Funkce pro práci s obrazovkou

Tato část popisuje běhové funkce používané k vyvolávání dialogů pro vstup a výstup uživatelských dat.

Funkce pro zobrazování

Tato část popisuje funkce používané pro výstup informací na obrazovku.

Funkce pro vstup na obrazovku

Tato část popisuje funkce používané pro vstup na obrazovku.

Funkce pro práci s barvami

Tato část popisuje funkce používané k určení barev.

Příkaz MsgBox [Runtime]

Zobrazí dialogové okno obsahující zprávu.

Syntaxe:

```
MsgBox Text As String [,Typ As Integer [,Název_dialogového_okna As String]] (jako
příkaz) nebo MsgBox (Text As String [,Typ As Integer [,Název_dialogového_okna As
String]]) (jako funkce)
```

Parametr:

Text: Řetězec zobrazený jako zpráva v dialogovém okně. Zalomení řádku lze vložit pomocí Chr$(13).

Název_dialogového_okna: Řetězec zobrazený v titulní liště dialogu. Je-li vynechán, zobrazí se v titulní liště název odpovídající aplikace.

Typ: Jakýkoliv celočíselný výraz, který určuje druh dialogu a také počet a druh zobrazených tlačítek a druh ikony. **Typ** představuje kombinaci bitových vzorků, takže lze použít kombinaci prvků součtem jejich hodnot:

0: Zobrazí se pouze tlačítko OK.

1: Zobrazí se tlačítka OK a Zrušit.

2: Zobrazí se tlačítka Zrušit, Opakovat a Ignorovat.

3: Zobrazí se tlačítka Ano, Ne a Zrušit.

4: Zobrazí se tlačítka Ano a Ne.

5: Zobrazí se tlačítka Opakovat a Zrušit.

16: Do dialogového okna je přidána ikona Zastavit.

32: Do dialogového okna je přidána ikona Otazník.

48: Do dialogového okna je přidána ikona Vykřičník.

64: Do dialogového okna je přidána ikona Informace.

128: První tlačítko v dialogu je použito jako výchozí tlačítko.

256: Druhé tlačítko v dialogovém okně je použito jako výchozí tlačítko.

512: Třetí tlačítko v dialogovém okně je použito jako výchozí tlačítko.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleMsgBox
Const sText1 = "Došlo k neočekávané chybě."
Const sText2 = "Program ale bude pokračovat."
Const sText3 = "Chyba"
MsgBox(sText1 + Chr(13) + sText2,16,sText3)
End sub
```

GlobalScope [Runtime]

Zdrojový kód a dialogy jazyka Basic jsou uspořádány v systému knihoven.

- LibraryContainer obsahuje knihovny.
- Knihovny mohou obsahovat moduly a dialogy

V jazyce Basic:

- LibraryContainer se nazývá **BasicLibraries**.

V dialozích:

- LibraryContainer se nazývá **DialogLibraries**.

LibraryContainer existuje na úrovni aplikace i v každém dokumentu. V dokumentu se LibraryContainer volá automaticky. Pokud chcete volat globální LibraryContainer z dokumentu, musíte použít klíčové slovo **GlobalScope**.

Syntaxe:

GlobalScope

Příklad:

Příklad v dokumentu Basic

' volání Dialog1 v knihovně dokumentu Standard

oDlgDesc = DialogLibraries.Standard.Dialog1

' volání Dialog2 v aplikační knihovně Library1

oDlgDesc = GlobalScope.DialogLibraries.Library1.Dialog2

Funkce MsgBox [Runtime]

Zobrazí dialogové okno obsahující zprávu a vrátí hodnotu.

Syntaxe:

```
MsgBox (Text As String [,Typ As Integer [,TitulekDialogu As String]])
```

Návratová hodnota:

Celé číslo

Parametr:

Text: Řetězec zobrazený jako zpráva v dialogovém okně. Zalomení řádku lze vložit pomocí Chr$(13).

TitulekDialogu: Řetězec zobrazený v titulní liště dialogu. Je-li vynechán, zobrazí se v titulní liště název odpovídající aplikace.

Typ: Jakýkoliv celočíselný výraz, který určuje druh dialogu a také počet a druh zobrazených tlačítek a druh ikony. **Typ** představuje kombinaci bitových vzorků, takže lze použít kombinaci prvků součtem jejich hodnot:

Hodnoty

0: Zobrazí se pouze tlačítko OK.

1: Zobrazí se tlačítka OK a Zrušit.

2: Zobrazí se tlačítka Zrušit, Opakovat a Ignorovat.

3 : Zobrazí se tlačítka Ano, Ne a Zrušit.

4: Zobrazí se tlačítka Ano a Ne.

5: Zobrazí se tlačítka Opakovat a Zrušit.

16: Do dialogového okna je přidána ikona Zastavit.

32: Do dialogového okna je přidána ikona Otazník.

48 : Přidá do dialogu ikonu vykřičníku.

64: Do dialogového okna je přidána ikona Informace.

128: První tlačítko v dialogu je použito jako výchozí tlačítko.

256: Druhé tlačítko v dialogovém okně je použito jako výchozí tlačítko.

512: Třetí tlačítko v dialogovém okně je použito jako výchozí tlačítko.

Návratová hodnota:

1 : OK

2 : Zrušit

3 : Zrušit

4 : Opakovat

5 : Ignorovat

6 : Ano

7 : Ne

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleMsgBox
Dim sVar as Integer
sVar = MsgBox("Las Vegas")
sVar = MsgBox("Las Vegas",1)
sVar = MsgBox( "Las Vegas",256 + 16 + 2,"Dialog title")
end sub
```

Příkaz Print [Runtime]

Vypíše zadané řetězce nebo číselné výrazy v dialogovém okně nebo do souboru.

Syntaxe:

```
Print [#JménoSouboru,]Výraz1[{;|,} [Spc(Číslo As Integer);] [Tab(pozice As Integer);]
[Výraz2[...]]
```

Parametr:

JménoSouboru: Jakýkoliv číselný výraz, který obsahuje číslo souboru nastavené příkazem Open pro daný soubor.

Výraz: Číselný nebo řetězcový výraz, který se má zobrazit. Více výrazů lze oddělit středníkem. Pokud jsou odděleny čárkou, zarovnají se výrazy na další tabulátor. Tabulátory nelze upravovat.

Číslo: Počet mezer, které vloží funkce **Spc**.

Pozice: Mezery se vkládají až do určité pozice.

Pokud po posledním výrazu následuje středník nebo čárka, OpenOffice.org Basic uloží text v interním bufferu a pokračuje v běhu programu bez výpisu. Až narazí na další příkaz Print bez středníku nebo čárky na konci, vytiskne celý text najednou.

Kladné číselné výrazy jsou vytištěny s uvozující mezerou. Záporné číselné výrazy jsou vytištěny s uvozujícím znaménkem minus. Pokud hodnoty s plovoucí desetinnou čárkou přesahují určitou oblast, je příslušný číselný výraz vytištěn v exponenciálním tvaru.

Přesahuje-li výraz, který má být vytištěn, určitou délku, bude zobrazení automaticky zalomeno na následující řádek.

Je možné vložit funkci Tab, uzavřenou středníky, mezi argumenty, pokud chcete výstup zarovnat na určitou pozici, nebo je možné použít funkci **Spc** ke vložení určitého počtu mezer.

Příklad:

```
Sub ExamplePrint
Print "ABC"
Print "ABC","123"
i = FreeFile()
Open "C:\Temp.txt" For Output As i
Print #i, "ABC"
Close #i
end Sub
```

ThisComponent [Runtime]

Adresuje aktivní komponentu, aby bylo možné číst a nastavovat její vlastnosti. ThisComponent se používá v dokumentu Basic, kde představuej dokument, do kterého Basic patří. Typ objektu zpřístupněného přes ThisComponent závisí na typu dokumentu.

Syntaxe:

ThisComponent

Příklad:

```
Sub Main
REM Aktualizuje "Obsah" v textovém dokumentu
Dim allindexes, index As Object
allindexes = ThisComponent.getDocumentIndexes()
index = allindexes.getByName("Obsah1")
REM používá výchozí název pro Obsah a 1
index.update()
End Sub
```

Funkce GetDefaultContext [Runtime]

Vrátí výchozí kontext process service factory, pokud existuje, jinak vrátí null.

Tato funkce vrátí výchozí použitý kontext, pokud k službám přistupujete pomocí XmultiServiceFactory. Více informací získáte v kapitole `Professional UNO` v knize `Developer's Guide` na adrese api.openoffice.org.

Funkce InputBox [Runtime]

Zobrazí dialogové okno s výzvou umožňující uživateli zadat text do pole. Vstup je přiřazen proměnné.

Příkaz **InputBox** je vhodná metoda pro zadávání textu v dialogu. Vstup potvrdíte klepnutím na OK nebo stiskem klávesy Enter. Vstup získáte jako návratovou hodnotu funkce. Pokud zavřete dialog pomocí Zrušit, **InputBox** vrátí řetězec nulové délky ("").

Syntaxe:

```
InputBox (Zpráva As String[, Titulek As String[, Výchozí As String[, Pozice_x As
Integer, Pozice_y As Integer]]])
```

Návratová hodnota:

Řetězec

Parametr:

Zpráva: Řetězec, který se zobrazí jako zpráva v dialogovém okně.

Titulek: Řetězec, který se zobrazí v titulní liště dialogu.

Výchozí: Řetězec, který se zobrazí v textovém poli jako výchozí, pokud není zadán vstup.

Pozice_x: Celočíselný výraz, který určuje vodorovné umístění dialogu. Pozice je absolutní souřadnice a nevztahuje se k oknu aplikace.

Pozice_y: Celočíselný výraz, který určuje svislé umístění dialogu. Pozice je absolutní souřadnice a nevztahuje se k oknu aplikace.

Pokud jsou **Pozice_x** a **Pozice_y** vynechány, dialog se na obrazovce vystředí. Pozice se určuje v twipech.

Příklad:

```
Sub ExampleInputBox
Dim sText As String
sText = InputBox ("Zadej heslo:","Drahý uživateli")
MsgBox ( sText , 64, "Opakuj heslo")
End Sub
```

Funkce pro práci s barvami

Tato část popisuje funkce používané k určení barev.

Funkce Blue [Runtime]

Vrací hodnotu modré složky pro zadaný kód barvy.

Funkce Green [Runtime]

Vrací hodnotu zelené složky pro zadaný kód barvy.

Funkce Red [Runtime]

Vrací hodnotu červené složky pro zadaný kód barvy.

Funkce QBColor [Runtime]

Vrátí RGB kód barvy pro barvu zadanou hodnotou ze starších systémů MS-DOS.

Funkce RGB [Runtime]

Vrací celočíselnou hodnotu barvy sestávající z modré, červené a zelené složky.

Funkce CreateObject [Runtime]

Vytvoří UNO objekt. Na Windows může vytvořit také OLE objekty.
Tato metoda vytvoří instanci typu, který je předán jako parametr.

Syntaxe:

oObj = CreateObject(type)

Příklad:

```
Type address
Name1 as String
City as String
End Type
Sub main
myaddress = CreateObject("address")
MsgBox IsObject(myaddress)
End Sub
```

Funkce CreateUnoValue [Runtime]

Vrátí objekt, který představuje striktně typovou hodnotu odkazující na typ Uno.

Tento objekt se automaticky při předání do Uno převede na odpovídající typ. Typ musí být určen plně kvalifikovaným názvem typu Uno.

OpenOffice.org API často používá typ Any. Je opakem typu Variant známého z jiných prostředí. Typ Any obsahuje jeden určený typ Uno a používají jej obecná Uno rozhraní.

Syntaxe:

oUnoValue = CreateUnoValue("[]byte", MyBasicValue) pro získání sekvence bajtů.

Pokud nelze CreateUnoValue převést na určený typ Uno, dojde k chybě. Pro převod se používá služba TypeConverter.

Tato funkce je určena pro použití v situacích, kdy nepostačuje základní mechanismus převodu z typu Basic na typ Uno. K tomu dojde, pokud se z OpenOffice.org Basic snažíte přistupovat k obecnému Any založenému na rozhraních, jako např. XPropertySet::setPropertyValue(Name, Value) nebo X???Container::insertBy???(???, Value). Basic tyto typy nerozezná, jelikož jsou definovány jen v příslušené službě.

V takové situaci OpenOffice.org Basic vybere nejlepší odpovídající typ pro typ Basic, který chcete převést. Ovšem pokud typ vybere špatně, dojde k chybě. Proto použijete funkci CreateUnoValue() pro vytvoření hodnoty neznámého typu Uno.

Tuto funkci je také možné použít pro předávání hodnot jiných typů než Any. To však nedoporučujeme. Pokud jazyk Basic již zná cílový typ, použitím funkce CreateUnoValue() jen vyvoláte další převodní operaci, která zpomalí běh programu.

Funkce Blue [Runtime]

Vrací hodnotu modré složky pro zadaný kód barvy.

Syntaxe:

```
Blue (Barva As Long)
```

Návratová hodnota:

Celé číslo

Parametr:

Barva: Celočíselný výraz určující jakýkoliv kód barvy, jehož modrou složku chcete vrátit.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleColor
Dim lVar As Long
lVar = rgb(128,0,200)
MsgBox "Barva " & lVar & " se skládá z:" & Chr(13) &_
"červená= " & Red(lVar) & Chr(13)&_
"zelená= " & Green(lVar) & Chr(13)&_
"modrá= " & Blue(lVar) & Chr(13) , 64,"barvy"
End Sub
```

Funkce Green [Runtime]

Vrací hodnotu zelené složky pro zadaný kód barvy.

Syntaxe:

```
Green (Barva As Long)
```

Návratová hodnota:

Celé číslo

Parametr:

Barva: Celočíselný výraz určující jakýkoliv kód barvy, jehož zelenou složku chcete vrátit.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleColor
Dim lVar As Long
lVar = rgb(128,0,200)
msgbox "The color " & lVar & " contains the components:" & Chr(13) &_
"red = " & red(lVar) & Chr(13)&_
"green = " & green(lVar) & Chr(13)&_
"blue = " & blue(lVar) & Chr(13) , 64,"barvy"
end sub
```

```
Sub ContListener_elementRemoved( oEvent )
MsgBox "elementRemoved"
MsgBox oEvent.Dbg_Properties
End Sub
Sub ContListener_elementReplaced( oEvent )
MsgBox "elementReplaced"
MsgBox oEvent.Dbg_Properties
End Sub
```

Pokud objekt nepoužíváte, nemusíte přidávat jeho parametr:

```
' Minimální implementace Sub disposing
Sub ContListener_disposing
End Sub
```

Metody Listeneru musíte **vždy** implementovat, abyste předešli chybám Basic.

Funkce CreateUnoListener [Runtime]

Vytvoří instanci Listener.

Mnoho UNO rozhraní vám umožní registrovat listener ve speciálním rozhraní. Tak můžete naslouchat určitým událostem a volat při nich odpovídající naslouchací metodu. Funkce CreateUnoListener čeká v odpovídajícím rozhraní a poté rozhraní předá objektu, který toto rozhraní podporuje. Tento objekt bude předán zaregistrované metodě.

Syntaxe:

oListener = CreateUnoListener(Prefixname, ListenerInterfaceName)

Příklad:

Následující příklad je založen na knihovně objektů Basic.

Dim oListener

oListener = CreateUnoListener("ContListener_","com.sun.star.container.XContainerListener")

Metoda CreateUnoListener vyžaduje dva parametry. První je prefix a podrobněji je vysvětlen níže. Druhý parametr je plně kvalifikovaný název rozhraní Listener, které chcete použít.

Listener poté musíte přidat k Broadcaster Object. To provedete zavoláním odpovídající metody pro přidání Listeneru. Tyto metody se vždy nazývají podle vzoru "addFooListener", kde "Foo" představuje typ rozhraní Listeneru bez 'X'. V tomto příkladu se volá metoda addContainerListener pro zaregistrování XContainerListener:

Dim oLib

oLib = BasicLibraries.Library1 ' Library1 musí existovat!

oLib.addContainerListener(oListener) ' Registerovat listener

Nyní je Listener zaregistrován. Když nastane událost, zavolá odpovídající Listener správnou metodu z rozhraní com.sun.star.container.XContainerListener.

Prefix volá registrované Listenery z podprogramů Basic. Za běhu Basic hledá procedury nebo funkce, které mají jméno "PrefixListenerMethode" a při nalezení je zavolá. Jinak dojde k chybě.

V tomto případě používá rozhraní Listener následující metody:

- disposing:
- Listener base interface (com.sun.star.lang.XEventListener): základní rozhraní pro všechna rozhraní Listener
- elementInserted:
- Metoda rozhraní the com.sun.star.container.XContainerListener
- elementRemoved:
- Metoda rozhraní the com.sun.star.container.XContainerListener
- elementReplaced:
- Metoda rozhraní the com.sun.star.container.XContainerListener

V tomto případě je prefixem 'ContListener_'. V jazyce OpenOffice.org Basic je proto třeba implementovat následující podprogramy:

- ContListener_disposing
- ContListener_elementInserted
- ContListener_elementRemoved
- ContListener_elementReplaced

Pro každý typ Listeneru existuje typ event structure obsahující informace o události. Když je zavolána metoda Listeneru, předá se metodě jako parametr instance této události. Metody Listeneru také mohou volat tyto objekty události, je-li v deklaraci Sub předán správný parametr. Například:

Sub ContListener_disposing(oEvent)

MsgBox "disposing"

MsgBox oEvent.Dbg_Properties

End Sub

Sub ContListener_elementInserted(oEvent)

MsgBox "elementInserted"

MsgBox oEvent.Dbg_Properties

End Sub

Funkce Red [Runtime]

Vrací hodnotu červené složky pro zadaný kód barvy.

Syntaxe:

```
Red (Barva As Long)
```

Návratová hodnota:

Celé číslo

Parametr:

Barva: Celočíselný výraz určující jakýkoliv kód barvy, jehož červenou složku chcete vrátit.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleColor
Dim lVar As Long
lVar = rgb(128,0,200)
msgbox "Barva " & lVar & " se skládá z:" & Chr(13) &_
"červená= " & red(lVar) & Chr(13)&_
"zelená= " & green(lVar) & Chr(13)&_
"modrá= " & blue(lVar) & Chr(13) , 64,"barvy"
end sub
```

Funkce QBColor [Runtime]

Vrátí RGB kód barvy pro barvu zadanou hodnotou ze starších systémů MS-DOS.

Syntaxe:

```
QBColor (ČísloBarvy As Integer)
```

Návratová hodnota:

Typu Long

Parametr:

ČísloBarvy: Celočíselný výraz, který určuje hodnotu barvy ze starších systémů MS-DOS.

ČísloBarvy může mít následující hodnoty:

0 : Černá
1: Modrá
2: Zelená
3: Azurová
4: Červená
5: Purpurová
6: Žlutá
7: Bílá
8: Šedá
9: Světle modrá
10: Světle zelená
11: Světle azurová
12: Světle červená
13: Světle purpurová
14: Světle žlutá
15: Jasná bílá

Tato funkce se používá jen pro převod starších aplikací BASIC, které používají zmíněné barevné kódy. Funkce vrací celočíselnou hodnotu určující barvu použitou v OpenOffice.org IDE.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleQBColor
Dim iColor As Integer
Dim sText As String
iColor = 7
sText = "RGB= " & Red(QBColor( iColor) ) & ":" & Blue(QBColor( iColor) ) & ":" &
Green(QBColor( iColor) )
MsgBox stext,0,"Color " & iColor
End Sub
```

Funkce CreateUnoDialog [Runtime]

Vytvoří objekt typu Uno jazyka Basic, který představuje ovládací prvek dialogu typu Uno v průběhu programu jazyka Basic.

Dialogy jsou definovány v knihovnách dialogů. Chcete-li dialog zobrazit, je třeba vytvořit „živý“ dialog z knihovny.

Viz Příklady.

Syntaxe:

CreateUnoDialog(oDlgDesc)

Příklad:

```
' Get dialog description from the dialog library
oDlgDesc = DialogLibraries.Standard.Dialog1
' generate "live" dialog
oDlgControl = CreateUnoDialog( oDlgDesc )
' display "live" dialog
oDlgControl.execute
```

Funkce GetProcessServiceManager [Runtime]

Vrátí službu typu ProcessServiceManager (centrální ServiceManager pro Uno).
Tato funkce je potřebná, pokud chcete inicializovat službu pomocí CreateInstanceWithArguments.

Syntaxe:

oServiceManager = GetProcessServiceManager()

Příklad:

oServiceManager = GetProcessServiceManager()
oIntrospection = oServiceManager.createInstance("com.sun.star.beans.Introspection");
to je stejné jako následující příkaz:
oIntrospection = CreateUnoService("com.sun.star.beans.Introspection")

Informace

Národní nastavení použité pro formátování čísel, data a měny v jazyce OpenOffice.org Basic je možné nastavit v **Nástroje - Volby - Jazyková nastavení - Jazyky**. Ve formátovacích kódech Basic se jako desetinný oddělovač používá vždy tečka (.), která se při zobrazení nahradí odpovídajícím znakem podle národního nastavení.

Totéž platí pro národní nastavení formátu data, času a měny. Formátovací kódy se interpretují a zobrazí podle aktuálního národního nastavení.

Hodnoty pro 16 základních barev jsou následující:

Hodnota	**Jméno barvy**
0	Černá
128	Modrá
32768	Zelená
32896	Modrozelená
8388608	Červená
8388736	Purpurová
8421376	Žlutá
8421504	Bílá
12632256	Šedá
255	Světle modrá
65280	Světle zelená
65535	Světle azurová
16711680	Světle červená
16711935	Světle purpurová
16776960	Světle žlutá
16777215	Průhledná bílá

Chybové kódy

2 Nespecifikovaná syntaktická chyba
3 Return bez Gosub
4 Znovu od začátku
5 Neplatné volání procedury
6 Přetečení
7 Nedostatek paměti
8 Velikost pole již byla určena
9 Index mimo rozsah
10 Dvojitá definice
11 Dělení nulou
12 Nedefinovaná proměnná
13 Nesoulad typů
14 Neplatný parametr
18 Došlo k přerušení uživatelem
20 Pokračovat bez chyby
28 Nedostatek místa v zásobníku
35 Procedura nebo funkce není definována

48 Chyba při načítání knihovny DLL
49 Špatné volání DLL
51 Interní chyba
52 Špatný název nebo číslo souboru
53 Soubor nenalezen
54 Špatný režim souboru
55 Soubor je již otevřen
57 Vstupně-výstupní chyba zařízení
58 Soubor již existuje
59 Špatná délka záznamu
61 Disk plný
62 Zápis za konec souboru
63 Špatné číslo záznamu
67 Příliš mnoho otevřených souborů
68 Zařízení není dostupné
70 Přístup zamítnut
71 Disk není připraven
73 Vlastnost není implementována
74 Nelze přesunout na jiný disk
75 Chyba přístupu k adresáři/souboru
76 Adresář nenalezen
91 Proměnná objektu není nastavena
93 Neplatný vzor řetězce
94 Neplatné použití Null
323 Není možné načíst modul
341 Neplatný index objektu
366 Žádný aktivní pohled nebo dokument
380 Špatná hodnota vlastnosti
382 Vlastnost je pouze pro čtení
394 Vlastnost je pouze pro zápis
420 Neplatný odkaz na objekt
423 Vlastnost nebo metoda nenalezena
424 Je vyžadován objekt
425 Neplatné použití objektu
430 Třída nepodporuje OLE
438 Objekt nepodporuje metodu
440 Chyba OLE propojení
445 Objekt nepodporuje tuto akci
446 Objekt nepodporuje pojmenované argumenty
447 Objekt nepodporuje současné národní nastavení
448 Pojmenovaný argument nenalezen
449 Argument je povinný
450 Špatný počet argumentů
451 Objekt není kolekce
452 Neplatné pořadí
453 Určená DLL funkce nenalezena
460 Neplatný formát schránky

Funkce CreateUnoService [Runtime]

Inicializuje Uno službu s ProcessServiceManager.

Syntaxe:

oService = CreateUnoService(Název služby Uno)

Seznam dostupných služeb najdete na adrese http://api.openoffice.org/docs/common/ref/com/sun/star/module-ix.html

Příklady:

oIntrospection = CreateUnoService("com.sun.star.beans.Introspection")

Následující příklad používá službu k otevření dialogu pro výběr souboru:

```
Sub Main
fName = FileOpenDialog ("Vyberte, prosím, soubor")
print "vybraný soubor: "+fName
End Sub
function FileOpenDialog(title as String) as String
filepicker = createUnoService("com.sun.star.ui.dialogs.FilePicker")
filepicker.Title = title
filepicker.execute()
files = filepicker.getFiles()
FileOpenDialog=files(0)
End function
```

Funkce CreateUnoStruct [Runtime]

Vytvoří instanci typu Uno structure.
Použijte následující příkazy:
Dim oStruct as new com.sun.star.beans.Property

Syntaxe:

oStruct = CreateUnoStruct(název typu Uno)

Příklad:

oStruct = CreateUnoStruct("com.sun.star.beans.Property")

Funkce RGB [Runtime]

Vrací celočíselnou hodnotu barvy sestávající z modré, červené a zelené složky.

