löytyy toiselta välilehdeltä. Nyt tiedosta voi hakea esimerkiksi tällä sivulla olevan ohjeen mukaan, ks. *Haku toiselta välilehdeltä ja löydetyn viereinen solu*.

Lähteitä

<https://learn.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office>