

Řečeno slovní hříčkou, pojmy uvedené v titulku mají společného „všechno a nic“. Každý z nich je vyjádřením stavu prázdné proměnné, a to podle datového typu či kontextu.

Empty

- vyjadřuje neinicializovanou proměnnou typu Variant (její výchozí hodnotu)
- ve VBA klíčové slovo Empty, funkcí IsEmpty testujeme například prázdnou buňku
- ve VBA funkce VarType vrací 0 (vbEmpty)
- dle kontextu je za Empty považována jak nula, tak prázdný řetězec

1	<code>Sub TestEmpty()</code>
2	
3	<code>Dim a, x</code>
4	
5	<code>a = 3</code>
6	
7	<code>'správně</code>
8	<code>a = Empty</code>
9	
10	<code>'špatně</code>
11	<code>'a = vbEmpty</code>
12	<code>'vbEmpty ... 0 (datový typ Long)</code>
13	<code>'běžně jen pro testování VarType</code>
14	
15	<code>'0</code>
16	<code>x = VarType(a)</code>
17	
18	<code>'True</code>
19	<code>x = VarType(a) = vbEmpty</code>
20	
21	<code>'Allen Browne:</code>
22	<code>'obojí True</code>
23	<code>Debug.Print Empty = 0</code>
24	<code>Debug.Print Empty = ""</code>
25	
26	<code>'-1 (True)</code>
27	<code>x = Not a</code>
28	
29	<code>'1</code>
30	<code>x = a ^ 0</code>
31	
32	<code>'0</code>
33	<code>x = a And True</code>
34	
35	<code>'test na prázdnou buňku, True</code>
36	<code>x = ActiveCell = Empty</code>
37	<code>'lépe, True</code>
38	<code>x = IsEmpty(ActiveCell)</code>
39	
40	<code>End Sub</code>

Null (no valid data, C NULL)

<http://cs.wikipedia.org/wiki/NULL>

- indikuje neznámou hodnotu pro datový typ Variant
- v databázi položka pole, na níž se nevyskytují žádná data
- ve VBA klíčové slovo Null, testujeme prostřednictvím funkce IsNull
- ve VBA funkce VarType vrátí 1 (vbNull)
- ve VBA Chr(0), resp. vbNullChar používán jako ukončovací znak řetězce (C NULL v programovacím jazyce C, typicky API funkce), na listu nelze zadat funkci ZNAK(0)

```

1 Sub TestNull()
2
3     'http://cs.wikipedia.org/wiki/NULL
4
5     Dim x, y, z
6
7     'špatně
8     'z = vbNull
9     'vbNull ... 1 (datový typ Long)
10    'běžně jen pro testování VarType
11
12    'znak Null (datový typ String)
13    'odpovídá NULL v jazyce C
14    z = Chr(0)
15    'ekvivalent
16    z = vbNullChar
17    '1
18    x = Len(z)
19
20    'správně
21    z = Null
22
23    '1
24    x = VarType(z)
25
26    'True
27    x = VarType(z) = vbNull
28
29    'Null
30    x = 1 + z
31
32    'y
33    x = "y" & z
34
35    'False
36    x = False And z
37
38    'Null v porovnání s Null nevrací True
39    x = z = z
40
41    'True
42    x = IsNull(z)
43
44    'bez chyby, ale pravděpodobně bez účinku
45    ActiveCell = Null
46    'podobně
47    'ActiveCell = Chr(0)
48    'na listu nelze zadat ZNAK(0)
49
50    Dim aPole(1 To 1)
51    aPole(1) = Null
52    'funkce listu TRANSPOZICE neumí Null převádět
53    ActiveCell = WorksheetFunction.Transpose(aPole)
54
55    'chybný test na prázdnou buňku, False
56    x = IsNull(ActiveCell)
57
58    'správný test na prázdnou buňku, True
59    x = IsEmpty(ActiveCell)
60
61 End Sub