Syntaxe:

```
RGB (Red, Green, Blue)
```

Návratová hodnota:

Typu Long

Parametr:

Red: Celočíselný výraz reprezentující červenou složku (0-255) barvy.
Green: Celočíselný výraz reprezentující zelenou složku (0-255) barvy.
Blue: Celočíselný výraz reprezentující modrou složku (0-255) barvy.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleColor
Dim lVar As Long
lVar = rgb(128,0,200)
msgbox "Barva " & lVar & " se skládá z:" & Chr(13) &_
"červená= " & red(lVar) & Chr(13)&_
"zelená= " & green(lVar) & Chr(13)&_
"modrá= " & blue(lVar) & Chr(13) , 64,"barvy"
end sub
```

Funkce pro práci se soubory

S pomocí těchto funkcí je možné vytvářet a spravovat uživatelské soubory.

Tyto funkce je možné použít pro vytváření "relativních" souborů, takže je možné ukládat a načítat určité záznamy podle čísla záznamu. Souborové funkce vám také pomohou spravovat soubory a poskytují informace, jako např. velikost souboru, aktuální nastavení cesty, nebo datum vytvoření souboru či adresáře.

Otevření a zavření souborů

Funkce pro vstup/výstup do souborů

Správa souborů

Zde jsou popsány funkce a příkazy pro správu souborů.

Funkce TwipsPerPixelY [Runtime]

Vrátí počet twips, které představují výšku pixelu.

Syntaxe:

n = TwipsPerPixelY

Návratová hodnota:

Celé číslo

Příklad:

```
Sub ExamplePixelTwips
MsgBox "" & TwipsPerPixelX() & " Twips * " & TwipsPerPixelY() & " Twips",0,"Pixel size"
End Sub
```

Funkce TwipsPerPixelX [Runtime]

Vrátí počet twips, které představují šířku pixelu.

Syntaxe:

n = TwipsPerPixelX

Návratová hodnota:

Celé číslo

Příklad:

```
Sub ExamplePixelTwips
MsgBox "" & TwipsPerPixelX() & " Twips * " & TwipsPerPixelY() & " Twips",0,"Pixel size"
End Sub
```

Otevření a zavření souborů

Příkaz Close [Runtime]

Zavře určený soubor, který byl otevřen příkazem Open.

Funkce FreeFile [Runtime]

Vrátí další dostupné číslo souboru pro otevření souboru. Pomocí této funkce je možné otevřít soubor s číslem souboru, které ještě není použito.

Příkaz Open [Runtime]

Otevře datový kanál.

Příkaz Reset [Runtime]

Zavře všechny otevřené soubory a zapíše obsah všech souborových bufferů na disk.

Příkaz Open [Runtime]

Otevře datový kanál.

Syntaxe:

```
Open FileName As String [For Mode] [Access IOMode] [Protected] As [#]FileNumber As
Integer [Len = DatasetLength]
```

Parametry:

FileName: Název a cesta souboru, který chcete otevřít. Pokud se pokusíte otevřít neexistující soubor (Access = Read), zobrazí se chybová zpráva. Pokud se pokusíte zapsat do neexistujícího souboru (Access = Write), vytvoří se nový soubor.

Mode: Klíčové slovo určující režim souboru. Platné hodnoty: Append (připojení na konec sekvenčního souboru), binary (k datům lze přistupovat po bajtech pomocí Get a Put), Input (otevře datový kanál pro čtení), Output (otevře datový kanál pro zápis) a Random (úprava relativních souborů).

IOMode: Klíčové slovo definující typ přístupu. Platné hodnoty: Read (jen pro čtení), Write (jen pro zápis), Read Write (obojí).

Protected: Klíčové slovo definující bezpečnostní stav souboru po otevření. Platné hodnoty: Shared (soubor mohou otevřít i další aplikace), Lock Read (soubor je chráněn proti čtení), Lock Write (soubor je chráněn proti zápisu), Lock Read Write (zamezí přístup k souboru).

FileNumber: Celočíselný výraz od 0 do 511 určující číslo volného datového kanálu. Skrze datový kanál je možné předávat příkazy k přístupu k souboru. Číslo souboru musí být těsně před příkazem Open určeno funkcí FreeFile.

DatasetLength: Pro relativní soubory vrací délku určitého záznamu.

Upravovat je možné jen obsah souboru otevřeného příkazem Open. Pokud se pokusíte otevřít již otevřený soubor, zobrazí se chybová zpráva.

Příklad:

```
Sub ExampleWorkWithAFile
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
Dim sMsg as String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "Toto je řádek textu"
Print #iNumber, "Toto je další řádek textu"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
While not eof(iNumber)
Line Input #iNumber, sLine
If sLine <>"" then
sMsg = sMsg & sLine & chr(13)
end if
wend
Close #iNumber
Msgbox sMsg
End Sub
```

Funkce GetGuiType [Runtime]

Vrátí číselnou hodnotu určující grafické uživatelské rozhraní.

Tato funkce je poskytnuta jen pro zpětnou kompatibilitu s předchozími verzemi. V prostředí klient-server není návratová hodnota definována.

Syntaxe:

GetGUIType()

Návratová hodnota:

Celé číslo

Návratové hodnoty:

1: Windows

4: UNIX

Příklad:

```
Sub ExampleEnvironment
MsgBox GetGUIType
End Sub
```

Funkce GetGuiType [Runtime]

Vrátí číselnou hodnotu určující grafické uživatelské rozhraní.

Tato funkce je poskytnuta jen pro zpětnou kompatibilitu s předchozími verzemi. V prostředí klient-server není návratová hodnota definována.

Syntaxe:

GetGUIType()

Návratová hodnota:

Celé číslo

Návratové hodnoty:

1: Windows

4: UNIX

Příklad:

```
Sub ExampleEnvironment
MsgBox GetGUIType
End Sub
```

Funkce FreeFile [Runtime]

Vrátí další dostupné číslo souboru pro otevření souboru. Pomocí této funkce je možné otevřít soubor s číslem souboru, které ještě není použito.

Syntaxe:

```
FreeFile
```

Návratová hodnota:

Celé číslo

Parametry:

Tuto funkci lze použít pouze okamžitě před příkazem Open. FreeFile vrátí další dostupné číslo souboru, ale nerezervuje ho.

Chybové kódy

5 Neplatné volání procedury

67 Příliš mnoho otevřených souborů

Příklad:

```
Sub ExampleWorkWithAFile
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
Dim sMsg as String
aFile = "c:\data.txt"
sMsg = ""
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "First line of text"
Print #iNumber, "Another line of text"
Close #iNumber
iNumber = Freefile
Open aFile For Input As #iNumber
While not eof(#iNumber)
Line Input #iNumber, sLine
If sLine <>"" then
sMsg = sMsg & sLine & chr(13)
end if
wend
Close #iNumber
Msgbox sMsg
End Sub
```

Příkaz Close [Runtime]

Zavře určený soubor, který byl otevřen příkazem Open.

Syntaxe:

```
Close Číslo_souboru As Integer[, Číslo_souboru_2 As Integer[,...]]
```

Parametry:

Číslo_souboru: Celočíselný výraz určující číslo datového kanálu, který byl otevřen příkazem **Open**.

Příklad:

```
Sub ExampleWorkWithAFile
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
Dim sMsg as String
aFile = "c:\data.txt"
sMsg = ""
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "First line of text"
Print #iNumber, "Another line of text"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
While not eof(iNumber)
Line Input #iNumber, sLine
If sLine <>"" then
sMsg = sMsg & sLine & chr(13)
end if
wend
Close #iNumber
Msgbox sMsg
End Sub
```

Funkce GetSolarVersion [Runtime]

Vrací interní číslo verze OpenOffice.org.

Syntaxe:

s = GetSolarVersion

Návratová hodnota:

Řetězec

Příklad:

```
Sub ExampleGetSolarVersion
Dim sSep As String
sSep = GetSolarVersion
MsgBox sSep,64,"Číslo verze"
End Sub
```

Funkce Environ [Runtime]

Vrátí hodnotu proměnné prostředí. Proměnné prostředí závisí na operačním systému.

Syntaxe:

```
Environ (Environment As String)
```

Návratová hodnota:

Řetězec

Parametry:

Environment: Proměnná prostředí, jejíž hodnotu chcete vrátit.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleEnviron
Dim sTemp As String
sTemp=Environ ("TEMP")
If sTemp = "" Then sTemp=Environ("TMP")
MsgBox "'" & sTemp & "'" ,64,"Adresář dočasných souborů:"
End Sub
```

Příkaz Reset [Runtime]

Zavře všechny otevřené soubory a zapíše obsah všech souborových bufferů na disk.

Syntaxe:

```
Reset
```

Příklad:

```
Sub ExampleReset
On Error Goto ErrorHandler
Dim iNumber As Integer
Dim iCount As Integer
Dim sLine As String
Dim aFile As String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "Toto je nový řádek textu"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
For iCount = 1 to 5
Line Input #iNumber, sLine
If sLine <>"" then
rem
end if
Next iCount
Close #iNumber
Exit Sub
ErrorHandler:
Reset
MsgBox "Všechny soubory budou zavřeny",0,"Error"
End Sub
```

Funkce pro vstup/výstup do souborů

Příkaz Get [Runtime]

Načte do proměnné záznam z relativního souboru nebo sekvenci bajtů z binárního souboru.

Příkaz Input# [Runtime]

Načte data z otevřeného sekvenčního souboru.

Příkaz Line Input # [Runtime]

Načítá řetězce ze sekvenčního souboru do proměnné.

Příkaz Put [Runtime]

Zapíše záznam do relativního souboru nebo sekvenci bajtů do binárního souboru.

Příkaz Write [Runtime]

Zapíše data do sekvenčního souboru.

Funkce Loc [Runtime]

Vrátí aktuální umístění v otevřeném souboru.

Funkce Seek [Runtime]

Vrátí pozici pro příští zápis nebo čtení v souboru, který byl otevřen příkazem Open.

Funkce Eof [Runtime]

Zjišťuje, zda ukazatel souboru dosáhl konce souboru.

Funkce Lof [Runtime]

Vrátí velikost otevřeného souboru v bajtech.

Funkce GetSystemTicks [Runtime]

Vrátí počet systémových tiků operačního systému. Tuto funkci je možné použít k optimalizaci některých procesů.

Syntaxe:

GetSystemTicks()

Návratová hodnota:

Typu Long

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleWait
Dim lTick As Long
lTick = GetSystemTicks()
wait 2000
lTick = (GetSystemTicks() - lTick)
MsgBox "" & lTick & " Ticks" ,0,"Konec přestávky"
End Sub
```

Příkaz Wait [Runtime]

Přeruší spouštění programu na dobu, kterou určíte v milisekundách.

Syntaxe:

Wait millisec

Parametry:

millisec: Číselný výraz určující dobu (v milisekundách), kterou bude program čekat.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleWait
Dim lTick As Long
lTick = GetSystemTicks()
wait 2000
lTick = (GetSystemTicks() - lTick)
MsgBox "" & lTick & " Ticks" ,0,"Konec přestávky"
End Sub
```

Příkaz Get [Runtime]

Načte do proměnné záznam z relativního souboru nebo sekvenci bajtů z binárního souboru.
Viz též: Příkaz PUT

Syntaxe:

```
Get [#] FileNumber As Integer, [Position], Variable
```

Parametry:

FileNumber: Celočíselný výraz určující číslo souboru.
Position: Pro soubory otevřené v režimu Random určuje číslo záznamu, který chcete načíst.
Pro soubory otevřené v režimu Binary určuje bajt v souboru, na kterém začne čtení.
Je-li **Position** vynecháno, použije se současná pozice nebo současný záznam v souboru.
Variable: Název proměnné, která bude přečtena. Kromě objektů je možné použít jakékoliv proměnné.

Příklad:

```
Sub ExampleRandomAccess
Dim iNumber As Integer
Dim sText As Variant REM Musí být variant
Dim aFile As String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Random As #iNumber Len=32
Seek #iNumber,1 REM Přesun na začátek
Put #iNumber,, "Toto je první řádka textu" REM Zaplnit řádku textem
Put #iNumber,, "Toto je druhá řádka textu"
Put #iNumber,, "Toto je třetí řádka textu"
Seek #iNumber,2
Get #iNumber,,sText
Print sText
Close #iNumber
iNumber = Freefile
Open aFile For Random As #iNumber Len=32
Get #iNumber,2,sText
Put #iNumber,,"Toto je nový text"
Get #iNumber,1,sText
Get #iNumber,2,sText
Put #iNumber,20,"Toto je text v záznamu 20"
Print Lof(#iNumber)
Close #iNumber
end sub
```

Příkaz Put [Runtime]

Zapíše záznam do relativního souboru nebo sekvenci bajtů do binárního souboru.
Viz též: Příkaz Get

Syntaxe:

```
Put [#] FileNumber As Integer, [position], Variable
```

Parametry:

FileNumber: Celočíselný výraz určující soubor, do kterého chcete zapisovat.

Position: Pro relativní soubory (přístup Random) určuje číslo záznamu, který chcete zapsat.

Pro binární soubory (přístup Binary) určuje pozici bajtu, kde má zápis začít.

Variable: Název proměnné, kterou chcete zapsat do souboru.

Poznámka pro relativní soubory: Pokud obsah této proměnné neodpovídá délce záznamu, která je určena v klauzuli **Len** příkazu **Open**, zaplní se mezera mezi koncem nově zapsaného záznamu a dalším záznamem existujícími daty ze souboru, do kterého zapisujete.

Poznámka pro binární soubory: Obsah proměnné se zapíše na určenou pozici a ukazatel souboru se přesune přímo za poslední bajt. Mezi záznamy nezůstává žádné místo.

Příklad:

```
Sub ExampleRandomAccess
Dim iNumber As Integer
Dim sText As Variant REM Musí být typu variant
Dim aFile As String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Random As #iNumber Len=32
Seek #iNumber,1 REM Pozice pro začátek zápisu
Put #iNumber,, "Toto je první řádka textu" REM Zaplnit řádku textem
Put #iNumber,, "Toto je druhá řádka textu"
Put #iNumber,, "Toto je třetí řádka textu"
Seek #iNumber,2
Get #iNumber,,sText
Print sText
Close #iNumber
iNumber = Freefile
Open aFile For Random As #iNumber Len=32
Get #iNumber,2,sText
Put #iNumber,,"Toto je nový text"
Get #iNumber,1,sText
Get #iNumber,2,sText
Put #iNumber,20,"Toto je text v záznamu 20"
Print Lof(#iNumber)
Close #iNumber
end sub
```

Funkce Shell [Runtime]

Spustí další aplikaci a v případě potřeby definuje příslušný styl okna.

Syntaxe

Shell (NázevCesty As String[, StylOkna As Integer][, Param As String][, bSync])

Parametr

NázevCesty

Kompletní cesta a název programu, který chcete spustit.

StylOkna

Volitelný číselný výraz určující styl okna, ve kterém se program spustí. Jsou možné následující hodnoty:

0	Je aktivní skryté okno programu.
1	Je aktivní okno programu ve standardní velikost.
2	Je aktivní minimalizované okno programu.
3	Je aktivní maximalizované okno programu.
4	Standardní velikost okna, není aktivní.
6	Minimalizované okno programu, není aktivní.
10	Zobrazení na celou obrazovku.

Param

Libovolný řetězec určující parametry předávané na příkazové řádce.

bSync

Je-li tato hodnota nastavena na **True**, příkaz **Shell** a všechny procesy OpenOffice.org čekají, dokud neskončí spuštěný program. Je-li tato hodnota nastavena na **False**, řízení se okamžitě předá zpět. Výchozí hodnota je **False**.

Chybové kódy

5 Neplatné volání procedury

53 Soubor nenalezen

73 Vlastnost není implementována

Příklad

Sub ExampleShellForWin

Shell("c:\windows\calc.exe",2)

end sub

Příkaz Beep [Runtime]

Zahraje tón na reproduktoru počítače. Tón závisí na systému. Není možné upravit jeho hlasitost nebo výšku.

Syntaxe:

Beep

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleBeep
beep
beep
beep
end sub
```

Příkaz Input# [Runtime]

Načte data z otevřeného sekvenčního souboru.

Syntaxe:

```
Input #FileNumber As Integer; var1[, var2[, var3[,...]]]
```

Parametry:

FileNumber: Číslo souboru, který obsahuje data, jež chcete načíst. Před použitím INPUT musí být soubor otevřen příkazem Open.

var: Číselná nebo řetězcová proměnná, které přiřadíte hodnoty načtené z otevřeného souboru.

Příkaz **Input#** přečte z otevřeného souboru číselné nebo řetězcové hodnoty a přiřadí data jedné či více proměnným. Číselné proměnné se načítají až do prvního znaku návrat vozíku (Asc=13), nový řádek (Asc=10), mezery nebo čárky. Řetězcové proměnné se načítají až do prvního znaku návrat vozíku (Asc=13), nový řádek (Asc=10) nebo čárky.

Data a datové typy v otevřeném souboru se musí objevit ve stejném pořadí jako proměnné, které jsou předány parametrem "var". Pokud přiřadíte nečíselnou hodnotu číselné proměnné, "var" se přiřadí hodnota "0".

U záznamů, které jsou odděleny čárkami, nelze čárku přiřadit k řetězcové proměnné. Uvozovky (") v souboru jsou také zahozeny. Pokud chcete ze souboru načíst tyto znaky, použijte příkaz **Line Input#** k načtení čistě textových souborů (souborů obsahujících jen tisknutelné znaky) řádek po řádku.

Pokud je při čtení datového prvku dosažen konec souboru, dojde k chybě a proces bude ukončen.

Příklad:

```
Sub ExampleWorkWithAFile
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
Dim sMsg as String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "Toto je řádek textu"
Print #iNumber, "Toto je další řádek textu"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
While not eof(iNumber)
Line Input #iNumber, sLine
If sLine <>"" then
sMsg = sMsg & sLine & chr(13)
end if
wend
Close #iNumber
Msgbox sMsg
End Sub
```

Příkaz Line Input # [Runtime]

Načítá řetězce ze sekvenčního souboru do proměnné.

Syntaxe:

```
Line Input #FileNumber As Integer, Var As String
```

Parametry:

ČísloSouboru: Číslo souboru, který obsahuje data, která chcete načíst. Soubor musí být předem otevřen příkazem Open s klíčovým slovem INPUT.

var: Název proměnné, která bude obsahovat výsledek.

Příkazem **Line Input#** je možné načítat řetězce z otevřeného souboru do proměnné. Řetězcové proměnné se načítají řádek po řádku, až do prvního znaku návrat vozíku (Asc=13) nebo nový řádek (Asc=10). Znaky konce řádku nejsou zahrnuty ve výsledném řetězci.

Příklad:

```
Sub ExampleWorkWithAFile
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
Dim sMsg as String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "Toto je řádek textu"
Print #iNumber, "Toto je další řádek textu"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
While not eof(iNumber)
Line Input #iNumber, sLine
If sLine <>"" then
sMsg = sMsg & sLine & chr(13)
end if
wend
Close #iNumber
Msgbox sMsg
End Sub
```

Funkce CreateObject [Runtime]

Vytvoří UNO objekt. Na Windows může vytvořit také OLE objekty.

Tato metoda vytvoří instanci typu, který je předán jako parametr.

Funkce GetDefaultContext [Runtime]

Vrátí výchozí kontext process service factory, pokud existuje, jinak vrátí null.

ThisComponent [Runtime]

Adresuje aktivní komponentu, aby bylo možné číst a nastavovat její vlastnosti. ThisComponent se používá v dokumentu Basic, kde představuej dokument, do kterého Basic patří. Typ objektu zpřístupněného přes ThisComponent závisí na typu dokumentu.

GlobalScope [Runtime]

Zdrojový kód a dialogy jazyka Basic jsou uspořádány v systému knihoven.

Další příkazy

Toto je seznam funkcí a příkazů, které nespadají do jiné kategorie.

Příkaz Beep [Runtime]

Zahraje tón na reproduktoru počítače. Tón závisí na systému. Není možné upravit jeho hlasitost nebo výšku.

Funkce Shell [Runtime]

Spustí další aplikaci a v případě potřeby definuje příslušný styl okna.

Příkaz Wait [Runtime]

Přeruší spouštění programu na dobu, kterou určíte v milisekundách.

Funkce GetSystemTicks [Runtime]

Vrátí počet systémových tiků operačního systému. Tuto funkci je možné použít k optimalizaci některých procesů.

Funkce Environ [Runtime]

Vrátí hodnotu proměnné prostředí. Proměnné prostředí závisí na operačním systému.

Funkce GetSolarVersion [Runtime]

Vrací interní číslo verze OpenOffice.org.

Funkce GetGuiType [Runtime]

Vrátí číselnou hodnotu určující grafické uživatelské rozhraní.

Funkce TwipsPerPixelX [Runtime]

Vrátí počet twips, které představují šířku pixelu.

Funkce TwipsPerPixelY [Runtime]

Vrátí počet twips, které představují výšku pixelu.

Funkce CreateUnoStruct [Runtime]

Vytvoří instanci typu Uno structure.

Funkce CreateUnoService [Runtime]

Inicializuje Uno službu s ProcessServiceManager.

Funkce GetProcessServiceManager [Runtime]

Vrátí službu typu ProcessServiceManager (centrální ServiceManager pro Uno).

Funkce CreateUnoDialog [Runtime]

Vytvoří objekt typu Uno jazyka Basic, který představuje ovládací prvek dialogu typu Uno v průběhu programu jazyka Basic.

Funkce CreateUnoListener [Runtime]

Vytvoří instanci Listener.

Funkce CreateUnoValue [Runtime]

Vrátí objekt, který představuje striktně typovou hodnotu odkazující na typ Uno.

Příkaz Write [Runtime]

Zapíše data do sekvenčního souboru.

Syntaxe:

```
Write [#FileName], [Expressionlist]
```

Parametry:

FileName: Jakýkoliv číselný výraz, který obsahuje číslo souboru nastavené příkazem Open pro daný soubor.

Expressionlist: Proměnné nebo výrazy, které chcete vložit do souboru, oddělené čárkami.

Je-li vynechán seznam výrazů, příkaz **Write** připojí do souboru prázdný řádek.

Chcete-li do nového či existujícího souboru přidat výraz, soubor musí být otevřen v režimu **Output** nebo **Append**.

Zapisované řetězce se uzavřou uvozovkami a oddělí čárkami. Tyto oddělovače nemusíte do seznamu výrazů uvádět.

Každý příkaz **Write** vypíše jako poslední záznam symbol konce řádku.

Čísla s desetinným oddělovačem se převedou podle jazykového nastavení.

Příklad:

```
Sub ExampleWrite
Dim iCount As Integer
Dim sValue As String
iCount = Freefile
open "C:\data.txt" for OutPut as iCount
sValue = "Hamburg"
Write #iCount,sValue,200
sValue = "New York"
Write #iCount,sValue,300
sValue = "Miami"
Write #iCount,sValue,450
close #iCount
end sub
```

Funkce Loc [Runtime]

Vrátí aktuální umístění v otevřeném souboru.

Syntaxe:

```
Loc(FileNumber)
```

Návratová hodnota:

Typu Long

Parametry:

FileNumber: Celočíselný výraz určující číslo otevřeného souboru.

Pokud se funkce Loc použije na soubor s přístupem Random, vrátí číslo posledního záznamu, který byl naposledy přečten nebo zapsán.

U sekvenčních souborů vrátí funkce Loc umístění v souboru děleno 128. U binárních souborů bude vráceno umístění posledního načteného nebo zapsaného bajtu.

Chybové kódy

5 Neplatné volání procedury

52 Špatný název nebo číslo souboru

Funkce ConvertFromURL [Runtime]

Převede adresu URL souboru na název systémového souboru.

Syntaxe:

ConvertFromURL(NázevSouboru)

Návratová hodnota:

Řetězec

Parametry:

NázevSouboru: Název souboru jako řetězec.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
systemFile$ = "c:\folder\mytext.txt"
url$ = ConvertToURL( systemFile$ )
print url$
systemFileAgain$ = ConvertFromURL( url$ )
print systemFileAgain$
```

Funkce ConvertToURL [Runtime]

Převede název systémového souboru na adresu URL souboru.

Syntaxe:

ConvertToURL(NázevSouboru)

Návratová hodnota:

Řetězec

Parametry:

NázevSouboru: Název souboru jako řetězec.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
systemFile$ = "c:\folder\mytext.txt"
url$ = ConvertToURL( systemFile$ )
print url$
systemFileAgain$ = ConvertFromURL( url$ )
print systemFileAgain$
```

Funkce Seek [Runtime]

Vrátí pozici pro příští zápis nebo čtení v souboru, který byl otevřen příkazem Open.
U souborů s náhodným přístupem vrátí funkce Seek číslo dalšího záznamu, který se má načíst.
U všech ostatních souborů tato funkce vrátí umístění bajtu, ve kterém má dojít k další operaci.
Viz též: Open, Seek.

Syntaxe:

```
Seek (FileNumber)
```

Návratová hodnota:

Typu Long

Parametry:

FileNumber: Celočíselný výraz určující číslo otevřeného souboru.

Funkce Eof [Runtime]

Zjišťuje, zda ukazatel souboru dosáhl konce souboru.

Syntaxe:

```
Eof (intexpression As Integer)
```

Návratová hodnota:

Bool

Parametry:

Intexpression: Celočíselný výraz určující číslo otevřeného souboru.

Pomocí EOF zabráníte chybám, při kterých se pokusíte získat data za koncem souboru. Když použijete příkaz Input nebo Get k načtení ze souboru, ukazatel se posune o počet načtených bajtů. Když dosáhne konce souboru, EOF vrátí hodnotu "True" (-1).

Chybové kódy

5 Neplatné volání procedury

52 Špatný název nebo číslo souboru

Příklad:

```
Sub ExampleWorkWithAFile
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
Dim sMsg as String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "First line of text"
Print #iNumber, "Another line of text"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
While not eof(iNumber)
Line Input #iNumber, sLine
If sLine <>"" then
sMsg = sMsg & sLine & chr(13)
end if
wend
Close #iNumber
Msgbox sMsg
End Sub
```

Funkce Join [Runtime]

Vrátí řetězec složený z několika podřetězců umístěných v poli.

Syntaxe:

Join (Text As String Array, Oddělovač)

Návratová hodnota:

Řetězec

Parametry:

Text: Pole řetězců.

Oddělovač (nepovinné): Řetězec, který je ve výsledném řetězci používán k oddělení podřetězců. Výchozí oddělovač je mezera. Je-li oddělovač prázdný řetězec "", podřetězce budou spojeny bez oddělovače.

Příklad:

```
Dim a(3)
Sub main()
a(0) = "ABCDE"
a(1) = 42
a(2) = "MN"
a(3) = "X Y Z"
JStr = Join1()
Call Show(JStr, Split1(JStr))
JStr = Join2()
Call Show(JStr, Split1(JStr))
JStr = Join3()
Call Show(JStr, Split1(JStr))
End Sub
Function Join1()
Join1 = Join(a(), "abc")
End Function
Function Join2()
Join2 = Join(a(), ",")
End Function
Function Join3()
Join3 = Join(a())
End Function
Function Split1(aStr)
Split1 = Split(aStr, "D")
End Function
Sub Show(JoinStr, TheArray)
l = LBound(TheArray)
u = UBound(TheArray)
total$ = "============================" + Chr$(13) + JoinStr + Chr$(13) + Chr$(13)
For i = l To u
total$ = total$ + TheArray(i) + Str(Len(TheArray(i))) + Chr$(13)
Next i
MsgBox total$
End Sub
```

Funkce Split [Runtime]

Rozdělí řetězec na pole podřetězců.

Syntaxe:

Split (Text As String, Oddělovač, Číslo)

Návratová hodnota:

Řetězec

Parametry:

Text: Jakýkoliv řetězcový výraz.
Oddělovač (nepovinné): Řetězec (jeden či více znaků), který se použije k rozdělení Textu. Výchozí znak je mezera.
Číslo (nepovinné): Počet podřetězců, které chcete vrátit.

Příklad:

```
Dim a(3)
Sub main()
a(0) = "ABCDE"
a(1) = 42
a(2) = "MN"
a(3) = "X Y Z"
JStr = Join1()
Call Show(JStr, Split1(JStr))
JStr = Join2()
Call Show(JStr, Split1(JStr))
JStr = Join3()
Call Show(JStr, Split1(JStr))
End Sub
Function Join1()
Join1 = Join(a(), "abc")
End Function
Function Join2()
Join2 = Join(a(), ",")
End Function
Function Join3()
Join3 = Join(a())
End Function
Function Split1(aStr)
Split1 = Split(aStr, "D")
End Function
Sub Show(JoinStr, TheArray)
l = LBound(TheArray)
u = UBound(TheArray)
total$ = "============================" + Chr$(13) + JoinStr + Chr$(13) + Chr$(13)
For i = l To u
total$ = total$ + TheArray(i) + Str(Len(TheArray(i))) + Chr$(13)
Next i
MsgBox total$
End Sub
```

Funkce Lof [Runtime]

Vrátí velikost otevřeného souboru v bajtech.

Syntaxe:

```
Lof (FileNumber)
```

Návratová hodnota:

Typu Long

Parametry:

FileNumber: Celočíselný výraz určující číslo otevřeného souboru.

Chcete-li zjistit velikost souboru, který není otevřen, použijte funkci **FileLen**.

Chybové kódy

5 Neplatné volání procedury
52 Špatný název nebo číslo souboru

Příklad:

```
Sub ExampleRandomAccess
Dim iNumber As Integer
Dim sText As Variant REM Musí být variant
Dim aFile As String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Random As #iNumber Len=32
Seek #iNumber,1 REM Pozice na začátku
Put #iNumber,, "Toto je první řádka textu" REM Zaplnit textem
Put #iNumber,, "Toto je druhá řádka textu"
Put #iNumber,, "Toto je třetí řádka textu"
Seek #iNumber,2
Get #iNumber,,sText
Print sText
Close #iNumber
iNumber = Freefile
Open aFile For Random As #iNumber Len=32
Get #iNumber,2,sText
Put #iNumber,,"Toto je nová řádka textu"
Get #iNumber,1,sText
Get #iNumber,2,sText
Put #iNumber,20,"Toto je text v záznamu 20"
Print Lof(#iNumber)
Close #iNumber
end sub
```

Správa souborů

Zde jsou popsány funkce a příkazy pro správu souborů.

Příkaz ChDir [Runtime]

Změní aktuální adresář nebo jednotku.

Příkaz ChDrive [Runtime]

Změní aktuální jednotku.

Funkce CurDir [Runtime]

Vrátí řetězec, který představuje aktuální cestu na určené jednotce.

Funkce Dir [Runtime]

Vrátí jméno souboru, adresáře nebo všech souborů a adresářů jednotky nebo adresáře, který splňuje zadanou vyhledávací podmínku.

Funkce FileAttr [Runtime]

Vrátí přístupový režim souboru nebo přístupové číslo k souboru, který byl otevřen příkazem Open. Přístupové číslo závisí na operačním systému (OSH = ukazatel operačního systému).

Příkaz FileCopy [Runtime]

Zkopíruje soubor.

Funkce FileDateTime [Runtime]

Vrací řetězec, který obsahuje datum a čas, kdy byl soubor vytvořen nebo naposledy upraven.

Funkce FileExists [Runtime]

Určuje, zda soubor nebo adresář existuje na datovém médiu.

Funkce FileLen [Runtime]

Vrátí délku souboru v bajtech.

Funkce GetAttr [Runtime]

Vrátí bitový vzorek, který určuje typ souboru nebo název jednotky či adresáře.

Příkaz Kill [Runtime]

Smaže soubor z disku.

Příkaz MkDir [Runtime]

Vytvoří na datovém médiu nový adresář.

Příkaz Name [Runtime]

Přejmenuje stávající soubor nebo adresář.

Příkaz RmDir [Runtime]

Odstraní stávající adresář z datového média.

Funkce UCase [Runtime]

Převede malá písmena na velká.
Viz též: Funkce LCase
Syntaxe:
UCase (Text As String)
Návratová hodnota:
Řetězec

Parametry:

Text: Řetězec, který chcete převést.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleLUCase
Dim sVar As String
sVar = "Las Vegas"
Print LCase(sVar) REM vrací "las vegas"
Print UCase(sVar) REM vrací "LAS VEGAS"
end Sub
```

Funkce Trim [Runtime]

Odstraní všechny mezery ze začátku a konce řetězce.

