


Jak změnit rastr v hotové tabulce Excelu

- 9.2.2017
- Petr Pecháček
- Excel a VBA
- Excel, mřížka, šířka sloupce

Znáte to. Máte v Excelu hotovou tabulku, ne-li celý formulář, hrajete si s každým pixelem, aby se vše vešlo na jednu A4, a zjistíte, že potřebujete přidat další informace, sloupec. Ve výsledku to znamená ještě lépe využít prostor, což je často doprovázeno nutností jemnějšího rastru (mřížky) a opětovným slučováním buněk. Práce k zbláznění. Předkládám makro, které ve výběru buněk přidá další sloupce tak, aby došlo ke zjemnění rastru (lidově řečeno se zdvojnásobí počet sloupců ve výběru při zachování původní šířky).

Na úvod tip, jak kontrolovat šířku tabulky před a po změně.

Q24								<i>f_x</i>
	A	B	C	D	E	F	G	
1								
2		11						
3		21	22			23		
4		31			32	33		
5		41	42		43			
6		51	52		53	54		
7		61			62			
8								
9								

Jemnější rastr - příprava

Chování makra ukazují obrázky.

B2							<i>f_x</i>	11
	A	B	C	D	E	F	G	
1								
2		11						
3		21	22				23	
4		31			32	33		
5		41	42	43				
6		51	52	53	54			
7		61			62			
8								
9								

B2											<i>f_x</i>	11
	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		11										
3		21	22									23
4		31					32	33				
5		41	42	43								
6		51	52	53	54							
7		61				62						
8												
9												

Jemnější rastr - Ukázka 1

E4							<i>f_x</i>	32
	A	B	C	D	E	F	G	
1								
2				11				
3		21		22			23	
4			31		32		33	
5		41		42		43		
6		51		52		53	54	
7			61			62		
8								
9								

E4								<i>f_x</i>	32
	A	B	C	D	E	F	G	H	
1									
2				11					
3		21		22				23	
4			31		32			33	
5		41		42		43			
6		51		52		53		54	
7			61			62			
8									
9									

Jemnější rastr - Ukázka 2

C6							fx	52
	A	B	C	D	E	F	G	
1								
2			11					
3		21		22		23		
4			31		32	33		
5		41	42		43			
6		51	52		53	54		
7			61		62			
8								
9								

C6							fx	52	
	A	B	C	D	E	F	G	H	I
1									
2			11						
3		21		22		23			
4			31		32	33			
5		41	42		43				
6		51	52		53	54			
7			61		62				
8									
9									

Jemnější rastr - Ukázka 3

A zde je slíbené makro. Důrazné varování: Změna provedená kódem je nevratná, proto sešit před použitím ukládejte/zálohujte!

Sub DvakrátJemnejsiRastr()

```
Dim rngBunka As Range
Dim rngSloupce As Range
Dim rngCeleSloupce As Range
```

```
Dim lngTopStyle As Long
Dim lngTopColor As Long
Dim lngRightStyle As Long
Dim lngRightColor As Long
Dim lngBottomStyle As Long
```

```
Dim lngBottomColor As Long
```

```
Dim intPocetSloupcu As Integer
```

```
Dim intSirkaPixely As Integer
```

```
Dim intNovaSirkaPixely As Integer
```

```
Dim i As Integer
```

```
Dim j As Integer
```

```
Dim dblNovaSirkaColumnWidth As Double
```

```
'dots per inch (DPI)
```

```
Const intPocetBoduNaPalec As Integer = 72
```

```
'pixels per inch (PPI)
```

```
'Microsoft: 96 PPI, Apple: 72 PPI
```

```
Const intPocetPixeluNaPalec As Integer = 96
```

```
'zamezeni prekreslovani a prepoctu listu
```

```
Application.ScreenUpdating = False
```

```
Application.Calculation = xlCalculationManual
```

```
'prevzeti zpracovavane oblasti z vyberu bunek
```

```
Set rngSloupce = Selection.Columns
```

```
Set rngCeleSloupce = Selection.EntireColumn.Columns
```

```
'pocet sloupcu
```

```
intPocetSloupcu = rngCeleSloupce.Columns.Count
```

```
'pro vsechny sloupce
```

```
For i = 1 To intPocetSloupcu
```

```
    'skutecne poradi puvodniho sloupce
```

```
    'po pridavani dalsich sloupcu
```

```
    j = 2 * i - 1
```

```
'zjisteni sirky sloupce a prepocet na pixely  
intSirkaPixely = rngCeleSloupce.Columns(j).Width / intPocetBoduNaPalec _  
    * intPocetPixeluNaPalec
```

```
'nova sirka sloupce bude polovicni...  
intNovaSirkaPixely = CInt(intSirkaPixely / 2)
```

```
'zpetny prepocet na vynucenou velikost v ColumnWidth  
' vychazi z regrese na experimentalnich hodnotach  
dblNovaSirkaColumnWidth = 0.142851762457295 * intNovaSirkaPixely - _  
    0.707857121632131
```

```
'pridani sloupce vlevo od sloupce nasledujiciho  
rngCeleSloupce.Columns(j + 1).Insert Shift:=xlToRight
```

```
'nastaveni nove sirky puvodniho a pridaneho sloupce  
rngCeleSloupce.Columns(j).ColumnWidth = dblNovaSirkaColumnWidth  
rngCeleSloupce.Columns(j + 1).ColumnWidth = dblNovaSirkaColumnWidth
```

```
'pro vsechny bunky pridaneho sloupce  
'a v nem vyuzite bunky  
For Each rngBunka In Intersect(ActiveSheet.UsedRange, _  
    rngCeleSloupce.Columns(j + 1)).Cells
```

```
'pokud se nejedna o sloucenou bunku  
If Not rngBunka.MergeCells Then
```

```
    'a pritom posledni sloupec...  
    If (j + 1) = (2 * intPocetSloupcu) Then
```

```
        With rngBunka.Offset(, -1).MergeArea
```

```
            'nacteni vlastnosti okraju z bunky vlevo
```

```
            lngTopStyle = .Borders(xlEdgeTop).LineStyle
```

```
lngTopColor = .Borders(xlEdgeTop).Color
```

```
lngRightStyle = .Borders(xlEdgeRight).LineStyle
```

```
lngRightColor = .Borders(xlEdgeRight).Color
```

```
lngBottomStyle = .Borders(xlEdgeBottom).LineStyle
```

```
lngBottomColor = .Borders(xlEdgeBottom).Color
```

```
End With
```

```
End If
```

```
'sloucení bunek
```

```
Range(rngBunka.Offset(-1), rngBunka).Merge
```

```
'a pro poslední sloupec...
```

```
If (j + 1) = (2 * intPocetSloupcu) Then
```

```
With rngBunka.MergeArea
```

```
'aplikování stylu okraje po sloučení
```

```
.Borders(xlEdgeTop).LineStyle = lngTopStyle
```

```
.Borders(xlEdgeTop).Color = lngTopColor
```

```
.Borders(xlEdgeRight).LineStyle = lngRightStyle
```

```
.Borders(xlEdgeRight).Color = lngRightColor
```

```
.Borders(xlEdgeBottom).LineStyle = lngRightStyle
```

```
.Borders(xlEdgeBottom).Color = lngBottomColor
```

```
End With
```

```
End If
```