```

Nothing

- vyjadřuje neinicializovanou (nepřirazenou) objektovou proměnnou (její výchozí hodnotu)
- platí pro proměnné deklarované jako As Object, Workbook, Worksheet, Range, UserForm, Connection, ...
- ve VBA ruší vazbu (asociaci) objektové proměnné na aktuální objekt (Set rngBunka = Nothing), čímž také uvolňuje systémové zdroje (alokovanou paměť)
- testujeme operátorem Is (rngBunka Is Nothing), tj. VBA nezná funkci IsNothing

```
1 Sub TestNothing()  
2  
3 Dim rngBunka As Range  
4  
5 Dim blnRes As Boolean  
6  
7 'True  
8 blnRes = rngBunka Is Nothing  
9  
10 'přirazení buňky A1 aktivního listu  
11 'do objektové proměnné (typu Range)  
12 Set rngBunka = Range("A1")  
13  
14 'False  
15 blnRes = rngBunka Is Nothing  
16  
17 'ruší vazbu objektové proměnné  
18 'na buňku A1 aktivního listu  
19 Set rngBunka = Nothing  
20  
21 'True  
22 blnRes = rngBunka Is Nothing  
23  
24  
25 'Allen Browne:  
26 'datový typ Variant/Empty  
27 Dim varPromenna As Variant  
28  
29 'nelze  
30 'x = varPromenna Is Nothing  
31  
32 'dojde k přetypování na Variant/Object  
33 Set varPromenna = Nothing  
34  
35 'True  
36 x = varPromenna Is Nothing  
37  
38 End Sub
```

Missing

- ve VBA funkce IsMissing ověřuje, zda-li byl proceduře (funkci) předán argument (parametr musí být typu Variant, jinak budeme nuceni testovat výchozí hodnotu daného datového typu)

```
1 Sub TestMissing()  
2  
3 Call VlastniFunkce1  
4  
5 Call VlastniFunkce2  
6 Call VlastniFunkce2("")  
7  
8 End Sub  
9  
10 Function VlastniFunkce1(Optional varPromenna As Variant)  
11  
12 'při volání Call VlastniFunkce  
13 'je proměnná varPromenna typu Variant/Error  
14 'a obsahuje hodnotu Missing  
15  
16 Dim blnRes As Boolean  
17  
18 'testovat na hodnotu Missing  
19 'lze jen proměnnu typu Variant  
20 blnRes = IsMissing(varPromenna)  
21  
22 'nebo  
23 blnRes = IsError(varPromenna)  
24  
25 End Function  
26  
27 Function VlastniFunkce2(Optional varPromenna As String)  
28  
29 'při volání Call VlastniFunkce2  
30 'je proměnná varPromenna typu String  
31 'a obsahuje prázdný řetězec coby výchozí hodnotu  
32  
33 Dim blnRes As Boolean  
34  
35 'nelze testovat na Missing  
36 'blnRes = IsMissing(varPromenna)  
37  
38 'True  
39 blnRes = varPromenna = vbNullString  
40  
41 End Function
```

Prázdný řetězec (null string, zero-length string, empty string)

http://en.wikipedia.org/wiki/Empty_string

- řetězec nulové délky, datový typ String neobsahující žádný znak
- nemá zastoupení v ASCII tabulce
- je-li ve VBA (nebo na listu) vyjadřován dvojicí uvozovek jdoucích bezprostředně za sebou (tj. „“, zero-length string, empty string), pak v paměti využívá 6 bitů a ukazatel na řetězec v paměti StrPtr(„“) = číslo
- ve VBA konstanta vbNullString (null string, null pointer, rychlejší při porovnávání než „“), nezabírá v paměti místo, ukazatel na řetězec v paměti StrPtr(vbNullString) = 0
- i když prázdná buňka obsahuje ve své vlastnosti Text prázdný řetězec, funkce listu JE.TEXT u ní narozdíl od skutečně obsaženého prázdného řetězce vrací NEPRAVDA

1	Sub TestPrazdnyRetezec()
2	
3	Dim a, b
4	
5	a = ""
6	b = vbNullString
7	
8	'rozlišení "" a vbNullString
9	
10	'číslo...
11	x = StrPtr(a)
12	
13	'0 (datový typ Long)
14	x = StrPtr(b)
15	
16	'test na prázdný řetězec
17	
18	'True
19	x = a = ""
20	
21	'lépe
22	'True
23	x = a = vbNullString
24	
25	'nejlépe
26	'True
27	x = Len(a) = 0
28	
29	End Sub

Co je to ta inicializace?