Syntaxe:

Trim(Text As String)

Návratová hodnota:

Řetězec

Parametry:

Text: Jakýkoliv řetězcový výraz.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSpaces
Dim sText2 as String,sText as String,sOut as String
sText2 = " <*Las Vegas*> "
sOut = """+sText2 +"""+ Chr(13)
sText = Ltrim(sText2) REM sText = "<*Las Vegas*> "
sOut = sOut + """+sText +""" + Chr(13)
sText = Rtrim(sText2) REM sText = " <*Las Vegas*>"
sOut = sOut +""+ sText +""" + Chr(13)
sText = Trim(sText2) REM sText = "<*Las Vegas*>"
sOut = sOut +""+ sText +"""
MsgBox sOut
end sub
```

Příkaz SetAttr [Runtime]

Nastaví atribut pro zadaný soubor.

Příkaz ChDir [Runtime]

Změní aktuální adresář nebo jednotku.

Tento příkaz momentálně nefunguje podle dokumentace. Více informací najdete v popisu této chyby.

Syntaxe:

```
ChDir Text As String
```

Parametry:

Text: Řetězcový výraz, který určuje cestu nebo jednotku.

Chcete-li změnit pouze aktuální jednotku, zadejte písmeno jednotky a dvojtečku.

Chybové kódy

5 Neplatné volání procedury
76 Adresář nenalezen

Příklad:

```
Sub ExampleChDir
Dim sDir1 as String , sDir2 as String
sDir1 = "c:\Test"
sDir2 = "d:\private"
ChDir( sDir1 )
msgbox CurDir
ChDir( sDir2 )
msgbox CurDir
end sub
```

Funkce RTrim [Runtime]

Smaže mezery na konci řetězce.
Viz též: Funkce LTrim

Syntaxe:

RTrim (Text As String)

Návratová hodnota:

Řetězec

Parametry:

Text: Libovolný řetězec.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSpaces
Dim sText2 as String,sText as String,sOut as String
sText2 = " <*Las Vegas*> "
sOut = """+sText2 +"""+ Chr(13)
sText = Ltrim(sText2) REM sText = "<*Las Vegas*> "
sOut = sOut + """+sText +"" + Chr(13)
sText = Rtrim(sText2) REM sText = " <*Las Vegas*>"
sOut = sOut +""+ sText +"" + Chr(13)
sText = Trim(sText2) REM sText = "<*Las Vegas*>"
sOut = sOut +""+ sText +""
MsgBox sOut
end sub
```

Příkaz RSet [Runtime]

Zarovná text typu String vpravo v rámci proměnné typu String nebo zkopíruje uživatelem definovaný typ proměnné do jiné.

Syntaxe:

RSet Text As String = Text nebo RSet Proměnná1 = Proměnná2

Parametry:

Text: Libovolný řetězec.

Text: Řetězec, který chcete zarovnat vpravo v řetězcové proměnné.

Proměnná1: Název proměnné uživatelem definovaného typu, do které chcete kopírovat.

Proměnná2: Název proměnné uživatelem definovaného typu, ze které chcete kopírovat.

Je-li řetězec kratší než řetězcová proměnná, **RSet** zarovná řetězec vpravo v řetězcové proměnné. Zbývající místa v řetězci budou nahrazena mezerami. Je-li řetězec delší než řetězcová proměnná, budou oříznuty znaky překračující délku proměnné a zbývající znaky budou zarovnány vpravo.

Příkaz RSet je také je možné použít k přiřazení proměnné uživatelem definovaného typu do jiné proměnné stejného typu.

Následující příklad používá příkazy **RSet** a **LSet** k úpravě levého a pravého zarovnání řetězce.

Příklad:

```
Sub ExampleRLSet
Dim sVar as string
Dim sExpr as string
sVar = String(40,"*")
sExpr = "SBX"
REM Zarovná "SBX" vpravo ve 40znakovém řetězci
REM nahradí hvězdičky mezerami
RSet sVar = sExpr
Print ">"; sVar; "<"
sVar = String(5,"*")
sExpr = "123457896"
RSet sVar = sExpr
Print ">"; sVar; "<"
sVar = String(40,"*")
sExpr = "SBX"
REM Zarovná "SBX" vlevo ve 40znakovém řetězci
LSet sVar = sExpr
Print ">"; sVar; "<"
sVar = String(5,"*")
sExpr = "123456789"
LSet sVar = sExpr
Print ">"; sVar; "<"
End Sub
```

Příkaz ChDrive [Runtime]

Změní aktuální jednotku.

Tento příkaz momentálně nefunguje podle dokumentace. Více informací najdete v popisu této chyby.

Syntaxe:

```
ChDrive Text As String
```

Parametry:

Text: Řetězec obsahující nové písmeno jednotky. Pokud chcete, je možné použít URL notaci.

Jednotka musí mít přiřazeno velké písmeno. V systému Windows je přiřazené písmeno omezeno nastavením LASTDRV. Je-li argumentem víceznakový řetězec, použije se pouze první znak. Pokud se pokusíte přistoupit na neexistující jednotku, dojde k chybě, na kterou je možné reagovat příkazem OnError.

Chybové kódy

5 Neplatné volání procedury

68 Zařízení není dostupné

76 Adresář nenalezen

Příklad:

```
Sub ExampleCHDrive
ChDrive "D" REM Je možné, jen pokud existuje jednotka 'D'.
End Sub
```

Funkce CurDir [Runtime]

Vrátí řetězec, který představuje aktuální cestu na určené jednotce.

Tento příkaz momentálně nefunguje podle dokumentace. Více informací najdete v popisu této chyby.

Syntaxe:

```
CurDir [(Text As String)]
```

Návratová hodnota:

Řetězec

Parametry:

Text: Řetězec určující existující jednotku (např. "C" pro první diskovou oblast na prvním pevném disku).

Pokud není určena jednotka nebo je určena prázdným řetězcem (""), vrátí CurDir cestu pro současnou jednotku. OpenOffice.org Basic ohlásí chybu, je-li syntaxe určení jednotky nesprávná, jednotka neexistuje nebo je zadaný znak vyšší než znak určený v CONFIG.SYS příkazem Lastdrive.

Tato funkce nerozlišuje malá a velká písmena.

Chybové kódy

5 Neplatné volání procedury

68 Zařízení není dostupné

7 Nedostatek paměti

51 Interní chyba

Příklad:

```
Sub ExampleCurDir
Dim sDir1 as String , sDir2 as String
sDir1 = "c:\Test"
sDir2 = "d:\private"
ChDir( sDir1 )
msgbox CurDir
ChDir( sDir2 )
msgbox CurDir
end sub
```

Funkce Right [Runtime]

Vrátí počet n znaků umístěných ve výrazu typu String nejvíce vpravo.

Viz též: Funkce Left.

Syntaxe:

Right (Text As String, n As Long)

Návratová hodnota:

Řetězec

Parametry:

Text: Řetězec, jehož část chcete získat.

n: Číselný výraz určující počet znaků, které chcete vrátit. Pokud **n** = 0, vrátí se prázdný řetězec. Maximální povolená hodnota je 65535.

Následující příklad převádí datum z formátu YYYY-MM-DD do formátu MM/DD/YYYY.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleUSDate
Dim sInput As String
Dim sUS_date As String
sInput = InputBox("Zadejte, prosím, datum v mezinárodním formátu 'YYYY-MM-DD'")
sUS_date = Mid(sInput, 6, 2)
sUS_date = sUS_date & "/"
sUS_date = sUS_date & Right(sInput, 2)
sUS_date = sUS_date & "/"
sUS_date = sUS_date & Left(sInput, 4)
MsgBox sUS_date
End Sub
```

Funkce Mid, příkaz Mid [Runtime]

Vrátí určenou část řetězce (**funkce Mid**) nebo nahradí část řetězce jiným řetězcem (**příkaz Mid**).

Syntaxe:

Mid (Text As String, Začátek As Long [, Délka As Long]) nebo Mid (Text As String, Začátek As Long , Délka As Long, Text As String)

Návratová hodnota:

Typu String (jen v případě funkce)

Parametry:

Text: Řetězec, který chcete upravit.

Začátek: Číselný výraz určující pozici znaku, kde začíná část řetězce, kterou chcete nahradit nebo vrátit. Maximální povolená hodnota je 65535.

Délka: Číselný výraz určující počet znaků, které chcete nahradit nebo vrátit. Maximální povolená hodnota je 65535.

Je-li ve **funkci Mid** vynechán parametr Délka, vrátí se všechny znaky od počáteční pozice do konce řetězce.

Je-li v **příkazu Mid** parametr Délka menší než délka textu, kterým chcete nahradit, text bude zkrácen na určenou délku.

Text: Řetězec, kterým chcete nahradit určenou část řetězce (**příkaz Mid**).

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleUSDate
Dim sInput As String
Dim sUS_date As String
sInput = InputBox("Zadejte, prosím, datum v mezinárodním formátu 'YYYY-MM-DD'")
sUS_date = Mid(sInput, 6, 2)
sUS_date = sUS_date & "/"
sUS_date = sUS_date & Right(sInput, 2)
sUS_date = sUS_date & "/"
sUS_date = sUS_date & Left(sInput, 4)
MsgBox sUS_date
End Sub
```

Funkce Dir [Runtime]

Vrátí jméno souboru, adresáře nebo všech souborů a adresářů jednotky nebo adresáře, který splňuje zadanou vyhledávací podmínku.

Syntaxe:

```
Dir [(Text As String) [, Attrib As Integer]]
```

Návratová hodnota:

Řetězec

Parametry:

Text: Řetězec určující hledanou cestu, adresář nebo soubor. Tento argument je možné určit jen při prvním volání funkce Dir. Pokud chcete, je možné zadat cestu v URL notaci.

Attrib: Celočíselný výraz určující atributy souboru. Funkce Dir vrací jen soubory, které odpovídají určeným atributům. Několik atributů je možné zkombinovat, když sečtete jejich hodnoty:

0 : Normální soubory

16 : Vrátí pouze název adresáře.

Tímto atributem ověříte, zda soubor nebo adresář existuje, nebo určíte všechny soubory a složky v určitém adresáři.

Chcete-li zjistit, zda soubor existuje, zadejte kompletní cestu a název souboru. Pokud soubor nebo adresář neexistuje, funkce Dir vrátí prázdný řetězec ("").

Chcete-li vygenerovat seznam všech existující souborů v určitém adresáři, postupujte takto: poprvé zavolejte funkci Dir s kompletní vyhledávací cestou pro soubory, např. "D:\Files*.sxw". Je-li cesta správná a vyhledávání nalezne alespoň jeden soubor, funkce Dir vrátí název prvního souboru, který odpovídá cestě. Další názvy souborů získáte opětovným voláním funkce Dir, ovšem bez argumentů.

Chcete-li vrátit jen adresáře, použijte parametr atributů. To stejné platí, pokud chcete určit název jednotky (např. pevného disku).

Chybové kódy

5 Neplatné volání procedury

53 Soubor nenalezen

Příklad:

```
Sub ExampleDir
REM Zobrazí všechny soubory a adresáře
Dim sPath As String
Dim sDir as String, sValue as String
sDir="Adresáře:"
sPath = CurDir
sValue = Dir$(sPath + getPathSeparator + "*",16)
Do
If sValue <> "." and sValue <> ".." Then
if (GetAttr( sPath + getPathSeparator + sValue) AND 16) >0 then
REM načtení adresářů
sDir = sDir & chr(13) & sValue
End If
End If
sValue = Dir$
Loop Until sValue = ""
MsgBox sDir,0,sPath
End sub
```

Funkce FileAttr [Runtime]

Vrátí přístupový režim souboru nebo přístupové číslo k souboru, který byl otevřen příkazem Open. Přístupové číslo závisí na operačním systému (OSH = ukazatel operačního systému).

Pokud používáte 32bitový operační systém, není možné pomocí funkce FileAttr určit přístupové číslo souboru.

Viz též: Open

Syntaxe:

```
FileAttr (FileNumber As Integer, Attribute As Integer)
```

Návratová hodnota:

Celé číslo

Parametry:

FileNumber: Číslo souboru, který byl otevřen příkazem Open.
Attribute: Celočíselný výraz určující, jaký druh informací chcete vrátit. Máte k dispozici tyto možnosti:
1: Funkce FileAttr označuje režim přístupu k souboru.
2: Funkce FileAttr vrátí přístupové číslo operačního systému.
Pokud u druhého parametru použijete hodnotu 1, použijí se následující návratové hodnoty:
1 - INPUT (soubor otevřen pro vstup)
2 - OUTPUT (soubor otevřen pro výstup)
4 - RANDOM (soubor otevřen pro náhodný přístup)
8 - APPEND (soubor otevřen pro připojení)
32 - BINARY (soubor otevřen v binárním režimu)

Chybové kódy

5 Neplatné volání procedury
52 Špatný název nebo číslo souboru

Příklad:

```
Sub ExampleFileAttr
Dim iNumber As Integer
Dim sLine As String
Dim aFile As String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "Toto je řádek textu"
MsgBox FileAttr(#iNumber, 1 ),0,"Access mode"
MsgBox FileAttr(#iNumber, 2 ),0,"File attribute"
Close #iNumber
End Sub
```

Funkce LTrim [Runtime]

Odstraní všechny mezery na počátku řetězce.

Syntaxe:

LTrim (Text As String)

Návratová hodnota:

Řetězec

Parametry:

Text: Jakýkoliv řetězcový výraz.
Pomocí této funkce odstraníte mezery ze začátku řetězce.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSpaces
Dim sText2 As String,sText As String,sOut As String
sText2 = " <*Las Vegas*> "
sOut = """+sText2 +"""+ Chr(13)
sText = Ltrim(sText2) REM sText = "<*Las Vegas*> "
sOut = sOut + """+sText +"" + Chr(13)
sText = Rtrim(sText2) REM sText = " <*Las Vegas*>"
sOut = sOut +""+ sText +"" + Chr(13)
sText = Trim(sText2) REM sText = "<*Las Vegas*>"
sOut = sOut +""+ sText +""
MsgBox sOut
end sub
```

Příkaz LSet [Runtime]

Zarovná řetězec vlevo v řetězcové proměnné nebo zkopíruje proměnnou typu definovaného uživatelem do jiné proměnné jiného typu definovaného uživatelem.

Syntaxe:

LSet Proměnná As String = Text nebo LSet Proměnná1 = Proměnná2

Parametry:

Proměnná: Proměnná typu String obsahující řetězec, který chcete zarovnat vlevo.

Text: Řetězec, který chcete zarovnat vlevo od řetězcové proměnné.

Proměnná1: Název proměnné uživatelem definovaného typu, do které chcete kopírovat.

Proměnná2: Název proměnné uživatelem definovaného typu, ze které chcete kopírovat.

Je-li řetězec kratší než řetězcová proměnná, **LSet** zarovná řetězec vlevo v řetězcové proměnné. Zbývající místa v řetězci budou nahrazena mezerami. Je-li řetězec delší než řetězcová proměnná, budou zkopírovány jen znaky nejvíce nalevo až do délky řetězcové proměnné. Příkazem **LSet** je také možné kopírovat proměnné uživatelem definovaného typu do jiné proměnné stejného typu.

Příklad:

```
Sub ExampleRLSet
Dim sVar As String
Dim sExpr As String
sVar = String(40,"*")
sExpr = "SBX"
REM Zarovná "SBX" ve 40znakovém řetězci
REM nahradí hvězdičky mezerami
RSet sVar = sExpr
Print ">"; sVar; "<"
sVar = String(5,"*")
sExpr = "123457896"
RSet sVar = sExpr
Print ">"; sVar; "<"
sVar = String(40,"*")
sExpr = "SBX"
REM Zarovná vlevo "SBX" ve 40znakovém řetězci
LSet sVar = sExpr
Print ">"; sVar; "<"
sVar = String(5,"*")
sExpr = "123456789"
LSet sVar = sExpr
Print ">"; sVar; "<"
End Sub
```

Příkaz FileCopy [Runtime]

Zkopíruje soubor.

Syntaxe:

```
FileCopy TextFrom As String, TextTo As String
```

Parametry:

TextFrom: Řetězec určující název souboru, který chcete zkopírovat. Řetězec může volitelně obsahovat cestu a určení jednotky. Pokud chcete, je možné použít URL notaci.

TextTo: Řetězec určující, kam chcete zkopírovat zdrojový soubor. Řetězec může obsahovat určení jednotky, cestu a název souboru, nebo cestu v URL notaci.

Příkaz FileCopy lze použít ke kopírování souborů, které nejsou otevřeny.

Chybové kódy

5 Neplatné volání procedury

76 Adresář nenalezen

Příklad:

```
Sub ExampleFilecopy
Filecopy "c:\autoexec.bat", "c:\Temp\Autoexec.sav"
end sub
```

Funkce FileDateTime [Runtime]

Vrací řetězec, který obsahuje datum a čas, kdy byl soubor vytvořen nebo naposledy upraven.

Syntaxe:

```
FileDateTime (Text As String)
```

Parametry:

Text: Řetězec určující jednoznačné (žádné zástupné znaky) jméno souboru. Také je možné použít URL notaci.

Tato funkce vrátí přesný čas vytvoření nebo poslední změny souboru ve formátu "DD.MM.RRRR HH:MM:SS".

Národní nastavení použité pro formátování čísel, data a měny v jazyce OpenOffice.org Basic je možné nastavit v **Nástroje - Volby - Jazyková nastavení - Jazyky**. Ve formátovacích kódech Basic se jako desetinný oddělovač používá vždy tečka (**.**), která se při zobrazení nahradí odpovídajícím znakem podle národního nastavení.

Totéž platí pro národní nastavení formátu data, času a měny. Formátovací kódy se interpretují a zobrazí podle aktuálního národního nastavení.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleFileDateTime
msgbox FileDateTime("C:\autoexec.bat")
end sub
```

Funkce Left [Runtime]

Vrátí určený počet znaků ze začátku řetězce.

Syntaxe:

Left (Text As String, n As Long)

Návratová hodnota:

Řetězec

Parametry:

Text: Řetězec, jehož část chcete získat.

n: Číselný výraz určující počet znaků, které chcete vrátit. Pokud **n** = 0, vrátí se prázdný řetězec. Maximální povolená hodnota je 65535.

Následující příklad převede datum z formátu YYYY.MM.DD na formát MM/DD/YYYY.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleUSDate
Dim sInput As String
Dim sUS_date As String
sInput = InputBox("Zadejte, prosím, datum v mezinárodním formátu 'YYYY-MM-DD'")
sUS_date = Mid(sInput, 6, 2)
sUS_date = sUS_date & "/"
sUS_date = sUS_date & Right(sInput, 2)
sUS_date = sUS_date & "/"
sUS_date = sUS_date & Left(sInput, 4)
MsgBox sUS_date
End Sub
```

Funkce LCase [Runtime]

Převede všechna velká písmena na malá.
Viz též: Funkce UCase

Syntaxe:

LCase (Text As String)

Návratová hodnota:

Řetězec

Parametry:

Text: Řetězec, který chcete převést.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleLUCase
Dim sVar As String
sVar = "Las Vegas"
Print LCase(sVar) REM vrací "las vegas"
Print UCase(sVar) REM vrací "LAS VEGAS"
end Sub
```

Funkce FileExists [Runtime]

Určuje, zda soubor nebo adresář existuje na datovém médiu.

Syntaxe:

```
FileExists(FileName As String | DirectoryName As String)
```

Návratová hodnota:

Bool

Parametry:

FileName, DirectoryName: Řetězec jednoznačně určující soubor. Také je možné použít URL notaci.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
sub ExampleFileExists
msgbox FileExists("C:\autoexec.bat")
msgbox FileExists("file:///d|/bookmark.htm")
msgbox FileExists("file:///d|/private")
end sub
```

Funkce FileLen [Runtime]

Vrátí délku souboru v bajtech.

Syntaxe:

```
FileLen (Text As String)
```

Návratová hodnota:

Typu Long

Parametry:

Text: Řetězec určující jednoznačné (žádné zástupné znaky) jméno souboru. Také je možné použít URL notaci.

Tato funkce určuje délku souboru. Je-li funkce FileLen zavolána na otevřený soubor, vrátí délku souboru před otevřením. Chcete-li určit aktuální délku otevřeného souboru, použijte funkci Lof.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleFileLen
msgbox FileLen("C:\autoexec.bat")
end sub
```

Currency: Vloží před číslo znak dolaru a záporná čísla uzavře do závorek.

Fixed: Zobrazí před desetinným oddělovačem alespoň jednu číslici.

Standard: Zobrazí čísla s oddělovačem tisíců.

Percent: Vynásobí číslo 100 a přidá k číslu znak procent.

Scientific: Zobrazí čísla ve vědeckém formátu (např. 1.00E+03 místo 1000).

Formátovací kód lze rozdělit na tři části oddělené středníky. První část určuje formát pro kladná čísla, druhá pro záporná čísla a třetí pro nulu. Pokud určíte jen jeden formátovací kód použije se pro všechna čísla.

Národní nastavení použité pro formátování čísel, data a měny v jazyce OpenOffice.org Basic je možné nastavit v **Nástroje - Volby - Jazyková nastavení - Jazyky**. Ve formátovacích kódech Basic se jako desetinný oddělovač používá vždy tečka (.), která se při zobrazení nahradí odpovídajícím znakem podle národního nastavení.

Totéž platí pro národní nastavení formátu data, času a měny. Formátovací kódy se interpretují a zobrazí podle aktuálního národního nastavení.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleFormat
MsgBox Format(6328.2, "##,##0.00")
REM když zadáváte čísla v kódu Basic, vždy jako desetinný oddělovač používejte tečku
REM zobrazí 6,328.20 pro anglické nastavení a 6.328,20 pro německé
End Sub
```

Funkce Format [Runtime]

Převede číslo na řetězec a ten upraví podle zadaného formátu.

Syntaxe:

Format (Číslo [, Formát As String])

Návratová hodnota:

Řetězec

Parametry:

Číslo: Číselný výraz, který chcete převést na formátovaný řetězec.

Formát: Řetězec, který určuje formátování čísla. Je-li Formát vynechán, funguje funkce Format stejně jako funkce Str.

Formátovací kódy

Následující seznam popisuje kódy, které je možné použít k formátování čísla:

0: Pokud má **Číslo** na pozici, kde je ve formátovacím kódu 0, číslici, zobrazí se číslice, jinak se zobrazí nula.

Pokud má **Číslo** méně číslic, než počet nul ve formátovacím kódu (před či za desetinnou čárkou), zobrazí se nuly navíc na začátku nebo na konci. Pokud má číslo před desetinnou čárkou více číslic, než počet nul ve formátovacím kódu, číslice navíc se zobrazí bez formátování.

Desetinná část čísla se zaokrouhlí podle počtu nul, které jsou ve formátovacím kódu za desetinným oddělovačem.

#: Pokud **Číslo** obsahuje na pozici znaku # číslici, tato číslice se zobrazí, jinak se na této pozici nezobrazí nic.

Tento symbol má stejnou funkci jako 0, kromě toho, že se úvodní a koncové nuly nezobrazí, je-li ve formátovacím kódu více znaků # než číslic v čísle. Zobrazí se pouze relevantní číslice daného čísla.

.: Zástupný znak pro desetinný oddělovač.

Pokud formátovací kód obsahuje zástupné znaky # pouze nalevo od tohoto symbolu, budou čísla menší než 1 začínat desetinnou čárkou. Pokud chcete u zlomkových čísel vždy zobrazit úvodní nulu, použijte 0 jako zástupný znak pro první číslici nalevo od desetinné čárky.

%: Vynásobí číslo 100 a vloží znak procent (%).

E- E+ e- e+ : Pokud formátovací kód obsahuje alespoň jeden zástupný znak pro číslici (0 nebo #) napravo do symbolu E-, E+, e- nebo e+, číslo se naformátuje podle vědeckého (exponenciálního) formátu. Mezi číslo a exponent se vloží písmeno E nebo e. Počet zástupných znaků vpravo symbolu určuje počet číslic exponentu.

Je-li exponent záporný, zobrazí se bezprostředně před exponentem s E-, E+, e-, e+ znaménko minus. Je-li exponent kladný, zobrazí se znaménko plus pouze před exponenty s E+ nebo e+.

Oddělovač tisíců se zobrazí, pokud formátovací kód obsahuje oddělovač obklopený zástupnými znaky pro číslice (0 nebo #).

Použití tečky jako oddělovače tisíců nebo desetinného oddělovače závisí na místním nastavení. Když zadáváte číslo přímo do kódu Basic, vždy používejte tečku jako desetinný oddělovač. Podle místního nastavení vašeho systému se zobrazí skutečný desetinný oddělovač.

- + $ ( ) mezera: Plus (+), mínus (-), dolar ($), mezera nebo závorky zadané ve formátovacím kódu se zobrazí jako znaky.

Chcete-li zobrazit jiné znaky, musíte jim předřadit zpětné lomítko (\) nebo je uzavřít do uvozovek (" ").

\ : Zobrazí přímo další znak ve formátovacím kódu.

Znaky ve formátovacím kódu, které mají zvláštní význam, lze jako znaky zobrazit pouze, pokud před ně přidáte zpětné lomítko. Samotné zpětné lomítko se nezobrazí, pokud nezapíšete dvojité zpětné lomítko (\\).

Znaky, před které musíte přidat zpětné lomítko, aby se zobrazily přímo jako znaky jsou formátovací znaky pro datum a čas (a, c, d, h, m, n, p, q, s, t, w, y, /, :), formátovací znaky pro čísla (#, 0, %, E, e, čárka, tečka) a formátovací znaky pro řetězce (@, &, <, >, !).

Také je možné použít následující předdefinované formáty čísel. Kromě "General Number" se všechny předdefinované formáty zobrazují jako čísla zaokrouhlená na dvě desetinná místa.

Pokud používáte předem definované formáty, musí být název formátu uveden v uvozovkách.

Předem definovaný formát

General Number: Čísla se zobrazí, jak jsou zadána.

Funkce GetAttr [Runtime]

Vrátí bitový vzorek, který určuje typ souboru nebo název jednotky či adresáře.

Syntaxe:

```
GetAttr (Text As String)
```

Návratová hodnota:

Celé číslo

Parametry:

Text: Řetězec určující jednoznačné (žádné zástupné znaky) jméno souboru. Také je možné použít URL notaci.

Tato funkce určuje atributy zadaného souboru a vrací bitový vzorek, který vám pomůže rozeznat následující atributy souboru:

Chybové kódy

5 Neplatné volání procedury

53 Soubor nenalezen

Hodnota

0 : Normální soubory

1 : Soubory pouze pro zápis

8 : Vrátí název svazku.

16 : Vrátí pouze název adresáře.

32 : Soubor byl od posledního zálohování změněn (bit Archivovat).

Pokud chcete zjistit, zda je některý bit v bajtu atributů nastaven, použijte následující metodu:

Příklad:

```
Sub ExampleSetGetAttr
On Error Goto ErrorHandler REM Určí zpracování chyb
If Dir("C:\test",16)="" Then MkDir "C:\test"
If Dir("C:\test\autoexec.sav")="" THEN Filecopy "c:\autoexec.bat",
"c:\test\autoexec.sav"
SetAttr "c:\test\autoexec.sav" ,0
Filecopy "c:\autoexec.bat", "c:\test\autoexec.sav"
SetAttr "c:\test\autoexec.sav" ,1
print GetAttr( "c:\test\autoexec.sav" )
end
ErrorHandler:
Print Error
end
end sub
```

Příkaz Kill [Runtime]

Smaže soubor z disku.

Syntaxe:

```
Kill File As String
```

Parametry:

File: Řetězec určující jednoznačné (žádné zástupné znaky) jméno souboru. Také je možné použít URL notaci.

Chybové kódy

5 Neplatné volání procedury
76 Adresář nenalezen

Příklad:

```
sub ExampleKill
Kill "C:\datafile.dat" REM Soubor musí předem existovat
end sub
```

Funkce ConvertFromURL [Runtime]

Převede adresu URL souboru na název systémového souboru.

Úprava obsahu řetězce

Následující funkce upravují, formátují a zarovnávají obsah řetězce. Řetězce sjednotíte pomocí operátoru &.

Funkce Format [Runtime]

Převede číslo na řetězec a ten upraví podle zadaného formátu.

Funkce LCase [Runtime]

Převede všechna velká písmena na malá.

Funkce Left [Runtime]

Vrátí určený počet znaků ze začátku řetězce.

Příkaz LSet [Runtime]

Zarovná řetězec vlevo v řetězcové proměnné nebo zkopíruje proměnnou typu definovaného uživatelem do jiné proměnné jiného typu definovaného uživatelem.

Funkce LTrim [Runtime]

Odstraní všechny mezery na počátku řetězce.

Funkce Mid, příkaz Mid [Runtime]

Vrátí určenou část řetězce (**funkce Mid**) nebo nahradí část řetězce jiným řetězcem (**příkaz Mid**).

Funkce Right [Runtime]

Vrátí počet n znaků umístěných ve výrazu typu String nejvíce vpravo.

Příkaz RSet [Runtime]

Zarovná text typu String vpravo v rámci proměnné typu String nebo zkopíruje uživatelem definovaný typ proměnné do jiné.

Funkce RTrim [Runtime]

Smaže mezery na konci řetězce.

Funkce Trim [Runtime]

Odstraní všechny mezery ze začátku a konce řetězce.

Funkce UCase [Runtime]

Převede malá písmena na velká.

Funkce Split [Runtime]

Rozdělí řetězec na pole podřetězců.

Funkce Join [Runtime]

Vrátí řetězec složený z několika podřetězců umístěných v poli.

Funkce ConvertToURL [Runtime]

Převede název systémového souboru na adresu URL souboru.

Příkaz MkDir [Runtime]

Vytvoří na datovém médiu nový adresář.

Syntaxe:

```
MkDir Text As String
```

Parametry:

Text: Řetězec určující název a cestu nového adresáře. Také je možné použít URL notaci.
Pokud není určena cesta, adresář se vytvoří v aktuálním adresáři.

Chybové kódy

5 Neplatné volání procedury
76 Adresář nenalezen

Příklad:

```
Sub ExampleFileIO
' Příklad funkcí pro správu souborů
Const sFile1 as String = "file://c|/autoexec.bat"
Const sDir1 as String = "file://c|/Temp"
Const sSubDir1 as String ="Test"
Const sFile2 as String = "Copied.tmp"
Const sFile3 as String = "Renamed.tmp"
Dim sFile as String
sFile = sDir1 + "/" + sSubDir1
ChDir( sDir1 )
If Dir(sSubDir1,16)="" then ' Does the directory exist ?
MkDir sSubDir1
MsgBox sFile,0,"Vytvořit adresář"
End If
sFile = sFile + "/" + sFile2
FileCopy sFile1 , sFile
MsgBox fSysURL(CurDir()),0,"Aktuální adresář"
MsgBox sFile & Chr(13) & FileDateTime( sFile ),0,"Čas vytvoření"
MsgBox sFile & Chr(13)& FileLen( sFile ),0,"Délka souboru"
MsgBox sFile & Chr(13)& GetAttr( sFile ),0,"Atributy souboru"
Name sFile as sDir1 + "/" + sSubDir1 + "/" + sFile3
' Přejmenování ve stejném adresáři
sFile = sDir1 + "/" + sSubDir1 + "/" + sFile3
SetAttr( sFile, 0 ) 'Smaže všechny atributy
MsgBox sFile & Chr(13) & GetAttr( sFile ),0,"Nové atributy souboru"
Kill sFile
RmDir sDir1 + "/" + sSubDir1
end sub
' Převede systémovou cestu na URL
Function fSysURL( fSysFp as String ) as String
Dim iPos As String
iPos = 1
iPos = Instr(iPos,fSysFp, getPathSeparator())
do while iPos > 0
mid( fSysFp, iPos , 1,"/")
iPos = Instr(iPos+1,fSysFp, getPathSeparator())
loop
' the colon with DOS
iPos = Instr(1,fSysFp,":")
```

```
if iPos > 0 then mid( fSysFp, iPos , 1,"|")
fSysURL = "file://" & fSysFp
End Function
```

Funkce String [Runtime]

Vytvoří řetězec podle zadaného znaku nebo prvního znaku řetězce předaného funkci.

Syntaxe:

String (n As Long, {výraz As Integer | znak As String})

Návratová hodnota:

Řetězec

Parametry:

n: Číslo určující počet znaků v řetězci. Maximální dovolená hodnota je 65535.
Výraz: Číslo, které určuje ASCII kód znaku.
Znak: Jeden znak, ze kterého se vytvoří řetězec, nebo jakýkoliv řetězec, ze kterého se použije první znak.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleString
Dim sText as String
sText = String(10,"A")
Msgbox sText
sText = String(10,65)
Msgbox sText
End Sub
```

Funkce Space [Runtime]

Vrátí řetězec, který se skládá ze zadaného počtu mezer.

Syntaxe:

Space (n As Long)

Návratová hodnota:

Řetězec

Parametry:

n: Číslo určující počet mezer v řetězci. Maximální dovolená hodnota je 65535.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSpace
Dim sText As String,sOut As String
DIm iLen As Integer
iLen = 10
sText = "Las Vegas"
sOut = sText & Space(iLen) & sText & Chr(13) &_
sText & Space(iLen*2) & sText & Chr(13) &_
sText & Space(iLen*4) & sText & Chr(13)
msgBox sOut,0,"Info:"
End Sub
```

Příkaz Name [Runtime]

Přejmenuje stávající soubor nebo adresář.

Syntaxe:

```
Name OldName As String As NewName As String
```

Parametry:

OldName, NewName: Řetězce určující název souboru, včetně cesty. Také je možné použít URL notaci.

Příklad:

```
Sub ExampleReName
On Error Goto Error
Filecopy "c:\autoexec.bat", "c:\temp\autoexec.sav"
Name "c:\temp\autoexec.sav" as "c:\temp\autoexec.bat"
end
Error:
if err = 58 then
msgbox "Soubor již existuje"
end if
end
end sub
```

Příkaz RmDir [Runtime]

Odstraní stávající adresář z datového média.

Syntaxe:

```
RmDir Text As String
```

Parametry:

Text: Řetězec určující název a cestu adresáře, který chcete odstranit. Také je možné použít notaci URL.

Pokud není určena cesta, příkaz **RmDir** hledá určený adresář v aktuálním adresáři. Pokud zde není nalezen, zobrazí se chybová zpráva.

Chybové kódy

5 Neplatné volání procedury

76 Adresář nenalezen

Příklad:

```
Sub ExampleRmDir
MkDir "C:\Test2"
ChDir "C:\test2"
msgbox Curdir
ChDir "\"
RmDir "C:\test2"
end sub
```

Opakování obsahu

Následující funkce opakují obsah řetězců.

Funkce Space [Runtime]

Vrátí řetězec, který se skládá ze zadaného počtu mezer.

Funkce String [Runtime]

Vytvoří řetězec podle zadaného znaku nebo prvního znaku řetězce předaného funkci.

Funkce CByte [Runtime]

Převede řetězec nebo číselný výraz na bajtový typ.

Syntaxe:

Cbyte(Výraz)

Návratová hodnota:

Bajt

Parametry:

Výraz: Řetězec nebo číslo.

Chybové kódy

5 Neplatné volání procedury

Příkaz SetAttr [Runtime]

Nastaví atribut pro zadaný soubor.

Syntaxe:

```
SetAttr FileName As String, Attribute As Integer
```

Parametry:

FileName: Řetězec určující název a cestu souboru, kterému chcete nastavit atributy. Pokud není zadána cesta, **SetAttr** hledá soubor v aktuálním adresáři. Také je možné použít URL notaci.

Attribute: Bitový vzorek určující, které atributy chcete nastavit nebo zrušit:

Hodnota

0 : Normální soubory

1 : Soubory pouze pro zápis

32 : Soubor byl od posledního zálohování změněn (bit Archivovat).

Kombinací příslušných hodnot s logickým příkazem OR je možné nastavit více atributů.

Chybové kódy

5 Neplatné volání procedury

53 Soubor nenalezen

70 Přístup zamítnut

Příklad:

```
Sub ExampleSetGetAttr
On Error Goto ErrorHandler REM Určí zpracování chyb
If Dir("C:\test",16)="" Then MkDir "C:\test"
If Dir("C:\test\autoexec.sav")="" THEN Filecopy "c:\autoexec.bat",
"c:\test\autoexec.sav"
SetAttr "c:\test\autoexec.sav" ,0
Filecopy "c:\autoexec.bat", "c:\test\autoexec.sav"
SetAttr "c:\test\autoexec.sav" ,1
print GetAttr( "c:\test\autoexec.sav" )
end
ErrorHandler:
Print Error
end
end sub
```

Datové a časové funkce

Pomocí těchto příkazů a funkcí je možné pracovat s datovými a časovými údaji.

OpenOffice.org Basic vám umožňuje počítat rozdíly v datech a časech pomocí převodu datových a časových hodnot na číselná vyjádření. Po vypočtení rozdílu se použije speciální funkce pro převod získaných hodnot na standardní časové nebo datové formáty.

Datové a časové údaje je možné zkombinovat do jednoho desetinného čísla. Data se převedou na celá čísla a časy na desetinné hodnoty. Jazyk OpenOffice.org Basic také podporuje typ proměnné Date, která obsahuje určení času skládající se z data i času.

Převádění hodnot dat

Následující funkce převádějí datové hodnoty na číselné a zpět.

Převod hodnot času

Následující funkce převádějí časové hodnoty na číselné.

Systémové datum a čas

Následující funkce a příkazy nastavují nebo vrací systémové datum a čas.

Funkce Val [Runtime]

Převede řetězec na číslo.

Syntaxe:

Val (Text As String)

Návratová hodnota:

Double

Parametry:

Text: Řetězec představující číslo.

Pomocí funkce Val je možné převést řetězec, který představuje číslo, na číselnou hodnotu. Tato funkce je inverzní k funkci **Str**. Pokud obsahuje čísla pouze část řetězce, budou převedeny první odpovídající znaky řetězce. Pokud řetězec neobsahuje žádná čísla, funkce **Val** vrátí hodnotu 0.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleVal
msgbox Val("123.123")
msgbox Val("A123.123")
end Sub
```

Funkce Str [Runtime]

Převede číselný výraz na řetězec.

Syntaxe:

Str (Výraz)

Návratová hodnota:

Řetězec

Parametry:

Výraz: Jakýkoliv číselný výraz.

Funkce **Str** převádí číselnou proměnnou nebo výsledek výpočtu na řetězec. Před záporná čísla se přidá znak mínus. Před kladná čísla se přidá mezera (místo znaku plus).

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleStr
Dim iVar As Single
Dim sVar As String
iVar = 123.123
sVar = LTrim(Str(iVar))
Msgbox sVar & " " & dVar
end sub
```

Převádění hodnot dat

Následující funkce převádějí datové hodnoty na číselné a zpět.

Funkce DateSerial [Runtime]

Vrátí hodnotu **Date** pro určený rok, měsíc, nebo den.

Funkce DateValue [Runtime]

Vrací datovou hodnotu zadaného datového řetězce. Datový řetězec je kompletní datum. Toto číslo je také možné použít pro zjištění rozdílu mezi dvěma daty.

Funkce Day [Runtime]

Vrací hodnotu určující den v měsíci odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Funkce Month [Runtime]

Vrací hodnotu určující měsíc v roce odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Funkce WeekDay [Runtime]

Vrací hodnotu určující den v týdnu odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Funkce Year [Runtime]

Vrací hodnotu určující rok odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Funkce CDateToIso [Runtime]

Vrací datum v ISO formátu odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Funkce CDateFromIso [Runtime]

Vrátí interní číselnou hodnotu data z řetězce obsahujícího datum ve formátu ISO.

Funkce DateAdd [Runtime]

Přidá k určenému datu interval (se zadaným počtem opakování) a vrátí výsledné datum.

Funkce DateDiff [Runtime]

Vrátí počet intervalů mezi dvěma daty.

Funkce DatePart [Runtime]

Funkce DatePart vrací určenou část data.

Funkce DateSerial [Runtime]

Vrátí hodnotu **Date** pro určený rok, měsíc, nebo den.

Syntaxe:

DateSerial (year, month, day)

Návratová hodnota:

Date

Parametry:

Year: Číselná hodnota určující rok. Všechny hodnoty mezi 0 a 99 se interpretují jako roky 1900-1999. Pro roky mimo tento rozsah musíte použít čtyři číslice.

Month: Číselná hodnota určují měsíc v určeném roce. Povolený rozsah je 1-12.

Day: Číselná hodnota určující den určeného měsíce. Povolený rozsah je 1-31 a závisí na měsíci. Pokud zadáte neexistující číslo pro měsíc kratší než 31 dní, není vrácena žádná chyba.

Funkce **DateSerial** vrací počet dní mezi 30. prosincem 1899 a určeným datem. Pomocí této funkce je možné vypočítat rozdíl mezi dvěma daty.

Funkce **DateSerial** vrací typ Variant s VarType 7 (Date). Interně se tato hodnota ukládá jako Double, takže je-li zadáno datum 1.1.1900, vrátí hodnotu 2. Záporné hodnoty odpovídají datům před 30. prosincem 1899 (není zahrnut).

Pokud definované datum leží mimo povolený rozsah, OpenOffice.org Basic zobrazí chybovou zprávu.

Zatímco ve funkci **DateValue** zadáváte řetězec obsahující datum, funkce **DateSerial** zpracovává každý parametr (rok, měsíc, den) jako samostatný číselný výraz.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleDateSerial
Dim lDate as Long
Dim sDate as String
lDate = DateSerial(1964, 4, 9)
sDate = DateSerial(1964, 4, 9)
msgbox lDate REM vrací 23476
msgbox sDate REM vrací 04/09/1964
end sub
```

Funkce Chr [Runtime]

Vrátí znak odpovídající zadanému kódu.

Syntaxe:

Chr(Výraz As Integer)

Návratová hodnota:

Řetězec

Parametry:

Výraz: Číselná proměnná, která představuje platný 8bitový ASCII kód (0-255) nebo 16bitový Unicode kód.

Pomocí funkce **Chr$** je možné posílat speciální řídící znaky tiskárně nebo jinému výstupnímu zdroji. Také je možné ji použít pro vložení uvozovek do řetězce.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
sub ExampleChr
REM Tento příklad vloží do řetězce uvozovky (ASCII hodnota 34)
MsgBox "A "+ Chr$(34)+"short" + Chr$(34)+" trip."
REM Dialog se zobrazí jako: A "short" trip.
end sub
```

Související témata

ASC

Funkce Asc [Runtime]

Vrátí ASCII (American Standard Code for Information Interchange) prvního znaku v řetězci.

Syntaxe:

Asc (Text As String)

Návratová hodnota:

Celé číslo

Parametry:

Text: Platný řetězec. V úvahu se bere jen první znak řetězce.

Pomocí funkce Asc nahradíte kliče hodnotami. Pokud funkce Asc narazí na prázdný řetězec, OpenOffice.org Basic oznámí chybu. Kromě 7bitových ASCII znaků (kódy 0-127) umí funkce Asc zjistit také netisknutelné znaky ASCII kódu. Tato funkce zpracuje také 16bitové Unicode znaky.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleASC
Print ASC("A") REM vrací 65
Print ASC("Z") REM vrací 90
Print ASC("Las Vegas") REM vrací 76, protože se bere v úvahu pouze první znak
End Sub
```

Související témata

CHR

Funkce DateValue [Runtime]

Vrací datovou hodnotu zadaného datového řetězce. Datový řetězec je kompletní datum. Toto číslo je také možné použít pro zjištění rozdílu mezi dvěma daty.

Syntaxe:

DateValue [(date)]

Návratová hodnota:

Datum

Parametry:

Date: Řetězec obsahující datum, jehož číselnou hodnotu chcete spočítat. Datum lze zadat v téměř jakémkoliv formátu.

Tuto funkci je možné použít pro převod dat mezi 1. prosincem 1582 a 31. prosincem 9999 na jednu číselnou hodnotu. Tuto číselnou hodnotu je poté možné použít pro zjištění rozdílu mezi daty. Je-li argument mimo povolený rozsah, OpenOffice.org Basic zobrazí chybovou zprávu.

Na rozdíl od funkce DateSerial, které předáváte rok, měsíc a den jako oddělné číselné hodnoty, předáváte funkci DateValue datum ve formátu "měsíc.[,]den.[,]rok".

Národní nastavení použité pro formátování čísel, data a měny v jazyce OpenOffice.org Basic je možné nastavit v **Nástroje - Volby - Jazyková nastavení - Jazyky**. Ve formátovacích kódech Basic se jako desetinný oddělovač používá vždy tečka (.), která se při zobrazení nahradí odpovídajícím znakem podle národního nastavení.

Totéž platí pro národní nastavení formátu data, času a měny. Formátovací kódy se interpretují a zobrazí podle aktuálního národního nastavení.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleDateValue
msgbox DateValue("12/02/1997")
end sub
```

Funkce Day [Runtime]

Vrací hodnotu určující den v měsíci odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Syntaxe:

Day (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselný výraz, který obsahuje datum v číselné podobě, ze kterého se určí den v měsíci.

Tato funkce je v podstatě opakem funkce DateSerial, protože vrací den v měsíci z číselné hodnoty vytvoření funkcí **DateSerial** nebo **DateValue**. Například výraz

Print Day (DateSerial(1994, 12, 20))

vrátí hodnotu 20.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
sub ExampleDay
Print "Day " & Day(DateSerial(1994, 12, 20)) & " of the month"
end sub
```

Převod ASCII/ANSI v řetězcích

Následující funkce převádejí řetězce mezi kódy ASCII a ANSI.

Funkce Asc [Runtime]

Vrátí ASCII (American Standard Code for Information Interchange) prvního znaku v řetězci.

Funkce Chr [Runtime]

Vrátí znak odpovídající zadanému kódu.

Funkce Str [Runtime]

Převede číselný výraz na řetězec.

Funkce Val [Runtime]

Převede řetězec na číslo.

Funkce CByte [Runtime]

Převede řetězec nebo číselný výraz na bajtový typ.

Řetězce

Následující funkce a příkazy zpracovávají a vracejí řetězce.
Pomocí řetězců je možné v programech OpenOffice.org Basic upravovat text.

Převod ASCII/ANSI v řetězcích

Následující funkce převádejí řetězce mezi kódy ASCII a ANSI.

Opakování obsahu

Následující funkce opakují obsah řetězců.

Úprava obsahu řetězce

Následující funkce upravují, formátují a zarovnávají obsah řetězce. Řetězce sjednotíte pomocí operátoru &.

Úprava délky řetězce

Následující funkce určují délku řetězce a porovnávají řetězce.

Funkce Month [Runtime]

Vrací hodnotu určující měsíc v roce odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Syntaxe:

Month (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselný výraz, který obsahuje datum v číselné podobě, ze kterého se určí měsíc v roce.

Tato funkce je v podstatě opakem funkce DateSerial, protože vrací měsíc v roce z číselné hodnoty vytvoření funkcí **DateSerial** nebo **DateValue**. Například výraz

Print Month(DateSerial(1994, 12, 20))

tedy vrátí hodnotu 12.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleMonth
MsgBox "" & Month(Now) ,64,"The current month"
End sub
```

Funkce WeekDay [Runtime]

Vrací hodnotu určující den v týdnu odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Syntaxe:

WeekDay (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselný výraz, který obsahuje datum v číselné podobě, ze kterého se určí den v týdnu (1-7).
Následující příklad určí po zadání data den v týdnu s použitím funkce WeekDay.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleWeekDay
Dim sDay As String
REM Vrátí a zobrazí den v týdnu
Select Case WeekDay( Now )
case 1
sDay="Neděle"
case 2
sDay="Pondělí"
case 3
sDay="Úterý"
case 4
sDay="Středa"
case 5
sDay="Čtvrtek"
case 6
sDay="Pátek"
case 7
sDay="Sobota"
End Select
msgbox "" + sDay,64,"Dnes je "
End Sub
```

Porovnávací operátory [Runtime]

Porovnávací operátory porovnávají dva výrazy. Výsledek je vrácen jako Booleovský výraz určující, zda porovnání bylo pravdivé (True, -1) nebo nepravdivé (False, 0).

Syntaxe:

Výsledek = Výraz1 { = | < | > | <= | >= } Výraz2

Parametry:

Výsledek: Booleovský výraz, který obsahuje výsledek porovnání (True nebo False)
Výraz1, Výraz2: Jakékoli číselné hodnoty nebo řetězce, které chcete porovnat.

Porovnávací operátory

= : Je rovno
< : Je menší než
> : Je větší než
<= : Je menší nebo rovno
>= : Je větší nebo rovno
<> : Není rovno

Příklad:

```
Sub ExampleUnequal
DIM sFile As String
DIM sRoot As String REM ' Kořenový adresář pro souborový vstup nebo výstup
sRoot = "c:\"
sFile = Dir$( sRoot ,22)
If sFile <> "" Then
Do
Msgbox sFile
sFile = Dir$
Loop Until sFile = ""
End If
End sub
```

Porovnávací operátory

Zde jsou popsány dostupné porovnávací operátory.

Porovnávací operátory [Runtime]

Porovnávací operátory porovnávají dva výrazy. Výsledek je vrácen jako Booleovský výraz určující, zda porovnání bylo pravdivé (True, -1) nebo nepravdivé (False, 0).

Funkce Year [Runtime]

Vrací hodnotu určující rok odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Syntaxe:

Year (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselný výraz, který obsahuje datum v číselné podobě, ze kterého se určí rok.

Tato funkce je v podstatě opakem funkce DateSerial, protože vrací rok z číselné hodnoty vytvoření funkcí **DateSerial** nebo **DateValue**. Například výraz

Print Year(DateSerial(1994, 12, 20))

vrátí hodnotu 1994.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleYear
MsgBox "" & Year(Now) ,64,"Current year"
End sub
```

Funkce CDateToIso [Runtime]

Vrací datum v ISO formátu odpovídající datu v číselné podobě (vytvořené **DateSerial** nebo **DateValue**).

Syntaxe:

CDateToIso(Number)

Návratová hodnota:

Řetězec

Parametry:

Number: Číselný výraz, který obsahuje datum v číselné podobě.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCDateToIso
MsgBox "" & CDateToIso(Now) ,64,"ISO Date"
End Sub
```

Funkce IsUnoStruct [Runtime]

Vrátí True, je-li daný objekt Uno struct.

Syntaxe:

IsUnoStruct(Uno type)

Návratová hodnota:

Bool

Parametry:

Uno type : A UnoObject

Příklad:

```
Sub Main
Dim bIsStruct
' Inicializuje službu
Dim oSimpleFileAccess
oSimpleFileAccess = CreateUnoService( "com.sun.star.ucb.SimpleFileAccess" )
bIsStruct = IsUnoStruct( oSimpleFileAccess )
MsgBox bIsStruct ' Zobrazí False, protože oSimpleFileAccess NENÍ struct
' Inicializace vlastnosti struct
Dim aProperty As New com.sun.star.beans.Property
msgid "bIsStruct = IsUnoStruct( aProperty )"
MsgBox bIsStruct ' Zobrazí True, protože aProperty je struct
bIsStruct = IsUnoStruct( 42 )
MsgBox bIsStruct ' Zobrazí False, protože 42 NENÍ struct
End Sub
```

Funkce EqualUnoObjects [Runtime]

Vrátí True, pokud dva určené Basic Uno objekty představují stejnou instanci Uno objektu.

Syntaxe:

EqualUnoObjects(oObj1, oObj2)

Návratová hodnota:

Bool

Příklad:

```
// Kopie objektů -> stejná instance
oIntrospection = CreateUnoService( "com.sun.star.beans.Introspection" )
oIntro2 = oIntrospection
print EqualUnoObjects( oIntrospection, oIntro2 )
// Kopie struct jako hodnoty -> nová instance
Dim Struct1 as new com.sun.star.beans.Property
Struct2 = Struct1
print EqualUnoObjects( Struct1, Struct2 )
```

Funkce CDateFromIso [Runtime]

Vrátí interní číselnou hodnotu data z řetězce obsahujícího datum ve formátu ISO.