End If

Next rngBunka

Next i

```
'povoleni prepoctu listu a prekreslovani
Application.Calculation = xlCalculationManual
Application.ScreenUpdating = True
```

End Sub

Pár poznámek. Jak si můžete přečíst v článku [Šířka sloupce a výška řádku v Excelu](#), nastavování šířky sloupce je peklo. V daném případě nestačí načítat a nastavovat vlastnost ColumnWidth, neboť je velmi nepřesná. Proto jsem pro definování nových šířek zpracovával informaci v pixelech (nutné hrátky s DPI) a následně podle sady experimentálních hodnot jsem si zjistil rovnici závislosti mezi ColumnWidth (písmo Calibri 11) a pixely (Zobrazení: Normálně) z lineární regrese.

Mode: normal x,y analysis

Polynomial degree 1, 1786 x,y data pairs.

Correlation coefficient (r^2) = 0.9999998493133465

Standard error = 0.02860707295026823

Coefficient output form: simple list:

-7.0785712163213077e-001

1.4285176245729497e-001

Mode: normal x,y analysis

Polynomial degree 2, 1786 x,y data pairs.

Correlation coefficient (r^2) = 0.9999998516878186

Standard error = 0.028380787766365163

Coefficient output form: simple list:

-6.9971323150786735e-001

1.4282467233145049e-001

1.5092743985123490e-008

Copyright (c) 2013, P. Lutus — <http://arachnoid.com>. All Rights Reserved.

Závislost je a není lineární. U malých hodnot bohužel hodnoty „ulítávají“ a kdo ví, jak je Microsoft aproximuje. Každopádně pro dostatečnou přesnost není potřeba užít polynom, korelační koeficient to nijak zvlášť neovlivní, jen holt data prokládáme přímkou, která (nelogicky) neprochází počátkem souřadného systému. Proč jsem nepoužil regresi dostupnou přímo Excelu? Trochu lenost použít funkce listu a koeficienty odečtené z rovnice v grafu nejsou dostatečně přesné (korelační koeficient je zaokrouhlen na 1, i když je chybovost podstatná). I přesto, že odchylka korelačního koeficientu od hodnoty 1 je až na 6 desetinném místě, přeci jen u sady sloupců s velmi malou šířkou již není možné zajistit původní celkovou šířku tabulky bez viditelné odchylky. Ale to je prostě daň. A byl tu další problém. Jak víme, Excel umí přidávat sloupce pouze vlevo od výběru. U posledního sloupce tak musíme řešit kupříkladu ohraničení přidaného sloupce, pokud nechceme v algoritmu udělat výjimku. V 95 % případů makro funguje. Občas se mi bohužel stalo, že Excel nenačetl barvu pravého ohraničení a namísto toho použil barvu černou (hodnota 0). Popravdě řečeno neměl jsem již sílu ošetřit těch 5 % případů, ani zjistit příčinu chování (chybu v algoritmu) a zjednat nápravu.

Slučování po řádcích přes všechny sloupce výběru

Jako bonus přikládám kód, který slučuje buňky výběru v každém řádku, a to přes všechny sloupce výběru (podle odhadu je to má druhá nejvyužívanější obecná procedura). Makro uvedené výše již funkčnost obsahuje, ale leckdy vám těch pár řádků zkrátí dobu otravného ručního „znovuslučování“ přes přidaný sloupec.

```
Sub SloucitPoRadcich()
```

```
    Dim rngRadek As Range
```

```
    Application.ScreenUpdating = False
```

```
    Application.Calculation = xlCalculationManual
```

```
    For Each rngRadek In Selection.Rows
```

```
        rngRadek.MergeCells = True
```

```
    Next rngRadek
```

```
    Application.Calculation = xlCalculationAutomatic
```

```
Application.ScreenUpdating = True
```

End Sub

Sešit ke stažení:

[uprava_rastru.zip](#)

[Uložit jako PDF](#)