Inicializace představuje explicitní přiřazení počáteční hodnoty do (deklarované) proměnné.

```
1 Sub PrikladDeklaraceInicializace()  
2  
3 'VBA  
4  
5 'deklarace proměnné s názvem intCislo datového typu  
6 'Integer, tj. celé číslo v rozsahu -32 768 až 32 767  
7 'vyžadující 2 bajty v paměti  
8 Dim intCislo As Integer  
9  
10 'pozn. přidělování paměťového prostoru proměnným  
11 'se říká alokace paměti  
12  
13 'již v tuto chvíli proměnná intCislo obsahuje nepřímo  
14 'definovanou (implicitní) hodnotu, což je pro datový typ  
15 'Integer nula  
16  
17 'inicializace, tj. přiřazení počáteční hodnoty přímo  
18 '(explicitně)  
19 intCislo = 5  
20  
21 'deklaraci a inicializaci současně lze ve VBA provést  
22 'pouze u konstant nebo volitelných parametrů procedur (funkcí)  
23  
24 'konstanta  
25 Const dblPi As Double = 3.14159  
26  
27 'VB.NET  
28  
29 'deklaraci i inicializaci lze i u proměnných  
30 'provádět současně  
31 'Dim intCislo As Integer = 5  
32  
33 End Sub
```

V souladu s teorií je tedy za tu pravou inicializaci považováno přiřazení `intCislo = 5`. Nicméně podle mého názoru prvotní inicializaci (nastavení výchozích hodnot) u jednoduchých datových typů provádí Excel už v rámci „továrního nastavení“. tj. bezprostředně po deklaraci. U objektových proměnných je inicializace čistě na nás (a stejně tak inicializace třídy).

A jaké jsou výchozí hodnoty pro jednoduché datové typy?

```
1 Sub VychodziHodnotyDatovychTypu()
2
3     'hodnota (datový typ/podtyp)
4
5     'Empty (Variant/Empty)
6     Dim a 'totéž jako As Variant
7
8     '0 (Integer)
9     Dim b As Integer
10
11     '0 (Double)
12     Dim c As Double
13
14     'vbNullString, konstanta pro prázdný řetězec (String)
15     Dim d As String
16
17     'Chr(0), znak pro prázdný řetězec (String)
18     Dim e As String * 255
19
20     'Nothing (Object)
21     Dim f As Object
22
23     'False (Boolean)
24     Dim g As Boolean
25
26     '? (Variant(), tj. pole budoucích hodnot typu Variant)
27     Dim h()
28
29     '? (String(), tj. pole budoucích hodnot typu String)
30     Dim i() As String
31
32 End Sub
```

Tip 1

Je možné snadno převést Null, Empty a prázdný řetězec na nulu? Ano.

```
1 Sub NulaZNiceho()
2
3     Dim a, b, c, d
4
5     a = Null
6     b = vbNullString
7     c = Empty
8
9     'univerzální převod, výsledkem 0
10    d = Val("0" & a)
11    e = Val("0" & b)
12    f = Val("0" & c)
13
14 End Sub
```


Tip 2

Je možné snadno testovat, zda-li je prázdné pole? Ano.

```
1 Sub TestyPole()
2
3 Dim Pole1(1 To 3)
4 Dim Pole2()
5
6 'True
7 boolPole1JePrazdne = Len(Join(Pole1, "")) = 0
8
9 Pole1(2) = "Liberec"
10 Pole1(3) = 256
11
12 'False
13 boolPole1JePrazdne = Len(Join(Pole1, "")) = 0
14
15 'True
16 boolPole2JePrazdne = (Not Pole2) = True
17
18 'Array nebo ReDim
19 Pole2 = Array(vbNullString, vbNullString)
20
21 'False
22 boolPole2JePrazdne = (Not Pole2) = True
23
24 End Sub
```

Je to vše?

Bohužel, ani náhodou. Hlavní „války o nic“ se odehrávají na listu Excelu. Do bitev vstupuje skrytí buňky přímo či filtrem, formát buňky pro zneviditelnění hodnoty (vynechání sekce po sobě jdoucími středníky), skrývání nul a další lahůdky související s XML (viz třeba článek [Záhada zmrazené buňky](#)).

Prázdná buňka

Její definici jsem čas od času nucen sám přehodnotit v souvislosti s novými poznatky. V souladu s tím, co bylo napsáno výše, se obsah skutečně prázdné buňky chová jako Variant/Empty. Podle kontextu je brána jako nula (běžný výsledek prostého odkazu na buňku), prázdný řetězec nebo NEPRAVDA. Přesto vás výsledky některých funkcí listu jistě zaskočí.