Syntaxe:

CDateFromIso(String)

Návratová hodnota:

Vnitřní číslo data

Parametry:

String: Řetězec obsahující datum ve formátu ISO. Rok může mít dvě nebo čtyři číslice.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
dateval = CDateFromIso("20021231")
```

vrátí 12/31/2002 v datovém formátu vašeho systému

Funkce DateAdd [Runtime]

Přidá k určenému datu interval (se zadaným počtem opakování) a vrátí výsledné datum.

Syntaxe:

DateAdd (Add, Count, Date)

Návratová hodnota:

Proměnná typu Variant, která obsahuje datum.

Parametry:

Add - Řetězec z následující tabulky, který určuje interval.

Add (řetězec)	Vysvětlení
yyyy	Rok
q	Čtvrtletí
m	Měsíc
y	Den v roce
w	Den v týdnu
ww	Týden v roce
d	Den
h	Hodina
n	Minuta
s	Sekunda

Count - Číselný výraz, který určuje, kolikrát se má interval přidat (Count je kladné) nebo odebrat (Count je záporné).

Date - Datum nebo název proměnné typu Variant obsahující datum. K této hodnotě se přidá určený počet intervalů.

Příklad:

```
Sub example_dateadd
msgbox DateAdd("m", 1, "1/31/2004") &" - "& DateAdd("m", 1, "1/31/2005")
End Sub
```

Funkce HasUnoInterfaces [Runtime]

Testuje, zda objekt Basic Uno podporuje určitá rozhraní Uno.
Vrátí True, pokud jsou podporována **všechna** uvedená Uno rozhraní, jinak vrátí False.

Syntaxe:

HasUnoInterfaces(oTest, NázevRozhraníUno1 [, NázevRozhraníUno2, ...])

Návratová hodnota:

Bool

Parametry:

oTest: Objekt Basic Uno, který chcete ověřit.
NázevRozhraníUno: Seznam názvů Uno rozhraní.

Příklad:

bHas = HasUnoInterfaces(oTest, "com.sun.star.beans.XIntrospection")

Funkce IsMissing [Runtime]

Testuje, zda byla funkce volána s volitelným parametrem.
Viz též: Optional

Syntaxe:

IsMissing(NázevArgumentu)

Parametry:

NázevArgumentu: Název nepovinného argumentu.
Je-li funkce IsMissing volána pomocí NázvuArgumentu, vrátí True.
Viz také: Příklady.

Chybové kódy

5 Neplatné volání procedury

Funkce DateDiff [Runtime]

Vrátí počet intervalů mezi dvěma daty.

Syntaxe:

DateDiff (Add, Date1, Date2 [, Week_start [, Year_start]])

Návratová hodnota:

Číslo.

Parametry:

Add - Řětezec z následující tabulky, který určuje interval.

Add (řetězec)	Vysvětlení
yyyy	Rok
q	Čtvrtletí
m	Měsíc
y	Den v roce
w	Den v týdnu
ww	Týden v roce
d	Den
h	Hodina
n	Minuta
s	Sekunda

Date1, Date2 - Dvě data k porovnání.
Week_start - Nepovinný parametr, který určuje první den v týdnu.

Hodnota Week_start	Vysvětlení
0	Použít výchozí systémovou hodnotu
1	Neděle (výchozí)
2	Pondělí
3	Úterý
4	Středa
5	Čtvrtek
6	Pátek
7	Sobota

Year_start - Nepovinný parametr, který určuje první týden v roce.

Hodnota Year_start	Vysvětlení
0	Použít výchozí systémovou hodnotu
1	První týden je ten, který obsahuje 1. leden (výchozí)
2	První týden je ten, který obsahuje alespoň čtyři dny roku
3	První týden je ten, který obsahuje pouze dny z nového roku

Příklad:

```
Sub example_datediff
msgbox DateDiff("d", "1/1/2005", "12/31/2005")
End Sub
```

Optional (v příkazu Function) [Runtime]

Umožňuje definovat nepovinné parametry funkce.
Viz též: IsMissing

Syntaxe:

Function MojeFunkce(Text1 As String, VolitelnýArgument2, VolitelnýArgument3)

Příklady:

Výsledek = MojeFunkce("Tady", 1, "Tam") ' jsou předány všechny argumenty
Výsledek = MojeFunkce("Test", ,1) ' chybí druhý argument
Viz také: Příklady.

Funkce FindPropertyObject [Runtime]

Touto funkcí lze prostřednictvím názvu objektu při provádění programu adresovat objekty jako parametr řetězce.
Následující příkaz:
MyObj.Prop1.Command = 5
například odpovídá následujícímu bloku příkazů:
Dim ObjVar as Object
Dim ObjProp as Object
ObjName As String = "MyObj"
ObjVar = FindObject(ObjName As String)
PropName As String = "Prop1"
ObjProp = FindPropertyObject(ObjVar, PropName As String)
ObjProp.Command = 5
Pro dynamické vytváření názvů za běhu použijte:
"TextEdit1" až TextEdit5" ve smyčce a tak vytvořit pět názvů.
Viz též: FindObject

Syntaxe:

FindPropertyObject(ObjektProměnná, NázevVlasnosti As String)

Parametry:

ObjektProměnná: Proměnná typu objekt, kterou chcete za běhu dynamicky definovat.
NázevVlastnosti: Řetězec určující název vlastnosti, kterou chcete za běhu adresovat.

Chybové kódy

5 Neplatné volání procedury
12 Nedefinovaná proměnná
14 Neplatný parametr
52 Špatný název nebo číslo souboru
57 Vstupně-výstupní chyba zařízení

Funkce DatePart [Runtime]

Funkce DatePart vrací určenou část data.

Syntaxe:

DatePart (Add, Date [, Week_start [, Year_start]])

Návratová hodnota:

Proměnná typu Variant, která obsahuje datum.

Parametry:

Add - Řětezec z následující tabulky, který určuje interval.

Add (řetězec)	Vysvětlení
yyyy	Rok
q	Čtvrtletí
m	Měsíc
y	Den v roce
w	Den v týdnu
ww	Týden v roce
d	Den
h	Hodina
n	Minuta
s	Sekunda

Date - Datum, ze kterého chcete vypočítat výsledek.
Week_start - Nepovinný parametr, který určuje první den v týdnu.

Hodnota Week_start	Vysvětlení
0	Použít výchozí systémovou hodnotu
1	Neděle (výchozí)
2	Pondělí
3	Úterý
4	Středa
5	Čtvrtek
6	Pátek
7	Sobota

Year_start - Nepovinný parametr, který určuje první týden v roce.

Hodnota Year_start	Vysvětlení
0	Použít výchozí systémovou hodnotu
1	První týden je ten, který obsahuje 1. leden (výchozí)
2	První týden je ten, který obsahuje alespoň čtyři dny roku
3	První týden je ten, který obsahuje pouze dny z nového roku

Příklad:

```
Sub example_datepart
msgbox DatePart("ww", "12/31/2005")
End Sub
```

Funkce FindObject [Runtime]

Umožňuje za běhu adresovat objekty pomocí parametru s názvem objektu.
Například následující příkaz:
MyObj.Prop1.Command = 5
odpovídá bloku příkazů:
Dim ObjVar as Object
Dim ObjProp as Object
ObjName As String = "MyObj"
ObjVar = FindObject(ObjName As String)
PropName As String = "Prop1"
ObjProp = FindPropertyObject(ObjVar, PropName As String)
ObjProp.Command = 5
Takto je možné dynamicky za běhu vytvářet názvy. Například:
"TextEdit1" až "TextEdit5" ve smyčce a vytvořit tak pět názvů ovládacích prvků.
Viz též: FindPropertyObject

Syntaxe:

FindObject(NázevObjektu As String)

Parametry:

NázevObjektu: Řetězec určující název objektu, který chcete za běhu adresovat.

Chybové kódy

5 Neplatné volání procedury
12 Nedefinovaná proměnná

Příkaz Set [Runtime]

Nastaví odkaz na objekt, proměnnou nebo vlastnost.

Syntaxe:

Set ObjektProm = Objekt

Parametry:

ObjektProm: Proměnná nebo vlastnost, která vyžaduje odkaz na objekt.
Objekt: Objekt, na který odkazuje proměnná nebo vlastnost.
Nothing - Přiřazením objektu **Nothing** proměnné odstraníte předchozí přiřazení.

Příklad:

```
Sub ExampleSet
Dim oDoc As Object
Set oDoc = ActiveWindow
Print oDoc.Name
End Sub
```

Převod hodnot času

Následující funkce převádějí časové hodnoty na číselné.

Funkce Hour [Runtime]

Vrátí hodinu z číselné časové hodnoty vygenerované funkcí TimeSerial nebo TimeValue.

Funkce Minute [Runtime]

Vrátí minuty z číselné časové hodnoty vygenerované funkcí TimeSerial nebo TimeValue.

Funkce Second [Runtime]

Vrátí sekundy z číselné časové hodnoty vygenerované funkcí TimeSerial nebo TimeValue.

Funkce TimeSerial [Runtime]

Vypočte číselnou časovou hodnotu ze zadaných hodin, minut a sekund v číselných parametrech. Tuto hodnotu je možné použít pro výpočet rozdílu mezi časy.

Funkce TimeValue [Runtime]

Ze zadané hodnoty hodin, minut a sekund (tyto parametry se zadávají jako řetězce) vypočítá číselnou hodnotu, která představuje časový údaj. Tuto hodnotu lze použít k výpočtu rozdílu mezi dvěma časy.

Funkce Hour [Runtime]

Vrátí hodinu z číselné časové hodnoty vygenerované funkcí TimeSerial nebo TimeValue.

Syntaxe:

Hour (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselný výraz, který obsahuje serializovanou hodnotu času, z něhož se vrátí hodiny.

Tato funkce je opakem funkce **TimeSerial**. Vrací číselnou hodnotu představující hodiny z číselné časové hodnoty vygenerované funkcí **TimeSerial** nebo **TimeValue**. Například výraz

Print Hour(TimeSerial(12,30,41))

tedy vrátí hodnotu 12.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleHour
Print "The current hour is " & Hour( Now )
End Sub
```

Funkce TypeName a VarType [Runtime]

Vrátí řetězec (TypeName) nebo číselnou hodnotu (VarType) obsahující informace o proměnné.

Syntaxe:

TypeName (Proměnná)VarType (Proměnná)

Návratová hodnota:

String; Integer

Parametry:

Proměnná: Proměnná, jejíž typ chcete určit. Možné je použít následující hodnoty:

klíčové slovo	VarType	Typ proměnné
Boolean	11	Logická proměnná
Date	7	Proměnná data
Double	5	Proměnná s dvojitou přesností a plovoucí desetinnou čárkou
Celé číslo	2	Celočíselná proměnná
Typu Long	3	Dlouhá celočíselná proměnná
Objekt	9	Objektová proměnná
Single	4	Proměnná s jednoduchou přesností a plovoucí desetinnou čárkou
Řetězec	8	Řetězec
Variant	12	Proměnná typu Variant (může obsahovat všechny typy a zadává se definicí)
Empty	0	Proměnná není inicializována
Null	1	Žádná platná data

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleType
Dim iVar As Integer
Dim sVar As String
Dim siVar As Single
Dim dVar As Double
Dim bVar As Boolean
Dim lVar As Long
Msgbox TypeName(iVar) & " " & VarType(iVar) & Chr(13) &_
TypeName(sVar) & " " & VarType(sVar) & Chr(13) &_
TypeName(siVar) & " " & VarType(siVar) & Chr(13) &_
TypeName(dVar) & " " & VarType(dVar) & Chr(13) &_
TypeName(bVar) & " " & VarType(bVar) & Chr(13) &_
TypeName(lVar) & " " & VarType(lVar),0,"Některé typy v OpenOffice.org Basic"
end Sub
```

Příkaz Static [Runtime]

Deklaruje proměnnou na úrovni podprogramu v rámci procedury nebo funkce, takže hodnota proměnné je platná i po ukončení procedury či funkce. Také platí konvence příkazu Dim.

Příkaz **Static** nelze použít k definici dynamických polí. Pole musí mít pevnou velikost.

Syntaxe:

Static NázevProměnné[(začátek To konec)] [As TypProměnné][, NázevProměnné2[(začátek To konec)] [As TypProměnné][,...]]

Příklad:

```
Sub ExampleStatic
Dim iCount as Integer, iResult as Integer
For iCount = 0 to 2
iResult = InitVar()
Next iCount
MsgBox iResult,0,"Výsledek je"
End Sub
REM Function for initialization of the static variable
Function InitVar() As Integer
Static iInit As Integer
Const iMinimum as Integer = 40 REM minimální návratová hodnota této funkce
if iInit = 0 then REM kontrola inicializace
iInit = iMinimum
else
iInit = iInit + 1
end if
InitVar = iInit
End Function
```

Funkce Minute [Runtime]

Vrátí minuty z číselné časové hodnoty vygenerované funkcí TimeSerial nebo TimeValue.

Syntaxe:

Minute (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselný výraz, který obsahuje serializovanou hodnotu času, z něhož se vrátí minuty.

Tato funkce je opakem funkce **TimeSerial**. Vrací číselnou hodnotu představující minuty z číselné časové hodnoty vygenerované funkcí **TimeSerial** nebo **TimeValue**. Například výraz

Print Minute(TimeSerial(12,30,41))

tedy vrátí hodnotu 30.

Chybové kódy

5 Neplatné volání procedury

Příklad:

Sub ExampleMinute

MsgBox "Právě je minut: "& Minute(Now)& "."

end sub

Funkce Second [Runtime]

Vrátí sekundy z číselné časové hodnoty vygenerované funkcí TimeSerial nebo TimeValue.

Syntaxe:

Second (Number)

Návratová hodnota:

Celé číslo

Parametry:

Number: Číselná podoba času, ze které se vrátí sekundy.

Tato funkce je opakem funkce **TimeSerial**. Vrací číselnou hodnotu představující sekundy z číselné časové hodnoty vygenerované funkcí **TimeSerial** nebo **TimeValue**. Například výraz

Print Second(TimeSerial(12,30,41))

tedy vrátí hodnotu 41.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSecond
MsgBox "Sekundy aktuálního času "& Second( Now )
End sub
```

Příkaz Global [Runtime]

Deklaruje proměnnou na globální úrovni (tj. ne do procedury nebo funkce), takže je přístupná ve všech knihovnách a modulech aktuálního sezení.

Syntaxe:

Global NázevProměnné[(začátek To konec)] [As TypProměnné][, NázevProměnné2[(začátek To konec)] [As TypProměnné][,...]]

Příklad:

```
Global iGlobalVar As Integer
Sub ExampleGlobal
iGlobalVar = iGlobalVar + 1
MsgBox iGlobalVar
End sub
```

Příkaz Public [Runtime]

Deklaruje proměnnou na úrovni modulu (tj. ne do procedury nebo funkce), takže je přístupná ve všech knihovnách a modulech.

Syntaxe:

Public NázevProměnné[(začátek To konec)] [As TypProměnné][, NázevProměnné2[(začátek To konec)] [As TypProměnné][,...]]

Příklad:

```
Public iPublicVar As Integer
Sub ExamplePublic
iPublicVar = iPublicVar + 1
MsgBox iPublicVar
End sub
```

Funkce TimeSerial [Runtime]

Vypočte číselnou časovou hodnotu ze zadaných hodin, minut a sekund v číselných parametrech. Tuto hodnotu je možné použít pro výpočet rozdílu mezi časy.

Syntaxe:

TimeSerial (hour, minute, second)

Návratová hodnota:

Date

Parametry:

hour: Číselná hodnota určující počet hodin. Platný rozsah: 0-23.

minute: Číselná hodnota určující počet minut. Obecně platí rozsah 0-59. Také je ovšem možné použít hodnoty ležící mimo tento rozsah, potom počet minut ovlivní i počet hodin.

second: Číselná hodnota určující počet sekund. Obecně platí rozsah 0-59. Také je ovšem možné použít hodnoty ležící mimo tento rozsah, potom počet sekund ovlivní i počet minut.

Příklady:

časový údaj 12, -5, 45 odpovídá údaji 11, 55, 45

časový údaj 12, 61, 45 odpovídá údaji 13, 2, 45

časový údaj 12, 20, -2 odpovídá údaji 12, 19, 58

časový údaj 12, 20, 63 odpovídá údaji 12, 21, 4

Funkci TimeSerial je možné použít k převodu času do jedné hodnoty, kterou je potom možné použít pro výpočet rozdílů času.

Funkce TimeSerial vrací typ Variant s VarType 7 (Date). Tato hodnota se interně ukládá jako hodnota Double v rozsahu 0 až 0,9999999999. Na rozdíl od funkcí DateSerial a DateValue, kde se rozdíl počítá jako počet dní k pevnému datu, je možné s hodnotami vrácenými funkcí TimeSerial počítat, ale není je možné převést zpět na čas.

Ve funkci TimeValue je možné zadat řetězec jako parametr, který obsahuje čas. Ve funkci TimeSerial je však možné jednotlivé parametry (hodiny, minuty, sekundy) zadávat jako samostatné číselné výrazy.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleTimeSerial
Dim dDate As Double, sDate As String
dDate = TimeSerial(8,30,15)
sDate = TimeSerial(8,30,15)
MsgBox dDate,64,"Time as a number"
MsgBox sDate,64,"Formatted time"
End Sub
```

Funkce TimeValue [Runtime]

Ze zadané hodnoty hodin, minut a sekund (tyto parametry se zadávají jako řetězce) vypočítá číselnou hodnotu, která představuje časový údaj. Tuto hodnotu lze použít k výpočtu rozdílu mezi dvěma časy.

Syntaxe:

TimeValue (Text As String)

Návratová hodnota:

Date

Parametry:

Text: Řetězec, který obsahuje čas ve formátu "HH:MM:SS".

Pomocí funkce TimeValue je možné převést jakýkoliv čas na jednu hodnotu, se kterou je možné počítat rozdíly časů.

Funkce TimeValue vrací typ Variant s VarType 7 (Date). Tato hodnota se interně ukládá jako hodnota Double v rozsahu 0 až 0,9999999999.

Na rozdíl od funkcí DateSerial a DateValue, kde se rozdíl počítá jako počet dní k pevnému datu, je možné s hodnotami vrácenými funkcí TimeValue počítat, ale není je možné převést zpět na čas.

Funkci TimeSerial je možné předávat jednotlivé parametry (hodiny, minuty, sekundy) jako samostatná čísla. Ovšem funkci TimeValue je možné předat řetězec jako parametr obsahující čas.

Chybové kódy

5 Neplatné volání procedury

13 Nesoulad typů

Příklad:

```
Sub ExampleTimerValue
Dim daDT as Date
Dim a1, b1, c1, a2, b2, c2 as String
a1 = "start time"
b1 = "end time"
c1 = "total time"
a2 = "8:34"
b2 = "18:12"
daDT = TimeValue(b2) - TimeValue(a2)
c2 = a1 & ": " & a2 & chr(13)
c2 = c2 & b1 & ": " & b2 & chr(13)
c2 = c2 & c1 & ": " & trim(Str(Hour(daDT))) & ":" & trim(Str(Minute(daDT))) & ":" & trim(Str(Second(daDT)))
Msgbox c2
end sub
```

Příkaz Option Explicit [Runtime]

Určuje, že je třeba každou proměnnou v programu explicitně deklarovat příkazem Dim.

Syntaxe:

Option Explicit

Parametry:

Tento příkaz musíte přidat před spustitelný kód v modulu.

Příklad:

```
Option Explicit
Sub ExampleExplicit
Dim sVar As String
sVar = "Las Vegas"
For i% = 1 to 10 REM Tento výsledek způsobí chybu
REM
Next i%
End Sub
```

Příkaz Option Base [Runtime]

Definuje nejnižší index pro pole jako 0 nebo 1.

Syntaxe:

Option Base { 0 | 1}

Parametry:

Tento příkaz musíte přidat před spustitelný kód v modulu.

Příklad:

```
option Base 1
Sub ExampleOptionBase
Dim sVar(20) As String
msgbox LBound(sVar())
end sub
```

Systémové datum a čas

Následující funkce a příkazy nastavují nebo vrací systémové datum a čas.

Příkaz Date [Runtime]

Vrací aktuální systémové datum jako řetězec nebo nastaví datum. Formát data závisí na místním nastavení systému.

Funkce Now [Runtime]

Vrátí aktuální systémové datum a čas jako hodnotu typu **Date**.

Příkaz Time [Runtime]

Tato funkce vrátí aktuální systémový čas jako řetězec ve formátu "HH:MM:SS".

Funkce Timer [Runtime]

Vrátí hodnotu, která určuje, kolik sekund uběhlo od půlnoci.

Příkaz Date [Runtime]

Vrací aktuální systémové datum jako řetězec nebo nastaví datum. Formát data závisí na místním nastavení systému.

Syntaxe:

Date ; Date = Text typu String

Parametry:

Text: Vyžadován pouze pro nastavení data. V takovém případě musí jeho obsah odpovídat formátu data definovanému místním nastavením systému.

Příklad:

```
Sub ExampleDate
msgbox "Aktuální datum je " & Date
end sub
```

Funkce Erase [Runtime]

Ruší obsah elementů polí s pevnou velikostí a uvolňuje paměť využitou poli s dynamickou velikostí.

Syntaxe:

Erase Arraylist

Parametry:

Arraylist - Seznam polí, která se mají smazat.

Funkce DimArray [Runtime]

Vrátí pole typu Variant.

Syntaxe:

DimArray (Seznam argumentů)

Viz také Array

Pokud nejsou předány žádné parametry, vytvoří se prázdné pole (Jako Dim A(), což odpovídá sekvenci délky 0 v Uno). Pokud jsou určeny parametry, je pro každý parametr vytvořen nový prvek.

Parametry:

Seznam argumentů: Seznam libovolného počtu argumentů oddělených čárkami.

Chybové kódy

9 Index mimo rozsah

Příklad:

DimArray(2, 2, 4) je stejné jako DIM a(2, 2, 4)

Funkce Now [Runtime]

Vrátí aktuální systémové datum a čas jako hodnotu typu **Date**.

Syntaxe:

Now

Návratová hodnota:

Date

Příklad:

```
Sub ExampleNow
msgbox "Právě je " & Now
End sub
```

Příkaz Time [Runtime]

Tato funkce vrátí aktuální systémový čas jako řetězec ve formátu "HH:MM:SS".

Syntaxe:

Čas

Parametry:

Text: Řetězec udávající nový čas ve formátu "HH:MM:SS".

Příklad:

```
Sub ExampleTime
MsgBox Time,0,"Aktuální čas je"
end sub
```

Funkce Array [Runtime]

Vrátí typ Variant s datovým polem.

Syntaxe:

Array (Seznam argumentů)
Viz také DimArray

Parametry:

Seznam argumentů: Seznam libovolného počtu argumentů oddělených čárkami.

Příklad:

```
Dim A As Variant
A = Array("Fred","Tom","Bill")
Msgbox A(2)
```

Příkaz Let [Runtime]

Přiřadí proměnné určitou hodnotu.

Syntaxe:

[Let] NázevProměnné=Výraz

Parametry:

NázevProměnné: Proměnná, které chcete přiřadit hodnotu. Typ hodnoty a proměnné musí být kompatibilní.

Stejně jako ve většině variant BASICu je klíčové slovo **Let** nepovinné.

Příklad:

```
Sub ExampleLen
Dim sText as String
Let sText = "Las Vegas"
msgbox Len(sText) REM vrací 9
End Sub
```

Funkce Timer [Runtime]

Vrátí hodnotu, která určuje, kolik sekund uběhlo od půlnoci.

Nejprve musíte deklarovat proměnnou a přiřadit jí datový typ "Long", jinak bude vrácena hodnota typu Date.

Syntaxe:

Timer

Návratová hodnota:

Date

Příklad:

```
Sub ExampleTimer
Dim lSec as long,lMin as long,lHour as long
lSec = Timer
MsgBox lSec,0,"Počet sekund od půlnoci"
lMin = lSec / 60
lSec = lSec Mod 60
lHour = lMin / 60
lMin = lMin Mod 60
MsgBox Right("00" & lHour , 2) & ":"& Right("00" & lMin , 2) & ":" & Right("00" & lSec , 2) ,0,"Právě je"
end sub
```

Funkce pro zpracovávání chyb

Pomocí následujících příkazů a funkcí je možné určit způsob, jakým bude OpenOffice.org Basic reagovat na chyby za běhu.
OpenOffice.org Basic nabízí několik způsobů, jak předejít ukončení programu, pokud nastane chyba.

Funkce Erl [Runtime]

Vrátí číslo řádku, na kterém došlo k chybě za běhu programu.

Funkce Err [Runtime]

Vrátí chybový kód identifikující chybu, k níž došlo za běhu programu.

Funkce Error [Runtime]

Vrátí chybovou zprávu odpovídající danému chybovému kódu.

Příkaz On Error GoTo ... Resume [Runtime]

Aktivuje rutinu pro zpracování chyb nebo obnoví spuštění programu, jakmile dojde k chybě.

Funkce UBound [Runtime]

Vrátí horní hranici pole.

Syntaxe:

UBound (NázevPole [, Rozměr])

Návratová hodnota:

Celé číslo

Parametry:

NázevPole: Název pole, jehož horní (**Ubound**) nebo dolní (**LBound**) hranici chcete zjistit.
[Rozměr]: Celé číslo určující, u kterého rozměru chcete zjistit horní (**Ubound**) nebo dolní (**LBound**) hranici. Pokud není hodnota určena, předpokládá se první rozměr.

Chybové kódy

5 Neplatné volání procedury
9 Index mimo rozsah

Příklad:

```
Sub ExampleUboundLbound
Dim sVar(10 to 20) As String
print LBound(sVar())
print UBound(sVar())
end Sub
Sub ExampleUboundLbound2
Dim sVar(10 to 20,5 To 70) As String
Print LBound(sVar()) REM Vrátí 10
Print UBound(sVar()) REM Vrátí 20
Print LBound(sVar(),2) REM Vrátí 5
Print UBound(sVar(),2) REM Vrátí 70
end Sub
```

Funkce LBound [Runtime]

Vrátí dolní hranici pole.

Syntaxe:

LBound (NázevPole [, Rozměr])

Návratová hodnota:

Celé číslo

Parametry:

NázevPole: Název pole, jehož horní (**Ubound**) nebo dolní (**LBound**) hranici chcete zjistit.

[Rozměr]: Celé číslo určující, u kterého rozměru chcete zjistit horní (**Ubound**) nebo dolní (**LBound**) hranici. Pokud není hodnota určena, předpokládá se první rozměr.

Chybové kódy

5 Neplatné volání procedury
9 Index mimo rozsah

Příklad:

```
Sub ExampleUboundLbound
Dim sVar(10 to 20) As String
print LBound(sVar())
print UBound(sVar())
end Sub
Sub ExampleUboundLbound2
Dim sVar(10 to 20,5 To 70) As String
Print LBound(sVar()) REM Vrátí 10
Print UBound(sVar()) REM Vrátí 20
Print LBound(sVar(),2) REM Vrátí 5
Print UBound(sVar(),2) REM Vrátí 70
end Sub
```

Funkce Erl [Runtime]

Vrátí číslo řádku, na kterém došlo k chybě za běhu programu.

Syntaxe:

Erl

Návratová hodnota:

Celé číslo

Parametry:

Funkce Erl pouze vrací číslo řádku, ne popis řádku.

Příklad:

```
sub ExampleError
on error goto ErrorHandler REM Set up error handler
Dim iVar as Integer
Dim sVar As String
REM Chyba způsobená neexistujícím souborem
iVar = Freefile
Open "\file9879.txt" for Input as #iVar
Line Input #iVar, sVar
Close #iVar
exit sub
ErrorHandler:
MsgBox "Chyba " & err & ": " & error$ + chr(13) + "Na řádku : " + Erl + chr(13) + Now , 16 ,"Nastala chyba"
end sub
```

Funkce Err [Runtime]

Vrátí chybový kód identifikující chybu, k níž došlo za běhu programu.

Syntaxe:

Err

Návratová hodnota:

Celé číslo

Parametry:

Funkcí Err se v rutinách pro zpracování chyb zjišťuje typ chyby a určuje způsob jejího odstranění.

Příklad:

```
sub ExampleError
on error goto ErrorHandler REM Set up error handler
Dim iVar as Integer
Dim sVar As String
REM Chyba kvůli neexistujícímu souboru
iVar = Freefile
Open "\file9879.txt" for Input as #iVar
Line Input #iVar, sVar
Close #iVar
exit sub
ErrorHandler:
MsgBox "Chyba " & Err & ": " & Error$ + chr(13) + "Na řádku : " + Erl + chr(13) + Now , 16 ,"Nastala chyba"
end sub
```

Funkce IsObject [Runtime]

Ověří, zda je proměnná typu object OLE objekt. Je-li proměnná OLE objekt, vrátí funkce True, jinak vrátí False.

Syntaxe:

IsObject (ObjectVar)

Návratová hodnota:

Bool

Parametry:

ObjectVar: Proměnná, kterou chcete ověřit. Pokud proměnná typu Object obsahuje OLE objekt, vrátí funkce True, jinak vrátí False.

Chybové kódy

5 Neplatné volání procedury

Funkce IsNumeric [Runtime]

Ověří, zda je výraz číslo. Je-li výraz číslo, vrátí funkce True, jinak vrátí False.

Syntaxe:

IsNumeric (Var)

Návratová hodnota:

Bool

Parametry:

Var: Proměnná, kterou chcete ověřit.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleIsNumeric
Dim vVar as variant
vVar = "ABC"
Print IsNumeric(vVar) REM Vrátí False
vVar = "123"
Print IsNumeric(vVar) REM Vrátí True
end sub
```

Funkce Error [Runtime]

Vrátí chybovou zprávu odpovídající danému chybovému kódu.

Syntaxe:

Error (Expression)

Návratová hodnota:

Řetězec

Parametry:

Expression: Číselná hodnota, která obsahuje chybový kód, k němuž chcete najít chybovou zprávu.

Pokud nejsou předány žádné parametry, funkce Error vrátí chybovou zprávu k chybě, která za běhu programu nastala naposledy.

Chybové kódy

2 Nespecifikovaná syntaktická chyba
3 Return bez Gosub
4 Znovu od začátku
5 Neplatné volání procedury
6 Přetečení
7 Nedostatek paměti
8 Velikost pole již byla určena
9 Index mimo rozsah
10 Dvojitá definice
11 Dělení nulou
12 Nedefinovaná proměnná
13 Nesoulad typů
14 Neplatný parametr
18 Došlo k přerušení uživatelem
20 Pokračovat bez chyby
28 Nedostatek místa v zásobníku
35 Procedura nebo funkce není definována
48 Chyba při načítání knihovny DLL
49 Špatné volání DLL
51 Interní chyba
52 Špatný název nebo číslo souboru
53 Soubor nenalezen
54 Špatný režim souboru
55 Soubor je již otevřen
57 Vstupně-výstupní chyba zařízení
58 Soubor již existuje
59 Špatná délka záznamu
61 Disk plný
62 Zápis za konec souboru
63 Špatné číslo záznamu
67 Příliš mnoho otevřených souborů
68 Zařízení není dostupné
70 Přístup zamítnut
71 Disk není připraven
73 Vlastnost není implementována
74 Nelze přesunout na jiný disk

75 Chyba přístupu k adresáři/souboru
76 Adresář nenalezen
91 Proměnná objektu není nastavena
93 Neplatný vzor řetězce
94 Neplatné použití Null
323 Není možné načíst modul
341 Neplatný index objektu
366 Žádný aktivní pohled nebo dokument
380 Špatná hodnota vlastnosti
382 Vlastnost je pouze pro čtení
394 Vlastnost je pouze pro zápis
420 Neplatný odkaz na objekt
423 Vlastnost nebo metoda nenalezena
424 Je vyžadován objekt
425 Neplatné použití objektu
430 Třída nepodporuje OLE
438 Objekt nepodporuje metodu
440 Chyba OLE propojení
445 Objekt nepodporuje tuto akci
446 Objekt nepodporuje pojmenované argumenty
447 Objekt nepodporuje současné národní nastavení
448 Pojmenovaný argument nenalezen
449 Argument je povinný
450 Špatný počet argumentů
451 Objekt není kolekce
452 Neplatné pořadí
453 Určená DLL funkce nenalezena
460 Neplatný formát schránky

Funkce IsNull [Runtime]

Ověří, zda proměnná typu Variant obsahuje speciální hodnotu Null, která sděluje, že proměnná neobsahuje data.

Syntaxe:

IsNull (Var)

Návratová hodnota:

Bool

Parametry:

Var: Proměnná, kterou chcete ověřit. Pokud Variant obsahuje hodnotu Null, funkce vrátí **True**, jinak vrátí **False** .
Null - Tato hodnota se používá pro vyjádření neplatné hodnoty.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleIsNull
Dim vVar As Variant
msgbox IsNull(vVar)
end sub
```

Funkce IsError [Runtime]

Testuje, zda proměnná obsahuje chybovou hodnotu.

Syntaxe:

IsError (Var)

Návratová hodnota:

Bool

Parametry:

Jakákoliv proměnná, kterou chcete testovat. Pokud proměnná obsahuje chybovou hodnotu, pak funkce vrací True, v ostatních případech vrací False.

Příkaz On Error GoTo ... Resume [Runtime]

Aktivuje rutinu pro zpracování chyb nebo obnoví spuštění programu, jakmile dojde k chybě.

Syntaxe:

On {Error GoTo Labelname | GoTo 0 | Resume Next}

Parametry:

GoTo Labelname: Pokud nastane chyba, spustí podprogram pro zpracování chyb, který začíná na řádku "Labelname".

Resume Next: Pokud nastane chyba, pokračuje vykonání programu dalším příkazem, který následuje za příkazem, u něhož nastala chyba.

GoTo 0: Vypne zpracování chyb v současné proceduře.

Příkaz On Error GoTo se používá k reakci na chyby, které nastanou v makru. Příkaz musí být vložen na začátek procedury (pro lokální zpracování chyb) nebo na začátek modulu.

Příklad:

```
Sub ExampleReset
On Error Goto ErrorHandler
Dim iNumber As Integer
Dim iCount As Integer
Dim sLine As String
Dim aFile As String
aFile = "c:\data.txt"
iNumber = Freefile
Open aFile For Output As #iNumber
Print #iNumber, "Toto je řádek textu"
Close #iNumber
iNumber = Freefile
Open aFile For Input As iNumber
For iCount = 1 to 5
Line Input #iNumber, sLine
If sLine <>"" then
rem
end if
Next iCount
Close #iNumber
Exit Sub
ErrorHandler:
Reset
MsgBox "Všechny soubory budou zavřeny",0,"Error"
End Sub
```

Logické operátory

OpenOffice.org Basic podporuje následující logické operátory.

Logické operátory porovnávají po bitech obsah dvou proměnných nebo výrazů, např. k ověření, zda jsou určité bity nastaveny nebo ne.

Operátor AND [Runtime]

Logické násobení dvou výrazů.

Operátor Eqv [Runtime]

Spočítá logickou ekvivalenci dvou výrazů.

Operátor Imp [Runtime]

Provede logickou implikaci mezi dvěma výrazy.

Operátor Not [Runtime]

Neguje výraz převrácením hodnot bitů.

Operátor Or [Runtime]

Provede logický součet dvou výrazů.

Operátor Xor [Runtime]

Provede logickou operaci EXCLUSIVE OR mezi dvěma výrazy.

Funkce IsEmpty [Runtime]

Ověří, zda proměnná typu Variant obsahuje hodnotu Empty. Hodnota Empty říká, že tato proměnná není inicializována.

Syntaxe:

IsEmpty (Var)

Návratová hodnota:

Bool

Parametry:

Var: Proměnná, kterou chcete ověřit. Pokud Variant obsahuje hodnotu Empty, funkce vrátí **True**, jinak vrátí **False** .

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleIsEmpty
Dim sVar as Variant
sVar = Empty
Print IsEmpty(sVar) REM Returns True
end sub
```

Funkce IsDate [Runtime]

Ověří, zda lze řetězec nebo číselný výraz převést na typ **Date**.

Syntaxe:

IsDate (Výraz)

Návratová hodnota:

Bool

Parametry:

Výraz: Číselný nebo řetězcový výraz, který chcete ověřit. Pokud lze výraz převést na datum, funkce vrátí **True**, jinak vrátí **False** .

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleIsDate
Dim sDateVar as String
sDateVar = "12.12.1997"
print IsDate(sDateVar) REM Returns True
sDateVar = "12121997"
print IsDate(sDateVar) REM Returns False
end sub
```

Operátor AND [Runtime]

Logické násobení dvou výrazů.

Syntaxe:

Výsledek = Výraz1 And Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.
Booleovské výrazy násobené pomocí AND vrací hodnotu **True**, pouze pokud jsou oba výrazy **True**.
True AND **True** vrací **True**; jiné kombinace vždy vrací **False**.
Použitím operátoru AND lze také provést porovnání stejně umístěných bitů ve dvou číselných výrazech.

Příklad:

```
Sub ExampleAnd
Dim A as Variant, B as Variant, C as Variant, D as Variant
Dim vVarOut as Variant
A = 10: B = 8: C = 6: D = Null
vVarOut = A > B And B > C REM vrací -1
vVarOut = B > A And B > C REM vrací 0
vVarOut = A > B And B > D REM vrací 0
vVarOut = (B > D And B > A) REM vrací 0
vVarOut = B And A REM vrátí 8 kvůli bitové kombinaci obou argumentů
End Sub
```

Operátor Eqv [Runtime]

Spočítá logickou ekvivalenci dvou výrazů.

Syntaxe:

Výsledek = Výraz1 Eqv Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.

Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Při ověřování ekvivalence mezi Booleovskými výrazy je výsledek **True** pokud jsou buď oba výrazy **True** nebo oba výrazy **False**.

Při bitovém porovnání nastaví operátor Eqv pouze odpovídající bit výsledku, pokud jsou bity nastaveny v obou výrazech nebo v žádném.

Příklad:

```
Sub ExampleEqv
Dim A as Variant, B as Variant, C as Variant, D as Variant
Dim vOut as Variant
A = 10: B = 8: C = 6: D = Null
vOut = A > B Eqv B > C REM vrací -1
vOut = B > A Eqv B > C REM vrací 0
vOut = A > B Eqv B > D REM vrací 0
vOut = (B > D Eqv B > A) REM vrací -1
vOut = B Eqv A REM vrací -3
End Sub
```

Funkce IsArray [Runtime]

Určuje, zda je proměnná pole.

Syntaxe:

IsArray (Var)

Návratová hodnota:

Bool

Parametry:

Var: Proměnná, u které chcete ověřit, zda jde o pole. Je-li proměnná pole, funkce vrátí **True**, jinak vrátí **False** .

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleIsArray
Dim sDatf(10) as String
print isarray(sdatf())
end Sub
```

Příkaz ReDim [Runtime]

Deklaruje proměnnou nebo pole.

Syntaxe:

[ReDim]Dim NázevProměnné [(začátek To konec)] [As TypProměnné][, NázevProměnné2 [(začátek To konec)] [As TypProměnné][,...]]

Volitelně je možné přidat klíčové slovo **Preserve**, které při změně velikosti pole zachová jeho obsah.

Parametry:

NázevProměnné: Platný název proměnné nebo pole.

Začátek, Konec: Číselné hodnoty nebo konstanty, které definují počet prvků (PočetPrvků=(konec-začátek)+1) a rozsah indexu.

Začátek a Konec musí být číselné výrazy, pokud se ReDim používá v podprogramu.

TypProměnné: Klíčové slovo určující datový typ proměnné.

Klíčové slovo: Typ proměnné

Bool: Booleovská proměnná (True, False)

Date: Datová proměnná

Double: Proměnná v plovoucí řádové čárce (1,79769313486232 x 10E308 - 4,94065645841247 x 10E-324)

Integer: Celočíselná proměnná (-32768 - 32767)

Long: Dlouhá celočíselná proměnná (-2147483648 - 2147483647)

Object: Objektová proměnná (Poznámka: tuto proměnnou lze následně definovat pomocí Set)

[Single]: Proměnná v plovoucí řádové čárce (3,402823 x 10E38 - 1,401298 x 10E-45). Pokud není zadáno klíčové slovo, je proměnná definována jako Single, pokud není použit příkaz DefBool až DefVar.

String: Řetězec obsahující maximálně 64000 ASCII znaků.

Variant: Proměnná typu Variant (může obsahovat všechny typy a určuje se definicí).

V jazyce OpenOffice.org Basic nemusíte proměnné deklarovat explicitně. Ovšem musíte před použitím deklarovat pole. Proměnnou je možné deklarovat pomocí příkazu Dim a použít čárky na oddělení několika deklarací. Chcete-li deklarovat typ proměnné, použijte znak typové deklarace nebo odpovídající klíčové slovo.

OpenOffice.org Basic podporuje jedno- i vícerozměrná pole, která se definují určeným typem proměnné. Pole jsou vhodná, pokud chcete v programu použít seznam či tabulku, které chcete upravovat. Výhodou polí je, že k jednotlivým prvkům je možné přistupovat pomocí indexů, které lze vyjádřit číselným výrazem nebo proměnnou.

Rozsah indexů pro pole deklarovaná příkazem Dim lze nastavit dvěma způsoby:

DIM text(20) As String REM 21 prvků číslovaných od 0 do 20

DIM text(5 to 25) As String REM 21 prvků číslovaných od 5 do 25

DIM text$(-15 to 5) As String REM 21 prvků (včetně 0)

REM číslováno od -15 do 5

Pole je možné deklarovat jako dynamické, pokud v podprogramu, kde chcete s polem pracovat, definujete rozměry pomocí příkazu ReDim. Obecně je možné rozměry určit jen jednou a poté je nelze měnit. V podprogramu je možné deklarovat pole pomocí ReDim. Rozměry je možné definovat pouze číselným výrazem.

Příklad:

```
Sub ExampleRedim
Dim iVar() As Integer, iCount As Integer
ReDim iVar(5) As integer
For iCount = 1 To 5
iVar(iCount) = iCount
Next iCount
ReDim iVar(10) As integer
For iCount = 1 To 10
iVar(iCount) = iCount
Next iCount
end sub
```

Operátor Imp [Runtime]

Provede logickou implikaci mezi dvěma výrazy.

Syntaxe:

Výsledek = Výraz1 Imp Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.

Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Pokud použijete operátor Imp v Booleovských výrazech, vrátí False pouze je-li první výraz True a druhý se vyhodnotí jako False.

Pokud použijete operátor Imp v bitovém porovnání, zruší se nastavení bitu ve výsledku, je-li odpovídající bit v prvním výraze nastaven a odpovídající bit ve druhém výraze nastaven není.

Příklad:

```
Sub ExampleImp
Dim A as Variant, B as Variant, C as Variant, D as Variant
Dim vOut as Variant
A = 10: B = 8: C = 6: D = Null
vOut = A > B Imp B > C REM vrací -1
vOut = B > A Imp B > C REM vrací -1
vOut = A > B Imp B > D REM vrací 0
vOut = (B > D Imp B > A) REM vrací -1
vOut = B Imp A REM vrací -1
End Sub
```

Operátor Not [Runtime]

Neguje výraz převrácením hodnot bitů.

Syntaxe:

Výsledek = Not Výraz

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz: Výraz, který chcete znegovat.
Při negaci Booleovského výrazu se hodnota True změní na False a hodnota False se změní na True.
Při negaci po jednotlivých bitech dojde k inverzi všech jednotlivých bitů.

Příklad:

```
Sub ExampleNot
Dim vA as Variant, vB as Variant, vC as Variant, vD as Variant
Dim vOut as Variant
vA = 10: vB = 8: vC = 6: vD = Null
vOut = Not vA REM vrací -11
vOut = Not(vC > vD) REM vrací -1
vOut = Not(vB > vA) REM vrací -1
vOut = Not(vA > vB) REM vrací 0
end Sub
```

```
sVar = "Office"
End Sub
Sub ExampleDim2
REM Dvourozměrné datové pole
Dim stext(20,2) as String
Const sDim as String = " Dimension:"
for i = 0 to 20
for ii = 0 to 2
stext(i,ii) = str(i) & sDim & str(ii)
next ii
next i
for i = 0 to 20
for ii = 0 to 2
msgbox stext(i,ii)
next ii
next i
End Sub
```

Příkaz Dim [Runtime]

Deklaruje proměnnou nebo pole.

Pokud jsou proměnné odděleny čárkou (např. DIM sPar1, sPar2, sPar3 AS STRING), lze definovat jen proměnné Variant. Každou proměnnou definujte na samostatném řádku.

```
DIM sPar1 AS STRING
DIM sPar2 AS STRING
DIM sPar3 AS STRING
```

Dim deklaruje místní proměnnou v podprogramu. Globální proměnné se deklarují příkazem PUBLIC nebo PRIVATE.

Syntaxe:

[ReDim]Dim NázevProměnné [(začátek To konec)] [As TypProměnné][, NázevProměnné2 [(začátek To konec)] [As TypProměnné][,...]]

Parametry:

NázevProměnné: Platný název proměnné nebo pole.

Začátek, Konec: Číselné hodnoty nebo konstanty, které definují počet prvků (PočetPrvků=(konec-začátek)+1) a rozsah indexu.

Začátek a Konec mohou být číselné výrazy, pokud se ReDim použije na úrovni procedury.

TypProměnné: Klíčové slovo určující datový typ proměnné.

Klíčové slovo: Typ proměnné

Bool: Booleovská proměnná (True, False)

Currency: Proměná měny (měna se 4 desetinnými místy)

Date: Datová proměnná

Double: Proměnná v plovoucí řádové čárce (1,79769313486232 x 10E308 - 4,94065645841247 x 10E-324)

Integer: Celočíselná proměnná (-32768 - 32767)

Long: Dlouhá celočíselná proměnná (-2147483648 - 2147483647)

Object: Objektová proměnná (Poznámka: tuto proměnnou lze následně definovat pomocí Set)

Single: Proměnná v plovoucí řádové čárce (3,402823 x 10E38 - 1,401298 x 10E-45).

String: Řetězec obsahující maximálně 64000 ASCII znaků.

[Variant]: Proměnná typu Variant (obsahuje všechny typy, určuje se v definici). Pokud není zadáno klíčové slovo, proměnné se automaticky přiřadí typ Variant, pokud se nepoužije příkaz DefBool až DefVar.

V jazyce OpenOffice.org Basic nemusíte proměnné deklarovat explicitně. Ovšem musíte před použitím deklarovat pole. Proměnnou je možné deklarovat pomocí příkazu Dim a použít čárky na oddělení několika deklarací. Chcete-li deklarovat typ proměnné, použijte znak typové deklarace nebo odpovídající klíčové slovo.

OpenOffice.org Basic podporuje jedno- i vícerozměrná pole, která se definují určeným typem proměnné. Pole jsou vhodná, pokud chcete v programu použít seznam či tabulku, které chcete upravovat. Výhodou polí je, že k jednotlivým prvkům je možné přistupovat pomocí indexů, které lze vyjádřit číselným výrazem nebo proměnnou.

Pole se deklarují příkazem Dim. Rozsah indexu lze definovat dvěma způsoby:

DIM text(20) as String REM 21 prvků číslovaných od 0 do 20

DIM text(5 to 25) as String REM 21 prvků číslovaných od 5 do 25

DIM text(-15 to 5) as String REM 21 prvků (včetně 0)

REM číslováno od -15 do 5

Dvourozměrné datové pole

DIM text(20,2) as String REM 63 prvků; od 0 do 20 úroveň 1, od 0 do 20 úroveň 2 a od 0 do 20 úroveň 3.

Pole je možné deklarovat jako dynamické, pokud v podprogramu, kde chcete s polem pracovat, definujete rozměry pomocí příkazu ReDim. Obecně lze rozměry určit jen jednou. Poté je nelze změnit. V podprogramu je možné deklarovat pole pomocí ReDim. Rozměry je možné definovat pouze číselným výrazem. To zaručuje, že jsou pole velká pouze tak, jak je třeba.

Příklad:

```
Sub ExampleDim1
Dim sVar As String
Dim iVar As Integer
```

Operátor Or [Runtime]

Provede logický součet dvou výrazů.

Syntaxe:

Výsledek = Výraz1 Or Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.

Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Logická disjunkce (OR) dvou Booleovských výrazů vrátí True, je-li alespoň jeden z porovnávaných výrazů True.

Při porovnání po jednotlivých bitech bude ve výsledku nastaven bit, je-li odpovídající bit nastaven alespoň v jednom z výrazů.

Příklad:

```
Sub ExampleOr
Dim vA as Variant, vB as Variant, vC as Variant, vD as Variant
Dim vOut as Variant
vA = 10: vB = 8: vC = 6: vD = Null
vOut = vA > vB Or vB > vC REM -1
vOut = vB > vA Or vB > vC REM -1
vOut = vA > vB Or vB > vD REM -1
vOut = (vB > vD Or vB > vA) REM 0
vOut = vB Or vA REM 10
End Sub
```

Operátor Xor [Runtime]

Provede logickou operaci EXCLUSIVE OR mezi dvěma výrazy.

Syntaxe:

Výsledek = Výraz1 Xor Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.

Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Logický součet Exclusive-Or dvou Booleovských výrazů vrátí True, pouze pokud se oba výrazy od sebe liší.

Při logickém součtu Exclusive-Or po jednotlivých bitech bude ve výsledku nastaven bit, je-li odpovídající bit nastaven pouze v jednom z výrazů.

Příklad:

```
Sub ExampleXor
Dim vA as Variant, vB as Variant, vC as Variant, vD as Variant
Dim vOut as Variant
vA = 10: vB = 8: vC = 6: vD = Null
vOut = vA > vB Xor vB > vC REM vrací 0
vOut = vB > vA Xor vB > vC REM vrací -1
vOut = vA > vB Xor vB > vD REM vrací -1
vOut = (vB > vD Xor vB > vA) REM vrací 0
vOut = vB Xor vA REM vrací 2
End Sub
```

Příkaz DefVar [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.

xxx: Klíčové slovo určující výchozí datový typ:

Klíčové slovo: Výchozí typ proměnné

DefVar: Variant

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefVar
vDiv=99 REM vDiv má výchozí typ variant
vDiv="Hello world"
end sub
```

Příkaz DefStr [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefStr nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefStr: Řetězec

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
DefStr s
Sub ExampleDefStr
sStr=String REM sStr je výchozí proměnná řetězce
end sub
```

Matematické operátory

OpenOffice.org Basic podporuje následující matematické operátory.
Tato kapitola poskytuje krátký přehled aritmetických operátorů, které se vám mohou hodit pro výpočty v programu.

Operátor "-" [Runtime]

Odečte jednu hodnotu od druhé.

Operátor "*" [Runtime]

Vynásobí dvě hodnoty.

Operátor "+" [Runtime]

Sečte nebo sloučí dva výrazy.

Operátor "/" [Runtime]

Vydělí jednu hodnotu druhou hodnotou.

Operátor "^" [Runtime]

Umocní číslo na zadanou mocninu.

Operátor Mod [Runtime]

Vrátí zbytek po dělení čísla.

Operátor "-" [Runtime]

Odečte jednu hodnotu od druhé.

Syntaxe:

Výsledek = Výraz1 - Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Příklad:

```
Sub ExampleSubtraction1
Print 5 - 5
End sub
Sub ExampleSubtraction2
Dim iValue1 as Integer
Dim iValue2 as Integer
iValue1 = 5
iValue2 = 10
Print iValue1 - iValue2
End Sub
```

Příkaz DefSng [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefSng nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefSng: Single

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
DefSng s
Sub ExampleDefSng
sSng=Single REM sSng je výchozí proměnná single
end sub
```

Příkaz DefObj [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefObj: Object

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
```

Operátor "*" [Runtime]

Vynásobí dvě hodnoty.

Syntaxe:

Výsledek = Výraz1 * Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Příklad:

```
Sub ExampleMultiplication1
Print 5 * 5
End sub
Sub ExampleMultiplication2
Dim iValue1 as Integer
Dim iValue2 as Integer
iValue1 = 5
iValue2 = 10
Print iValue1 * iValue2
End Sub
```

Operátor "+" [Runtime]

Sečte nebo sloučí dva výrazy.

Syntaxe:

Výsledek = Výraz1 + Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Příklad:

```
Sub ExampleAddition1
Print 5 + 5
End sub
Sub ExampleAddition2
Dim iValue1 as Integer
Dim iValue2 as Integer
iValue1 = 5
iValue2 = 10
Print iValue1 + iValue2
End Sub
```

Příkaz DefLng [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefLng: Long

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefLng
lCount=123456789 REM lCount má výchozí typ long integer
end sub
```

Příkaz DefInt [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefInt: Integer

Příklad:

```
REM Prefixová definice typů proměnných
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefInt
iCount=200 REM iCount má výchozí typ Integer
end sub
```

Operátor "/" [Runtime]

Vydělí jednu hodnotu druhou hodnotou.

Syntaxe:

Výsledek = Výraz1 / Výraz2

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Příklad:

```
Sub ExampleDivision1
Print 5 / 5
End sub
Sub ExampleDivision2
Dim iValue1 as Integer
Dim iValue2 as Integer
iValue1 = 5
iValue2 = 10
Print iValue1 / iValue2
End Sub
```

Operátor "^" [Runtime]

Umocní číslo na zadanou mocninu.

Syntaxe:

Výsledek = Výraz ^ Exponent

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz: Číselná hodnota, kterou chcete umocnit.
Exponent: Číselná hodnota, na kterou chcete umocnit výraz.

Příklad:

```
Sub Example
Print ( 12.345 ^ 23 )
Print Exp ( 23 * Log( 12.345 ) ) REM Umocnění pomocí logaritmu
End Sub
```

Příkaz DefInt [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefInt: Integer

Příklad:

```
REM Prefixová definice typů proměnných
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefInt
iCount=200 REM iCount má výchozí typ Integer
end sub
```

Příkaz DefErr [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefErr nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefErr: Chyba

Příklad:

REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
DefErr e
Sub ExampleDefErr
eErr=Error REM eErr je výchozí proměnná chyby
end sub

Operátor Mod [Runtime]

Vrátí zbytek po dělení čísla.

Syntaxe:

Výsledek = Výraz1 MOD Výraz2

Návratová hodnota:

Celé číslo

Parametry:

Výsledek: Číselná proměnná, do které se uloží výsledek operace.
Výraz1, Výraz2: Výrazy, které chcete použít jako vstup operace.

Příklad:

sub ExampleMod
print 10 mod 2.5 REM vrací 0
print 10 / 2.5 REM vrací 4
print 10 mod 5 REM vrací 0
print 10 / 5 REM vrací 2
print 5 mod 10 REM vrací 5
print 5 / 10 REM vrací 0.5
end sub

Číselné funkce

Následující číselné funkce provádějí výpočty. Matematické a logické operátory jsou popsány v samostatné části. Funkce se od operátorů liší tím, že funkci předáváte parametry a vrací výsledek, zatímco operátory vrací výsledek spojením dvou číselných výrazů.

Trigonometrické funkce

OpenOffice.org Basic podporuje následující trigonometrické funkce.

Exponenciální a logaritmické funkce

OpenOffice.org Basic podporuje následující exponenciální a logaritmické funkce.

Generování náhodných čísel

Následující příkazy a funkce generují náhodná čísla.

Výpočet druhé odmocniny

Pro výpočet druhých odmocnin použijte tuto funkci.

Celá čísla

Následující funkce zaokrouhlují hodnoty na celá čísla.

Absolutní hodnoty

Tato funkce vrací absolutní hodnoty.

Znaménka výrazů

Tato funkce vrací algebraické znaménko (signum) číselného výrazu.

Převádění čísel

Následující funkce převádí čísla z jednoho formátu do jiného.

Příkaz DefDbl [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefDbl: Double

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefDBL
dValue=1.23e43 REM dValue je výchozí proměnná typu Double
end sub
```

Příkaz DefDate [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefDate nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefDate: Datum

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefDate
tDate=Date REM tDate je výchozí proměnná Date
end sub
```

Trigonometrické funkce

OpenOffice.org Basic podporuje následující trigonometrické funkce.

Funkce Atn [Runtime]

Trigonometrická funkce, která vrátí arkustangens číselného výrazu. Vrácená hodnota je v rozsahu -Pi/2 až +Pi/2.

Funkce Cos [Runtime]

Vypočítá kosinus úhlu. Úhel se zadává v radiánech. Výsledek je v rozsahu -1 až 1.

Funkce Sin [Runtime]

Vrátí sinus úhlu. Úhel se zadává v radiánech. Výsledek je v rozsahu -1 až 1.

Funkce Tan [Runtime]

Určí tangens úhlu. Úhel se zadává v radiánech.

Funkce Atn [Runtime]

Trigonometrická funkce, která vrátí arkustangens číselného výrazu. Vrácená hodnota je v rozsahu -Pi/2 až +Pi/2.

Arkustanges je inverzní funkce k tangens. Funkce Atn vrací úhel "alfa", v radiánech, s použitím tangens tohoto úhlu. Funkce také může vrátit úhel "alfa" porovnáním délky protilehlé a přilehlé odvěsny v pravoúhlém trojúhelníku.

Atn(protilehlá odvěsna/přilehlá odvěsna)= Alfa

Syntaxe:

Atn (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz, který představuje poměr dvou odvěsen trojúhelníku. Funkce Atn vrací odpovídající úhel v radiánech (arkustangens).

Chcete-li převést radiány na stupně, vynásobte je hodnotou 180/Pi.

hodnota ve stupních=(hodnota v radiánech*180)/Pi

hodnota radiánech=(hodnota ve stupních*Pi)/180

Pi je konstanta s hodnotou přibližně 3.14159.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
REM Následující příklad spočítá v pravoúhlém trojúhelníku
REM úhel alfa z tangens úhlu alfa:
Sub ExampleATN
REM zaokrouhlené Pi = 3.14159 je předdefinovaná konstanta
Dim d1 As Double
Dim d2 As Double
d1 = InputBox$ ("Zadejte délku odvěsny přilehlé úhlu: ","Přilehlá")
d2 = InputBox$ ("Zadejte délku odvěsny protilehlé úhlu: ","Protilehlá")
Print "Úhel alfa je "; (atn (d2/d1) * 180 / Pi); " stupňů"
End Sub
```

Příkaz DefCur [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefCur nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.

xxx: Klíčové slovo určující výchozí datový typ:

Klíčové slovo: Výchozí typ proměnné

DefCur: Měna

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
DefCur c
Sub ExampleDefCur
cCur=Currency REM cCur je výchozí proměnná měny
end sub
```

Příkaz DefBool [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefBool nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Syntaxe:

Defxxx RozsahZnaků1[, RozsahZnaků2[,...]]

Parametry:

RozsahZnaků: Znaky určující proměnné, pro které chcete nastavit výchozí datový typ.
xxx: Klíčové slovo určující výchozí datový typ:
Klíčové slovo: Výchozí typ proměnné
DefBool: Logické proměnné

Příklad:

```
REM Prefixová definice typů proměnných:
DefBool b
DefDate t
DefDbL d
DefInt i
DefLng l
DefObj o
DefVar v
Sub ExampleDefBool
bOK=TRUE REM bOK je výchozí Booleovská proměnná
end sub
```

Funkce Cos [Runtime]

Vypočítá kosinus úhlu. Úhel se zadává v radiánech. Výsledek je v rozsahu -1 až 1.
Funkce Cos počítá poměr délky přilehlé odvěsny a délky přepony pravoúhlého trojúhelníku.
Cos(Alfa) = Přilehlá odvěsna/Přepona

Syntaxe:

Cos (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz určující úhel v radiánech, pro který chcete spočítat kosinus.
Stupně lze převést na radiány vynásobením hodnoty ve stupních hodnotou Pi/180. Radiány lze převést na stupně vynásobením hodnoty v radiánech hodnotou 180/Pi.
hodnota ve stupních=(hodnota v radiánech*180)/Pi
hodnota radiánech=(hodnota ve stupních*Pi)/180
Pi je konstanta s hodnotou přibližně 3.14159.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
REM Následující příklad umožňuje pro pravoúhlý trojúhelník
REM zadat odvěsnu a úhel (ve stupních) a spočítat délku přepony:
Sub ExampleCosinus
REM zaokrouhlené Pi = 3.14159
Dim d1 as Double, dAngle as Double
d1 = InputBox$ ("Zadejte délku přilehlé odvěsny: ","Adjacent")
dAngle = InputBox$ ("Zadejte úhel alfa (ve stupních): ","Alpha")
Print "Délka přepony je "; (d1 / cos (dAngle * Pi / 180))
End Sub
```

Funkce Sin [Runtime]

Vrátí sinus úhlu. Úhel se zadává v radiánech. Výsledek je v rozsahu -1 až 1.