Popis dat	Data									
		.Value (VBA)	.Value2 (VBA)	.Text (VBA)	TYP	POLÍČKO("Formát")	Číslo (JE.ČÍSLO)	Číslo (POČET)	Logická (JE.LOGIČNĚ)	Textová (JE.TEXT)
skutečně prázdná buňka		Empty	Empty	"" (String)	1	G	NEPRAVDA	0	NEPRAVDA	NEPRAVDA
nula	0	0 (Double)	0 (Double)	"0" (String)	1	G	PRAVDA	1	NEPRAVDA	NEPRAVDA
skrývaná nula (Soubor → Možnosti → Upřesnit...)	0	0 (Double)	0 (Double)	"0" (String)	1	G	PRAVDA	1	NEPRAVDA	NEPRAVDA
číslo 45 s formátem ;;;		45 (Double)	45 (Double)	"" (String)	1	H	PRAVDA	1	NEPRAVDA	NEPRAVDA
text "Jana" s formátem ;;;		Jana (String)	Jana (String)	"" (String)	2	H	NEPRAVDA	0	NEPRAVDA	PRAVDA
prázdný řetězec vzorcem =""		"" (String)	"" (String)	"" (String)	2	G	NEPRAVDA	0	NEPRAVDA	PRAVDA
prázdný řetězec převedený na hodnotu		"" (String)	"" (String)	"" (String)	2	G	NEPRAVDA	0	NEPRAVDA	PRAVDA
apostrof		"" (String)	"" (String)	"" (String)	2	G	NEPRAVDA	0	NEPRAVDA	PRAVDA

Prázdná buňka a funkce listu – 1. část

Popis dat	Data											
		Textová (COUNTIF(*))	Netextová (JE.NETEXT)	Neprázdná (POČET2)	Prázdná (JE.PRAZDNE)	Prázdná (POLÍČKO("type"))	Prázdná (COUNTBLANK)	DĚLKA	HODNOTA.NA.TEXT("@")	KÓD	N	T
skutečně prázdná buňka		0	PRAVDA	0	PRAVDA	b	1	0	0	#HODNOTA!	0	
nula	0	0	PRAVDA	1	NEPRAVDA	v	0	1	0	48	0	
skrývaná nula (Soubor → Možnosti → Upřesnit...)	0	0	PRAVDA	1	NEPRAVDA	v	0	1	0	48	0	
číslo 45 s formátem ;;;		0	PRAVDA	1	NEPRAVDA	v	0	2	45	52	45	
text "Jana" s formátem ;;;		1	NEPRAVDA	1	NEPRAVDA	l	0	4	Jana	74	0	Jana
prázdný řetězec vzorcem =""		1	NEPRAVDA	1	NEPRAVDA	l	1	0		#HODNOTA!	0	
prázdný řetězec převedený na hodnotu		1	NEPRAVDA	1	NEPRAVDA	l	1	0		#HODNOTA!	0	
apostrof		1	NEPRAVDA	1	NEPRAVDA	l	1	0		#HODNOTA!	0	

Prázdná buňka a funkce listu – 2. část

C10		fx =A(B2;PRAVDA)	
	A	B	C
1		Prázdná buňka	Test
2			0 =B2
3			PRAVDA =NE(B2)
4			=B2&""
5			#ČÍSLO! =B2^0
6			#ČÍSLO! =0^B2
7			=T(B2)
8			0 =N(B2)
9			0 =DĚLKA(B2)
10			PRAVDA =A(B2;PRAVDA)
11			#HODNOTA! =A(B2;B2)
12			PRAVDA =NEBO(B2;PRAVDA)
13			1 =B2+PRAVDA
14			0 =B2*PRAVDA
15			NEPRAVDA =B2=PRAVDA
16			PRAVDA =B2=NEPRAVDA
17			PRAVDA =B2=0
18			PRAVDA =B2=""
19			NEPRAVDA =JE.ČÍSLO(B2)
20			NEPRAVDA =JE.TEXT(B2)
21			PRAVDA =JE.NETEXT(B2)
22			PRAVDA =ISEVEN(B2)
23			NEPRAVDA =ISODD(B2)
24			
25			

Prázdná buňka a funkce listu - 3. část

U prázdné buňky převažuje numerické chování nad textovým. To je bohužel nepříjemné. Z odkazu ve vzorci obdržíme nulu, čímž Excel smaže rozdíl mezi „mít nulu“ a „nemít nic“. To v praxi znamená nerozlišovat mezi „nemít založen v bance účet“ a „nemít na existujícím účtu ani korunu“. Nebo - a to je horší - neprovádět měření versus naměřit nulovou odchylku. Do třetice příklad vypůjčený spíš z oblasti databází - odpojený elektroměr pana Kropáčka (potřebujeme přitom zachovat záznam o tom, že byl klientem naší společnosti), a paní Šebestová, která za daný měsíc neodebrala elektřinu, protože je momentálně s přítelem Machem v zahraničí.