Na základě úhlu Alfa vrátí funkce Sin v pravoúhlém trojúhelníku poměr délky protilehlé odvěsny k délce přepony.

Sin(Alfa) = Protilehlá odvěsna/Přepona

Syntaxe:

Sin (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz určující úhel v radiánech, pro který chcete spočítat sinus.

Stupně lze převést na radiány vynásobením hodnoty ve stupních hodnotou Pi/180 a radiány lze převést na stupně vynásobením hodnoty v radiánech hodnotou 180/Pi.

hodnota ve stupních=(hodnota v radiánech*180)/Pi

hodnota v radiánech=(hodnota ve stupních*Pi)/180

Pi je přibližně rovno 3,141593.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
REM V tomto příkladu je možný následující záznam pro pravoúhlý trojúhelník:
REM Z protilehlé odvěsny a úhlu (ve stupních) bude vypočtena délka přepony:
Sub ExampleSine
REM Pi = 3.1415926 je předdefinovaná proměnná
Dim d1 as Double
Dim dAlpha as Double
d1 = InputBox$ ("Zadejte délku protilehlé odvěsny: ","Protilehlá odvěsna")
dAlpha = InputBox$ ("Zadejte úhel alfa (ve stupních): ","Alfa")
Print "Délka přepony je"; (d1 / sin (dAlpha * Pi / 180))
End Sub
```

Funkce CVErr [Runtime]

Konvertuje řetězec nebo číselnou hodnotu na podtyp typu Variant "Error".

Syntaxe:

CVErr(Výraz)

Návratová hodnota:

Variant.

Parametr:

Výraz: Jakýkoliv řetězec nebo číslo, které chcete převést.

Funkce CVar [Runtime]

Konvertuje řetězec nebo číselnou hodnotu na typ Variant.

Syntaxe:

CVar(Výraz)

Návratová hodnota:

Variant.

Parametr:

Výraz: Jakýkoliv řetězec nebo číslo, které chcete převést.

Funkce Tan [Runtime]

Určí tangens úhlu. Úhel se zadává v radiánech.

Na základě úhlu Alfa funkce Tan vypočítá poměr délky protilehlé odvěsny k délce přilehlé odvěsny v pravoúhlém trojúhelníku.

Tan (Alfa) = Protilehlá odvěsna/Přilehlá odvěsna

Syntaxe:

Tan (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz určující úhel v radiánech, pro který chcete spočítat tangens.

Stupně lze převést na radiány vynásobením hodnoty ve stupních hodnotou Pi/180. Radiány lze převést na stupně vynásobením hodnoty v radiánech hodnotou 180/Pi.

hodnota ve stupních=(hodnota v radiánech*180)/Pi

hodnota v radiánech=(hodnota ve stupních*Pi)/180

Pi je přibližně rovno 3,141593.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
REM V tomto příkladu je možný následující záznam pro pravoúhlý trojúhelník:
REM Z protilehlé odvěsny a úhlu (ve stupních) bude vypočtena délka přilehlé odvěsny:
Sub ExampleTangens
REM Pi = 3.1415926 je předdefinovaná konstanta
Dim d1 as Double
Dim dAlpha as Double
d1 = InputBox$ ("Zadejte délku odvěsny protilehlé danému úhlu: ","Protilehlá odvěsna")
dAlpha = InputBox$ ("Zadejte úhel alfa (ve stupních): ","Alfa")
Print "Délka odvěsny přilehlé danému úhlu je"; (d1 / tan (dAlpha * Pi / 180))
End Sub
```

Exponenciální a logaritmické funkce

OpenOffice.org Basic podporuje následující exponenciální a logaritmické funkce.

Funkce Exp [Runtime]

Vrátí základ přirozeného logaritmu (e = 2,718282) umocněný na zadané číslo.

Funkce Log [Runtime]

Vrátí přirozený logaritmus čísla.

Funkce CStr [Runtime]

Převede libovolný číselný výraz na řetězec.

Syntaxe:

CStr (Výraz)

Návratová hodnota:

Řetězec

Parametry:

Výraz: Platný řetězec nebo číselný výraz, který chcete převést.

Typy výrazů a výsledky převodu

Logická hodnota:	Řetězec obsahující **True** nebo **False**.
Datum:	Řetězec obsahující datum a čas.
Null:	Chyba.
Prázdná hodnota:	Prázdný řetězec.
Libovolná hodnota:	Odpovídající číslo jako řetězec.

Vrácený řetězec nebude obsahovat nuly, které jsou na konci čísla s plovoucí desetinnou čárkou.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCSTR
Dim sVar As String
Msgbox CDbl(1234.5678)
Msgbox CInt(1234.5678)
Msgbox CLng(1234.5678)
Msgbox CSng(1234.5678)
sVar = CStr(1234.5678)
MsgBox sVar
end sub
```

Funkce CSng [Runtime]

Převede libovolný řetězec nebo číselný výraz na datový typ Single.

Syntaxe:

CSng (Výraz)

Návratová hodnota:

Single

Parametry:

Výraz: Řetězec nebo číselný výraz, který chcete převést. Pro převod řetězce musíte číslo zadat jako normální text ("123.5") ve formátu, jaký používá váš operační systém.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCSNG
Msgbox CDbl(1234.5678)
Msgbox CInt(1234.5678)
Msgbox CLng(1234.5678)
Msgbox CSng(1234.5678)
end sub
```

Funkce Exp [Runtime]

Vrátí základ přirozeného logaritmu (e = 2,718282) umocněný na zadané číslo.

Syntaxe:

Exp (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz určující mocninu "e" (základ přirozeného logaritmu). Mocnina musí být pro typ Single menší nebo rovna 88,02969 a pro čísla typu Double menší nebo rovna 709,782712893, jinak OpenOffice.org Basic vrátí chybu přetečení.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleLogExp
Dim dValue as Double
const b1=12.345e12
const b2=1.345e34
dValue=Exp( Log(b1)+Log(b2) )
MsgBox "" & dValue & chr(13) & (b1*b2) ,0,"Násobení pomocí logaritmu"
end sub
```

Funkce Log [Runtime]

Vrátí přirozený logaritmus čísla.

Syntaxe:

Log (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz určující úhel v radiánech, pro který chcete spočítat přirozený logaritmus.

Přirozený logaritmus je logaritmus o základu e. "e" je konstanta s hodnotou přibližně 2,718282...

Logaritmus čísla x při libovolném základu n lze vypočítat tak, že následujícím způsobem vydělíte přirozený logaritmus čísla x přirozeným logaritmem čísla n:

Log n(x) = Log(x) / Log(n)

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleLogExp
Dim a as Double
Dim const b1=12.345e12
Dim const b2=1.345e34
a=Exp( Log(b1)+Log(b2) )
MsgBox "" & a & chr(13) & (b1*b2) ,0,"Multiplication by logarithm function"
end sub
```

Příkaz Const [Runtime]

Definuje řetězec jako konstantu.

Syntaxe:

Const Text = Výraz

Parametry:

Text: Název konstanty, který splňuje pravidla pro pojmenování proměnných.

Konstanta je proměnná, která pomáhá zlepšit čitelnost programu. Konstanty se nedefinují jako proměnné určitého typu, ale spíše se používají jako zástupné znaky. Konstantu je možné definovat jen jednou a nelze ji měnit. Pro definici konstant použijte následující příkazy:

CONST Jméno=Výraz

Typ výrazu není důležitý. Po spuštění programu je OpenOffice.org Basic interně převádí tak, aby při každém použití konstanta vyhovovala.

Příklad:

```
Sub ExampleConst
Const iVar = 1964
Msgbox iVar
Const sVar = "Program", dVar As Double = 1.00
Msgbox sVar & " " & dVar
end sub
```

Funkce CLng [Runtime]

Převede libovolný řetězec nebo číselný výraz na dlouhé celé číslo.

Syntaxe:

CLng (Výraz)

Návratová hodnota:

Typu Long

Parametry:

Výraz: Číselný výraz, který chcete převést. Pokud Výraz nespadá do rozsahu -2.147.483.648 až 2.147.483.647, OpenOffice.org Basic ohlásí chybu přetečení. Pro převod řetězce musíte číslo zadat jako normální text ("123.5") ve formátu, jaký používá váš operační systém.

Tato funkce vždy zaokrouhlí desetinnou část na nejbližší celé číslo.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCountryConvert
Msgbox CDbl(1234.5678)
Msgbox CInt(1234.5678)
Msgbox CLng(1234.5678)
end sub
```

Generování náhodných čísel

Následující příkazy a funkce generují náhodná čísla.

Příkaz Randomize [Runtime]

Inicializuje generátor náhodných čísel.

Funkce Rnd [Runtime]

Vrátí náhodné číslo mezi 0 a 1.

Příkaz Randomize [Runtime]

Inicializuje generátor náhodných čísel.

Syntaxe:

Randomize [Number]

Parametry:

Number: Číselná hodnota, která inicializuje generátor náhodných čísel. Pokud tento parametr vynecháte, použije generátor aktuální hodnotu systémového časovače.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleRandomize
Dim iVar As Integer, sText As String
Dim iSpectral(10) As Integer
Randomize 2^14-1
For iCount = 1 To 1000
iVar = Int((10 * Rnd) ) REM Rozsah 0 až 9
iSpectral(iVar) = iSpectral(iVar) +1
Next iCount
sText = " | "
For iCount = 0 To 9
sText = sText & iSpectral(iCount) & " | "
Next iCount
MsgBox sText,0,"Spectral Distribution"
end sub
```

Funkce CInt [Runtime]

Převede libovolný řetězec nebo číselný výraz na celé číslo.

Syntaxe:

CInt (Výraz)

Návratová hodnota:

Celé číslo

Parametry:

Výraz: Číselný výraz, který chcete převést. Pokud Výraz nespadá do rozsahu -32768 až 32767, OpenOffice.org Basic ohlásí chybu přetečení. Pro převod řetězce musíte číslo zadat jako normální text ("123.5") ve formátu, jaký používá váš operační systém.

Tato funkce vždy zaokrouhlí desetinnou část na nejbližší celé číslo.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCountryConvert
Msgbox CDbl(1234.5678)
Msgbox CInt(1234.5678)
Msgbox CLng(1234.5678)
end sub
```

Funkce CDbl [Runtime]

Převede libovolný řetězec nebo číselný výraz na hodnotu typu double.

Syntaxe

CDbl (Výraz)

Vrácená hodnota

Double

Parametry:

Výraz: Řetězec nebo číselný výraz, který chcete převést. Pro převod řetězce musíte číslo zadat jako normální text ("123.5") ve formátu, jaký používá váš operační systém.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCountryConvert
Msgbox CDbl(1234.5678)
Msgbox CInt(1234.5678)
Msgbox CLng(1234.5678)
end sub
```

Funkce Rnd [Runtime]

Vrátí náhodné číslo mezi 0 a 1.

Syntaxe:

Rnd [(Výraz)]

Návratová hodnota:

Double

Parametry:

Výraz: Číselný výraz, který určuje, jak generovat náhodná čísla.

Menší než nula: Vždy vrátí stejné náhodné číslo.

Větší než nula: Vrací další náhodné číslo v pořadí.

Nula: Vrátí náhodné číslo, které bylo vygenerováno naposledy.

Vynecháno: Vrací další náhodné číslo v pořadí.

Pokud pokaždé předáte funkci Rnd stejné číslo, vygeneruje se stejná sekvence náhodných čísel. To proto, že parametr Výraz se použije jako počáteční bod pro další číslo.

Funkce **Rnd** vrací jen hodnoty v rozsahu 0 až 1. Pro vygenerování náhodných celých čísel v daném rozsahu použijte vzorec z následujícího příkladu:

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleRandomSelect
Dim iVar As Integer
iVar = Int((15 * Rnd) -2)
Select Case iVar
Case 1 To 5
Print "Čísla od 1 do 5"
Case 6, 7, 8
Print "Čísla od 6 do 8"
Case Is > 8 And iVar < 11
Print "Větší než 8"
Case Else
Print "Mimo rozsah od 1 do 10"
End Select
end sub
```

Výpočet druhé odmocniny

Pro výpočet druhých odmocnin použijte tuto funkci.

Funkce Sqr [Runtime]

Vypočítá druhou odmocninu číselného výrazu.

Funkce CDec [Runtime]

Konvertuje řetězec nebo číslo do desítkové soustavy.

Syntaxe:

CDec(Výraz)

Návratová hodnota:

Číslo v desítkové soustavě.

Parametr:

Výraz: Jakýkoliv řetězec nebo číslo, které chcete převést.

Funkce CDate [Runtime]

Převede řetězcový nebo číselný výraz na datum.

Syntaxe:

CDate (Výraz)

Návratová hodnota:

Date

Parametry:

Výraz: Řetězec nebo číselný výraz, který chcete převést.

Při převádění řetězce musí být datum a čas zadáno ve formátu MM.DD.YYYY HH.MM.SS, jak určují konvence funkcí **DateValue** a **TimeValue**. V číselných výrazech představuje část vlevo od desetinné čárky datum (počítáno od 31. prosince 1899) a část vpravo od desetinné čárky představuje čas.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
sub ExampleCDate
MsgBox cDate(1000.25) REM 09.26.1902 06:00:00
MsgBox cDate(1001.26) REM 09.27.1902 06:14:24
end sub
```

Funkce Sqr [Runtime]

Vypočítá druhou odmocninu číselného výrazu.

Syntaxe:

Sqr (Number)

Návratová hodnota:

Double

Parametry:

Number: Číselný výraz, jehož druhou odmocninu chcete spočítat.

Druhá odmocnina je číslo, které po vynásobení samo sebou dá výchozí číslo. Např. druhá odmocnina 36 je 6.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSqr
Dim iVar As Single
iVar = 36
Msgbox Sqr(iVar)
end sub
```

Celá čísla

Následující funkce zaokrouhlují hodnoty na celá čísla.

Funkce Fix [Runtime]

Vrátí celočíselnou hodnotu číselného výrazu, z něhož odstraní zlomkovou část.

Funkce Int [Runtime]

Vrátí celočíselnou část čísla.

Funkce CBool [Runtime]

Převede řetězcové nebo číselné porovnání na Booleovský výraz, nebo převede jeden číselný výraz na Booleovský výraz.

Syntaxe:

CBool (Výraz1 {= | <> | < | > | <= | >=} Výraz2) nebo CBool (Číslo)

Návratová hodnota:

Bool

Parametry:

Výraz1, Výraz2: Jakékoliv řetězce nebo čísla, které chcete porovnat. Pokud se výrazy shodují, funkce **CBool** vrátí **True**, jinak vrátí **False**.

Číslo: Číselný výraz, který chcete převést. Je-li výraz roven 0, vrátí se **False**, jinak vrátí **True**.

Následující příklad používá funkci **CBool** k vyhodnocení hodnoty vrácené funkcí **Instr**. Funkce kontroluje, zda je ve větě zadané uživatelem slovo "and".

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleCBool
Dim sText As String
sText = InputBox("Zadejte krátkou větu:")
REM Zjistí, zda se ve větě vyskytuje »and«
REM Namísto příkazu
REM If Instr(Input, "and")<>0 Then...
REM se použije funkce CBool takto:
If CBool(Instr(sText, "and")) Then
MsgBox "V zadané větě se vyskytuje slovo »and«!"
EndIf
End Sub
```

Funkce CCur [Runtime]

Konvertuje řetězec nebo číslo na měnu. Pro zobrazení oddělovačů a symbolů měny bude použito systémové nastavení.

Syntaxe:

CCur(Výraz)

Návratová hodnota:

Měna

Parametr:

Výraz: Jakýkoliv řetězec nebo číslo, které chcete převést.

Funkce Fix [Runtime]

Vrátí celočíselnou hodnotu číselného výrazu, z něhož odstraní zlomkovou část.

Syntaxe:

Fix (Výraz)

Návratová hodnota:

Double

Parametry:

Výraz: Číselný výraz, z něhož chcete vrátit celočíselnou hodnotu.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
sub ExampleFix
Print Fix(3.14159) REM vrací 3.
Print Fix(0) REM vrací 0.
Print Fix(-3.14159) REM vrací -3.
end sub
```

Funkce Int [Runtime]

Vrátí celočíselnou část čísla.

Syntaxe:

Int (Číslo)

Návratová hodnota:

Double

Parametry:

Číslo: Platný číselný výraz.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
sub ExampleINT
Print Int(3.99) REM vrací hodnotu 3.0
Print Int(0) REM vrací hodnotu 0.0
Print Int(-3.14159) REM vrací hodnotu -4.0
end sub
```

Příkaz Set [Runtime]

Nastaví odkaz na objekt, proměnnou nebo vlastnost.

Funkce FindObject [Runtime]

Umožňuje za běhu adresovat objekty pomocí parametru s názvem objektu.

Funkce FindPropertyObject [Runtime]

Touto funkcí lze prostřednictvím názvu objektu při provádění programu adresovat objekty jako parametr řetězce.

Optional (v příkazu Function) [Runtime]

Umožňuje definovat nepovinné parametry funkce.

Funkce IsMissing [Runtime]

Testuje, zda byla funkce volána s volitelným parametrem.

Funkce HasUnoInterfaces [Runtime]

Testuje, zda objekt Basic Uno podporuje určitá rozhraní Uno.

Funkce EqualUnoObjects [Runtime]

Vrátí True, pokud dva určené Basic Uno objekty představují stejnou instanci Uno objektu.

Funkce IsUnoStruct [Runtime]

Vrátí True, je-li daný objekt Uno struct.

Funkce IsError [Runtime]

Testuje, zda proměnná obsahuje chybovou hodnotu.

Funkce IsNull [Runtime]

Ověří, zda proměnná typu Variant obsahuje speciální hodnotu Null, která sděluje, že proměnná neobsahuje data.

Funkce IsNumeric [Runtime]

Ověří, zda je výraz číslo. Je-li výraz číslo, vrátí funkce True, jinak vrátí False.

Funkce IsObject [Runtime]

Ověří, zda je proměnná typu object OLE objekt. Je-li proměnná OLE objekt, vrátí funkce True, jinak vrátí False.

Funkce LBound [Runtime]

Vrátí dolní hranici pole.

Funkce UBound [Runtime]

Vrátí horní hranici pole.

Příkaz Let [Runtime]

Přiřadí proměnné určitou hodnotu.

Funkce Array [Runtime]

Vrátí typ Variant s datovým polem.

Funkce DimArray [Runtime]

Vrátí pole typu Variant.

Funkce Erase [Runtime]

Ruší obsah elementů polí s pevnou velikostí a uvolňuje paměť využitou poli s dynamickou velikostí.

Příkaz Option Base [Runtime]

Definuje nejnižší index pro pole jako 0 nebo 1.

Příkaz Option Explicit [Runtime]

Určuje, že je třeba každou proměnnou v programu explicitně deklarovat příkazem Dim.

Příkaz Public [Runtime]

Deklaruje proměnnou na úrovni modulu (tj. ne do procedury nebo funkce), takže je přístupná ve všech knihovnách a modulech.

Příkaz Global [Runtime]

Deklaruje proměnnou na globální úrovni (tj. ne do procedury nebo funkce), takže je přístupná ve všech knihovnách a modulech aktuálního sezení.

Příkaz Static [Runtime]

Deklaruje proměnnou na úrovni podprogramu v rámci procedury nebo funkce, takže hodnota proměnné je platná i po ukončení procedury či funkce. Také platí konvence příkazu Dim.

Funkce TypeName a VarType [Runtime]

Vrátí řetězec (TypeName) nebo číselnou hodnotu (VarType) obsahující informace o proměnné.

Absolutní hodnoty

Tato funkce vrací absolutní hodnoty.

Funkce Abs [Runtime]

Vrátí absolutní hodnotu číselného výrazu.

Funkce Abs [Runtime]

Vrátí absolutní hodnotu číselného výrazu.

Syntaxe:

Abs (Číslo)

Návratová hodnota:

Double

Parametry:

Číslo: Jakýkoli numerický výraz, jehož absolutní hodnotu chcete vrátit. Kladná čísla (včetně 0) jsou vrácena nezměněná, záporná čísla jsou převedena na kladná.

Následující příklad používá funkci Abs pro výpočet rozdílu mezi dvěma hodnotami. Nezáleží na tom, kterou hodnotu zadáte jako první.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleDifference
Dim siW1 As Single
Dim siW2 As Single
siW1 = Int(InputBox$ ("Zadejte, prosím, první hodnotu","Vstup hodnoty"))
siW2 = Int(InputBox$ ("Zadejte, prosím, druhou hodnotu","Vstup hodnoty"))
Print "Rozdíl je "; Abs(siW1 - siW2)
End Sub
```

Příkaz DefDate [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefDate nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefDbl [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefErr [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefErr nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefInt [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefLng [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefObj [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefSng [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefSng nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefStr [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefStr nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefVar [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz Dim [Runtime]

Deklaruje proměnnou nebo pole.

Příkaz ReDim [Runtime]

Deklaruje proměnnou nebo pole.

Funkce IsArray [Runtime]

Určuje, zda je proměnná pole.

Funkce IsDate [Runtime]

Ověří, zda lze řetězec nebo číselný výraz převést na typ **Date**.

Funkce IsEmpty [Runtime]

Ověří, zda proměnná typu Variant obsahuje hodnotu Empty. Hodnota Empty říká, že tato proměnná není inicializována.

Proměnné

Pomocí následujících příkazů a funkcí je možné pracovat s proměnnými. Tyto funkce je možné použít pro deklaraci a definování proměnných, převod proměnných mezi typy nebo pro určení typu proměnné.

Funkce CCur [Runtime]

Konvertuje řetězec nebo číslo na měnu. Pro zobrazení oddělovačů a symbolů měny bude použito systémové nastavení.

Funkce CBool [Runtime]

Převede řetězcové nebo číselné porovnání na Booleovský výraz, nebo převede jeden číselný výraz na Booleovský výraz.

Funkce CDate [Runtime]

Převede řetězcový nebo číselný výraz na datum.

Funkce CDec [Runtime]

Konvertuje řetězec nebo číslo do desítkové soustavy.

Funkce CDbl [Runtime]

Převede libovolný řetězec nebo číselný výraz na hodnotu typu double.

Funkce CInt [Runtime]

Převede libovolný řetězec nebo číselný výraz na celé číslo.

Funkce CLng [Runtime]

Převede libovolný řetězec nebo číselný výraz na dlouhé celé číslo.

Příkaz Const [Runtime]

Definuje řetězec jako konstantu.

Funkce CSng [Runtime]

Převede libovolný řetězec nebo číselný výraz na datový typ Single.

Funkce CStr [Runtime]

Převede libovolný číselný výraz na řetězec.

Funkce CVar [Runtime]

Konvertuje řetězec nebo číselnou hodnotu na typ Variant.

Funkce CVErr [Runtime]

Konvertuje řetězec nebo číselnou hodnotu na podtyp typu Variant "Error".

Příkaz DefBool [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefBool nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Příkaz DefCur [Runtime]

Pokud není určeno klíčové slovo nebo znak typové deklarace, příkaz DefCur nastaví výchozí datový typ pro proměnné podle rozsahu znaku.

Znaménka výrazů

Tato funkce vrací algebraické znaménko (signum) číselného výrazu.

Funkce Sgn [Runtime]

Vrací celé číslo mezi -1 a 1, které indikuje jestli zadané číslo bylo kladné, záporné nebo nula.

Funkce Sgn [Runtime]

Vrací celé číslo mezi -1 a 1, které indikuje jestli zadané číslo bylo kladné, záporné nebo nula.

Syntaxe:

Sgn (Číslo)

Návratová hodnota:

Celé číslo

Parametry:

Číslo: Číselná hodnota rozhodující o návratové hodnotě funkce.

Číslo	Vrácená hodnota
záporné	Funkce Sgn vrátí hodnotu -1.
0	Funkce Sgn vrátí hodnotu 0.
kladné	Funkce Sgn vrátí hodnotu 1.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSgn
Print sgn(-10) REM vrací -1
Print sgn(0) REM vrací 0
Print sgn(10) REM vrací 1
end sub
```

Příkaz With [Runtime]

Nastaví objekt jako výchozí objekt. Pokud není určen jiný název objektu, všechny vlastnosti a metody se odkazují k výchozímu objektu, dokud program nedosáhne příkaz End With.

Syntaxe:

With Objekt Blok příkazů End With

Parametry:

With a **End With** použijte, pokud máte pro jeden objekt několik vlastností nebo metod.

Další příkazy

Zde najdete příkazy, které nepatří do žádné jiné kategorie:

Příkaz Call [Runtime]

Přenese vykonávání programu do procedury, funkce nebo DLL procedury.

Funkce Choose [Runtime]

Vrátí vybranou hodnotu ze seznamu argumentů.

Příkaz Declare [Runtime]

Deklaruje a definuje podprogram v DLL souboru, který chcete spustit z OpenOffice.org Basic.

Příkaz End [Runtime]

Ukončí proceduru nebo blok.

Příkaz Exit [Runtime]

Ukončí **Do...Loop**, **For...Next**, funkci nebo proceduru.

Funkce FreeLibrary [Runtime]

Uvolní DLL, které bylo načteno příkazem Declare. Uvolněné DLL se automaticky znovu načte, pokud se zavolá některá funkce z něj. Viz také: Declare

Příkaz Function [Runtime]

Definuje podprogram, který lze zavolat z programu a který vrací hodnotu.

Příkaz Rem [Runtime]

Určuje, že řádek programu je komentář.

Příkaz Stop [Runtime]

Zastaví provádění programu Basic.

Příkaz Sub [Runtime]

Definuje podprogram.

Funkce Switch [Runtime]

Vyhodnotí seznam argumentů složený z výrazů a hodnot. Funkce Switch vrátí hodnotu, která je spojena s výrazem, který splní podmínku.

Příkaz With [Runtime]

Nastaví objekt jako výchozí objekt. Pokud není určen jiný název objektu, všechny vlastnosti a metody se odkazují k výchozímu objektu, dokud program nedosáhne příkaz End With.

Převádění čísel

Následující funkce převádí čísla z jednoho formátu do jiného.

Funkce Hex [Runtime]

Vrátí řetězec, který představuje šestnáctkovou hodnotu čísla.

Funkce Oct [Runtime]

Vrátí osmičkovou hodnotu čísla.

Funkce Hex [Runtime]

Vrátí řetězec, který představuje šestnáctkovou hodnotu čísla.

Syntaxe:

Hex (Číslo)

Návratová hodnota:

Řetězec

Parametry:

Číslo: Jakákoli číselná hodnota, kterou chcete převést na šestnáctkové číslo.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleHex
REM používá BasicFormula v OpenOffice.org Calc
Dim a2, b2, c2 as String
a2 = "&H3E8"
b2 = Hex2Int(a2)
MsgBox b2
c2 = Int2Hex(b2)
MsgBox c2
End Sub
Function Hex2Int( sHex As String ) As Long
REM Vrací celočíselnou hodnotu z šestnáctkové
Hex2Int = clng( sHex )
End Function
Function Int2Hex( iLong As Long) As String
REM Vypočte šestnáctkovou hodnotu celého čísla
Int2Hex = "&H" & Hex( iLong )
End Function
```

Funkce Switch [Runtime]

Vyhodnotí seznam argumentů složený z výrazů a hodnot. Funkce Switch vrátí hodnotu, která je spojena s výrazem, který splní podmínku.

Syntaxe:

Switch (Výraz1, Hodnota1[, Výraz2, Hodnota2[..., Výraz_n, Hodnota_n]])

Parametry:

Funkce **Switch** vyhodnocuje výrazy zleva doprava a poté vrátí hodnotu, jež odpovídá výrazu splňujícímu podmínku. Pokud nejsou zadány dvojice výraz-hodnota, dojde k chybě.

Výraz: Výraz, který chcete vyhodnotit.

Hodnota: Hodnota, která se vrátí, je-li výraz True.

V následujícím příkladu přiřadí funkce **Switch** odpovídající pohlaví podle jména předaného funkci:

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleSwitch
Dim sGender As String
sGender = GetGenderIndex( "John" )
MsgBox sGender
End Sub
Function GetGenderIndex (sName As String) As String
GetGenderIndex = Switch(sName = "Jane", "female", sName = "John", "male")
End Function
```

Příkaz Sub [Runtime]

Definuje podprogram.

Syntaxe

```
Sub Název[(Název_proměnné1 [As Typ][, Název_proměnné2 [As Typ][,...]])]
blok příkazů
End Sub
```

Parametry:

Název: Název podprogramu.
Název_proměnné: Parametr, který předáte proceduře.
Typ: Klíčové slovo deklarace typu.

Příklad:

```
Sub Example
REM nějaké příkazy
end sub
```

Funkce Oct [Runtime]

Vrátí osmičkovou hodnotu čísla.

Syntaxe:

Oct (Číslo)

Návratová hodnota:

Řetězec

Parametry:

Číslo: Číselný výraz, který chcete převést na osmičkovou hodnotu.

Chybové kódy

5 Neplatné volání procedury

Příklad:

Sub ExampleOkt
Msgbox Oct(255)
end sub

Řízení běhu programu

Následující příkazy řídí běh programu.

Program se obecně spouští od první řádky kódu po poslední. Také je možné podle určitých podmínek spouštět určité procedury nebo opakovat části programu v podprogramech. Pomocí smyček je možné opakovat části programu, kolikrát je třeba, nebo dokud se nesplní nějaká podmínka. Běh programu je možné ovlivňovat pomocí podmíněných příkazů, smyček nebo příkazů skoku.

Podmíněné příkazy

Následující příkazy jsou založeny na podmínkách.

Smyčky

Následující příkazy spouštějí smyčky.

Skoky

Následující příkazy provádějí skoky.

Další příkazy

Zde najdete příkazy, které nepatří do žádné jiné kategorie:

Příkaz Sub [Runtime]

Definuje podprogram.

Syntaxe

```
Sub Název[(Název_proměnné1 [As Typ][, Název_proměnné2 [As Typ][,...]])]
blok příkazů
End Sub
```

Parametry:

Název: Název podprogramu.
Název_proměnné: Parametr, který předáte proceduře.
Typ: Klíčové slovo deklarace typu.

Příklad:

```
Sub Example
REM nějaké příkazy
end sub
```

Příkaz Stop [Runtime]

Zastaví provádění programu Basic.

Syntaxe:

Stop

Příklad:

```
Sub ExampleStop
Dim iVar As Single
iVar = 36
Stop
Msgbox Sqr(iVar)
end sub
```

Podmíněné příkazy

Následující příkazy jsou založeny na podmínkách.

Příkaz If...Then...Else [Runtime]

Určuje jeden či více bloků příkazů, které chcete spustit pouze, je-li daná podmínka True.

Příkaz Select...Case [Runtime]

V závislosti na hodnotě výrazu definuje jeden nebo více bloků příkazů.

Příkaz IIf [Runtime]

Vrátí jeden ze dvou možných výsledků funkce, v závislosti na logické hodnotě vyhodnoceného výrazu.

Příkaz If...Then...Else [Runtime]

Určuje jeden či více bloků příkazů, které chcete spustit pouze, je-li daná podmínka True.

Syntaxe:

If podmínka=true Then Blok příkazů [ElseIf podmínka=true Then] Block příkazů [Else] Blok příkazů EndIf
Místo Else If můžete psát ElseIf, místo End If můžete psát EndIf.

Parametry:

Příkaz **If...Then** spustí blok příkazů v závislosti na dané podmínce. Když OpenOffice.org Basic narazí na příkaz **If**, ověří podmínku. Je-li podmínka True, spustí se všechny následující příkazy až do příštího příkazu **Else** nebo **ElseIf**. Je-li podmínka False a následuje příkaz **ElseIf**, OpenOffice.org Basic ověří podmínku a spustí následující příkazy, je-li podmínka True. Je-li False, program přeskočí příkazy až k dalšímu příkazu **ElseIf** nebo **Else**. Příkazy následující za příkazem **Else** se spustí pouze, nebyla-li žádná z předchozích podmínek splněna. Po vyhodnocení všech podmínek a spuštění odpovídajících příkazů pokračuje program příkazy následujícími po **EndIf**.

Možné je do sebe vložit více příkazů **If...Then**.

Příkazy **Else** a **ElseIf** jsou volitelné.

Pomocí příkazů **GoTo** a **GoSub** je možné vyskočit z bloku **If...Then**, ale není možné skočit do struktury **If...Then**.

Následující příkaz vám umožní zadat datum trvanlivosti výrobku a určí, zda již toto datum proběhlo:

Příklad:

```
Sub ExampleIfThenDate
Dim sDate as String
Dim sToday as String
sDate = InputBox("Zadejte trvanlivost (MM.DD.YYYY)")
sDate = Right$(sDate, 4) + Mid$(sDate, 4, 2) + Left$(sDate, 2)
sToday = Date$
sToday = Right$(sToday, 4)+ Mid$(sToday, 4, 2) + Left$(sToday, 2)
If sDate < sToday Then
MsgBox "Trvanlivost již vypršela"
ElseIf sDate > sToday Then
MsgBox "Trvanlivost ještě nevypršela"
Else
MsgBox "Datum trvanlivosti je dnes"
End If
End Sub
```

Příkaz Rem [Runtime]

Určuje, že řádek programu je komentář.

Syntaxe:

Rem Text

Parametry:

Text: Jakýkoliv text komentáře.

Pro označení komentáře je možné namísto klíčového slova Rem použít apostrof. Tento symbol napište přímo vpravo od kódu programu a za něj komentář.

Dlouhé řádky je možné rozdělit na několik kratších, když jako poslední dva znaky na řádku vložíte mezeru a podtržítko _. Aby to platilo i pro komentáře, musíte do stejného modulu Basic zadat "Option Compatible".

Příklad:

```
Sub ExampleMid
DIM sVar As String
sVar = "Las Vegas"
Print Mid(sVar,3,5) REM Returns "s Veg"
REM Tady se nic neděje
end sub
```

Příkaz Function [Runtime]

Definuje podprogram, který lze zavolat z programu a který vrací hodnotu.

Syntaxe

viz Parametr

Parametry:

Syntaxe
Function Název[(Název_proměnné 1 [As Typ][, Název_proměnné 2 [As Typ][,...]]]) [As Typ]
blok příkazů
[Exit Function]
blok příkazů
End Function
Parametr
Název: Název podprogramu.
Název_proměnné: Parametr, který bude předán podprogramu.
Typ: Klíčové slovo deklarace typu.

Příklad:

```
Sub ExampleExit
Dim sReturn As String
Dim sListArray(10) as String
Dim siStep as Single
For siStep = 0 to 10 REM Naplní pole testovacími údaji
sListArray(siStep) = chr$(siStep + 65)
msgbox sListArray(siStep)
next siStep
sReturn = LinSearch(sListArray(), "B")
Print sReturn
end sub
Function LinSearch( sList(), sItem As String ) as integer
dim iCount as Integer
REM Linsearch vyhledá v TextArray:sList() položku TextEntry:
REM Návratová hodnota je index záznamu nebo 0 (Null)
for iCount=1 to Ubound( sList() )
if sList( iCount ) = sItem then
exit for REM sItem found
end if
next iCount
if iCount = Ubound( sList() ) then iCount = 0
LinSearch = iCount
end function
```

Příkaz Select...Case [Runtime]

V závislosti na hodnotě výrazu definuje jeden nebo více bloků příkazů.

Syntaxe:

Select Case podmínka Case výraz Blok příkazů [Case podmínka2 Blok příkazů][Case Else] Blok příkazů End Select

Parametry:

Podmínka: Jakýkoliv výraz, podle kterého se určí spuštění bloků příkazů u klauzulí Case.
Výraz: Jakýkoliv výraz, který je kompatibilní s typem podmínky. Pokud **výraz** odpovídá **podmínce**, spustí se následující blok příkazů.

Příklad:

```
Sub ExampleRandomSelect
Dim iVar As Integer
iVar = Int((15 * Rnd) -2)
Select Case iVar
Case 1 To 5
Print "Čísla od 1 do 5"
Case 6, 7, 8
Print "Čísla od 6 do 8"
Case 8 To 10
Print "Větší než 8"
Case Else
Print "Mimo rozsah 1 až 10"
End Select
end sub
```

Příkaz Iif [Runtime]

Vrátí jeden ze dvou možných výsledků funkce, v závislosti na logické hodnotě vyhodnoceného výrazu.

Syntaxe:

Iif (Výraz, VýrazTrue, VýrazFalse)

Parametry:

Výraz: Výraz, který chcete vyhodnotit. Pokud se výraz vyhodnotí jako **True**, funkce vrátí výsledek VýrazTrue, jinak vrátí výsledek VýrazFalse.

VýrazTrue, VýrazFalse: Výraz, který se podle hodnoty logického výrazu vrátí jako výsledek funkce.

Chybové kódy

5 Neplatné volání procedury

Funkce FreeLibrary [Runtime]

Uvolní DLL, které bylo načteno příkazem Declare. Uvolněné DLL se automaticky znovu načte, pokud se zavolá některá funkce z něj. Viz také: Declare

Syntaxe:

FreeLibrary (Název knihovny As String)

Parametry:

LibName: Řetězec, který určuje název DLL.

FreeLibrary může uvolnit jen DLL načtené za běhu programu Basic.

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Declare Sub MyMessageBeep Lib "user32.dll" Alias "MessageBeep" ( long )
Sub ExampleDeclare
Dim lValue As Long
lValue = 5000
MyMessageBeep( lValue )
FreeLibrary("user32.dll" )
End Sub
```

Příkaz Exit [Runtime]

Ukončí **Do...Loop**, **For...Next**, funkci nebo proceduru.

Syntaxe:

viz Parametry

Parametry:

Exit Do

Je platné pouze v příkazu **Do...Loop** a ukončí smyčku. Běh programu pokračuje příkazem, který následuje za příkazem Loop. Pokud jsou příkazy **Do...Loop** vnořeny, předá se řízení smyčce na vyšší úrovni.

Exit For

Je platné pouze v příkazu **For...Next** a ukončí smyčku. Běh programu pokračuje příkazem, který následuje za příkazem Next. Pokud jsou příkazy smyčky vnořeny, předá se řízení smyčce na vyšší úrovni.

Exit Function

Okamžitě ukončí **Function**. Provádění programu pokračuje příkazem, který následuje po volání **Function**.

Exit Sub

Okamžitě ukončí proceduru. Provádění programu pokračuje příkazem, který následuje po volání **Sub**.

Příkaz Exit neurčuje konec struktury, nepleťte si jej s příkazem End.

Příklad:

```
Sub ExampleExit
Dim sReturn As String
Dim sListArray(10) as String
Dim siStep as Single
For siStep = 0 to 10 REM Naplní pole testovacími údaji
sListArray(siStep) = chr(siStep + 65)
msgbox sListArray(siStep)
next siStep
sReturn = LinSearch(sListArray(), "B")
Print sReturn
end sub
Function LinSearch( sList(), sItem As String ) as integer
dim iCount as Integer
REM LinSearch hledá v TextArray:sList() položku TextEntry:
REM Vrátí index záznamu nebo 0 (Null)
for iCount=1 to Ubound( sList() )
if sList( iCount ) = sItem then
Exit For REM nalezeno sItem
end if
next iCount
if iCount = Ubound( sList() ) then iCount = 0
LinSearch = iCount
end function
```

Smyčky

Následující příkazy spouštějí smyčky.

Příkaz Do...Loop [Runtime]

Příkazy mezi příkazy Do a Loop se budou opakovat, je-li podmínka splněna nebo do splnění podmínky.

Příkaz For...Next [Runtime]

Určíte, kolikrát se mají opakovat příkazy mezi For...Next.

Příkaz While...Wend [Runtime]

Když program narazí na příkaz While, ověří podmínku. Je-li podmínka False, program pokračuje přímo příkazem následujícím za příkazem Wend. Je-li podmínka True, smyčka se provádí, dokud program nenarazí na Wend a poté skočí zpět na **While**. Je-li podmínka stále True, smyčka se spustí znovu.

Příkaz Do...Loop [Runtime]

Příkazy mezi příkazy Do a Loop se budou opakovat, je-li podmínka splněna nebo do splnění podmínky.

Syntaxe

Do [{While | Until} Podmínka = True]
blok příkazů
[Exit Do]
blok příkazů
Loop
or
Do
blok příkazů
[Exit Do]
blok příkazů
Loop [{While | Until} Podmínka = True]

Parametry/elementy

Podmínka: Porovnávací, číselný nebo řetězcový výraz, který se vyhodnotí jako True nebo False.

Blok příkazů: Příkazy, které se opakují podle platnosti podmínky.

Příkaz **Do...Loop** se spouští ve smyčce podle platnosti podmínky. Podmínka pro ukončení smyčky se zadává buď po **Do** nebo **Loop**. Následující příkazy jsou platné kombinace:

Syntaxe

Do While Podmínka = True
...blok příkazů
Loop
Blok příkazů mezi příkazy Do While a Loop se opakuje tak dlouho, dokud je podmínka True.
Do Until Podmínka = True
...blok příkazů
Loop
Blok příkazů mezi příkazy Do Until a Loop se opakuje, dokud je podmínka False.
Do
...blok příkazů
Loop While Podmínka = True
Blok příkazů mezi Do a Loop se opakuje tak dlouho, dokud je podmínka True.
Do
...blok příkazů
Loop Until Podmínka = True
Blok příkazů mezi příkazy Do a Loop se opakuje, dokud pomídnka je True.

Bezpodmínečně smyčku ukončíte příkazem **Exit Do**. Tento příkaz je možné přidat kamkoliv mezi příkazy **Do**...**Loop**. Ukončovací podmínku je také možné určit pomocí struktury **If...Then** takto:

Do...
příkazy
If Podmínka = True Then Exit Do
příkazy
Loop...

Příklad

Sub ExampleDoLoop
Dim sFile As String
Dim sPath As String

Příkaz End [Runtime]

Ukončí proceduru nebo blok.

Syntaxe:

End, End Function, End If, End Select, End Sub

Parametry:

Příkaz End použijte následujícím způsobem:

Příkaz

End: Tento příkaz není povinný, ale lze jej zadat kdekoli v proceduře a ukončit tak průběh programu.
End Function: Ukončí příkaz **Function**.
End If: Označuje konec bloku **If...Then...Else**.
End Select: Označuje konec bloku **Select Case**.
End Sub: Ukončuje příkaz **Sub**.

Příklad:

Sub ExampleRandomSelect
Dim iVar As Integer
iVar = Int((15 * Rnd) -2)
Select Case iVar
Case 1 To 5
Print "Čísla od 1 do 5"
Case 6, 7, 8
Print "Čísla od 6 do 8"
Case Is > 8 And iVar < 11
Print "Větší než 8"
Case Else
Print "Mimo rozsah od 1 do 10"
End Select
end sub

Příkaz Declare [Runtime]

Deklaruje a definuje podprogram v DLL souboru, který chcete spustit z OpenOffice.org Basic.
Viz též: FreeLibrary

Syntaxe:

Declare {Sub | Function} Name Lib "Libname" [Alias "Aliasname"] [Parameter] [As Type]

Parametry:

Name: Název, pod kterým se podprogram volá z OpenOffice.org Basic, jiný než určený v DLL.

Aliasname: Název podprogramu určený v DLL.

Libname: Soubor nebo systémový název DLL. Tato knihovna se automaticky načte při prvním použití funkce.

Argumentlist: Seznam parametrů představující argumenty, které se při volání předávají proceduře. Typ a počet parametrů závisí na spouštěné proceduře.

Type: Definuje datový typ hodnoty, kterou funkce vrací. Tento parametr je možné vynechat, pokud za jménem funkce zadáte znak typové deklarace.

Chcete-li podprogramu předat parametr hodnotou namísto odkazem, musíte parametr určit klíčovým slovem **ByVal**.

Příklad:

```
Declare Sub MyMessageBeep Lib "user32.dll" Alias "MessageBeep" ( long )
Sub ExampleDeclare
Dim lValue As Long
lValue = 5000
MyMessageBeep( lValue )
FreeLibrary("user32.dll" )
End Sub
```

```
sPath = "c:\"
sFile = Dir$( sPath ,22)
If sFile <> "" Then
Do
MsgBox sFile
sFile = Dir$
Loop Until sFile = ""
End If
End Sub
```

Příkaz For...Next [Runtime]

Určíte, kolikrát se mají opakovat příkazy mezi For...Next.

Syntaxe:

For Počítadlo=Začátek To Konec [Step Krok]

blok příkazů

[Exit For]

blok příkazů

Next [Počítadlo]

Proměnné:

Počítadlo: Počítadlu smyčky se původně přiřadí hodnota vpravo od rovnítka (Začátek). Jsou platné jen číselné hodnoty. Počítadlo smyčky se zvyšuje nebo snižuje podle proměnné Krok, než dosáhne proměnné Konec.

Začátek: Číselná proměnná, která určuje počáteční hodnotu počítadla smyčky.

Konec: Číselná proměnná určující konečnou hodnotu počítadla smyčky.

Krok: Nastaví hodnotu, o kterou se zvyšuje nebo snižuje počítadlo smyčky. Pokud není Krok určen, zvyšuje se počítadlo o 1. V tom případě musí být Konec větší než Začátek. Pokud chcete počítadlo snižovat, musí být Konec nižší než Začátek a krok musí mít zápornou hodnotu.

Smyčka **For...Next** provádí všechny příkazy ve smyčce po určený počet opakování.

Při snižování nebo zvyšování počítadla kontroluje OpenOffice.org Basic, zda již přesáhlo koncovou hodnotu. Jakmile počítadlo dosáhne nebo překročí koncovou hodnotu, smyčka automaticky skončí.

Je možné vkládat do sebe příkazy **For...Next**. Pokud neurčíte proměnnou následující za **Next**, automaticky odkazuje na nejbližší poslední příkaz **For**.

Pokud určíte krok 0, příkazy mezi **For** a **Next** se opakují neustále.

Při snižování či zvyšování počítadla kontroluje OpenOffice.org Basic přetečení nebo podtečení. Smyčka skončí, když Počítadlo překročí Konec (kladný Krok) nebo je nižší než Konec (záporný Krok).

Pro bezpodmínečné ukončení smyčky použijte příkaz **Exit For**. Tento příkaz musí být ve smyčce **For...Next**. Pro ukončovací podmínku je možné použít příkaz **If...Then**:

For...

příkazy

If Podmínka = True Then Exit For

příkazy

Další

Poznámka: Ve vnořených podmínkách **For...Next** platí, že pokud použijete bezpodmínečné ukončení **Exit For**, ukončí se jen jedna smyčka.

Příklad

V následujícím příkladě jsou použity dvě vnořené smyčky k seřazení skupiny řetězců s 10 prvky (sEntry()), které jsou nejprve vyplněny různým obsahem:

```
Sub ExampleSort
Dim sEntry(9) As String
Dim iCount As Integer
Dim iCount2 As Integer
Dim sTemp As String
sEntry(0) = "Jerry"
sEntry(1) = "Patty"
sEntry(2) = "Kurt"
sEntry(3) = "Thomas"
sEntry(4) = "Michael"
sEntry(5) = "David"
sEntry(6) = "Cathy"
sEntry(7) = "Susie"
sEntry(8) = "Edward"
```

Funkce Choose [Runtime]

Vrátí vybranou hodnotu ze seznamu argumentů.

Syntaxe:

Choose (Index, Výběr1[, Výběr2, ... [,Výběr_n]])

Parametry:

Index: Číselný výraz, který určuje vrácenou hodnotu.

Výběr: Jakýkoliv výraz, který obsahuje jednu z možných voleb.

Funkce **Choose** vrátí hodnotu ze seznamu podle hodnoty indexu. Je-li Index = 1, funkce vrátí první hodnotu ze seznamu, je-li Index = 2, vrátí druhou hodnotu ze seznamu atd.

Je-li hodnota indexu menší než 1 nebo větší než počet výrazů v seznamu, vrátí funkce hodnotu Null.

Následují příkaz používá **Choose** k výběru řetězce z několika řetězců v nabídce:

Chybové kódy

5 Neplatné volání procedury

Příklad:

```
Sub ExampleChoose
Dim sReturn As String
sReturn = ChooseMenu(2)
Print sReturn
end sub
Function ChooseMenu(Index As Integer)
ChooseMenu = Choose(Index, "Quick Format", "Save Format", "System Format")
End Function
```

Příkaz Call [Runtime]

Přenese vykonávání programu do procedury, funkce nebo DLL procedury.

Syntaxe:

[Call] Název [Parametr]

Parametry:

Název: Název procedury, funkce nebo DLL, které chcete zavolat
Parametr: Parametry, které se předají proceduře. Typ a číslo parametrů závisí na spouštěném podprogramu.

Je-li volána procedura, klíčové slovo není povinné. Je-li spuštěna funkce, musí být parametry uzavřeny v závorkách. Je-li volána funkce knihovny DLL, musí být nejprve určena příkazem **Declare**.

Příklad:

```
Sub ExampleCall
Dim sVar As String
sVar = "Office"
Call f_callFun sVar
end Sub
Sub f_callFun (sText as String)
Msgbox sText
end sub
```

```
sEntry(9) = "Christine"
For iCount = 0 To 9
For iCount2 = iCount + 1 To 9
If sEntry(iCount) > sEntry(iCount2) Then
sTemp = sEntry(iCount)
sEntry(iCount) = sEntry(iCount2)
sEntry(iCount2) = sTemp
End If
Next iCount2
Next iCount
For iCount = 0 To 9
Print sEntry(iCount)
Next iCount
End Sub
```

Příkaz While...Wend [Runtime]

Když program narazí na příkaz While, ověří podmínku. Je-li podmínka False, program pokračuje přímo příkazem následujícím za příkazem Wend. Je-li podmínka True, smyčka se provádí, dokud program nenarazí na Wend a poté skočí zpět na **While**. Je-li podmínka stále True, smyčka se spustí znovu.

Na rozdíl od příkazu Do...Loop, není možné smyčku **While...Wend** přerušit pomocí Exit. Nikdy neukončujte smyčku While...Wend příkazem GoTo. Pokus o takovou akci způsobí chybu.

Smyčka Do...Loop je flexibilnější než While...Wend.

Syntaxe:

While Podmínka [Příkaz] Wend

Příklad:

```
Sub ExampleWhileWend
Dim stext As String
Dim iRun As Integer
sText ="This is a short text"
iRun = 1
while iRun < Len(sText)
if Mid(sText,iRun,1 )<> " " then Mid( sText ,iRun, 1, Chr( 1 + Asc( Mid(sText,iRun,1 )) )
iRun = iRun + 1
Wend
MsgBox sText,0,"Text encoded"
end sub
```

Další příkazy

Zde najdete příkazy, které nepatří do žádné jiné kategorie:

Příkaz Call [Runtime]

Přenese vykonávání programu do procedury, funkce nebo DLL procedury.

Funkce Choose [Runtime]

Vrátí vybranou hodnotu ze seznamu argumentů.

Příkaz Declare [Runtime]

Deklaruje a definuje podprogram v DLL souboru, který chcete spustit z OpenOffice.org Basic.

Příkaz End [Runtime]

Ukončí proceduru nebo blok.

Příkaz Exit [Runtime]

Ukončí **Do...Loop**, **For...Next**, funkci nebo proceduru.

Funkce FreeLibrary [Runtime]

Uvolní DLL, které bylo načteno příkazem Declare. Uvolněné DLL se automaticky znovu načte, pokud se zavolá některá funkce z něj. Viz také: Declare

Příkaz Function [Runtime]

Definuje podprogram, který lze zavolat z programu a který vrací hodnotu.

Příkaz Rem [Runtime]

Určuje, že řádek programu je komentář.

Příkaz Stop [Runtime]

Zastaví provádění programu Basic.

Příkaz Sub [Runtime]

Definuje podprogram.

Funkce Switch [Runtime]

Vyhodnotí seznam argumentů složený z výrazů a hodnot. Funkce Switch vrátí hodnotu, která je spojena s výrazem, který splní podmínku.

Příkaz With [Runtime]

Nastaví objekt jako výchozí objekt. Pokud není určen jiný název objektu, všechny vlastnosti a metody se odkazují k výchozímu objektu, dokud program nedosáhne příkaz End With.

Příkaz On...GoSub; příkaz On...GoTo [Runtime]

Přejde na jeden ze zadaných řádků v programovém kódu, v závislosti na hodnotě číselného výrazu.

Syntaxe:

On N GoSub Návěstí 1[, Návěstí 2[, Návěstí 3[,...]]]
On Číselný_výraz GoTo Návěstí1[, Návěstí2[, Návěstí3[,...]]]

Parametry:

Číselný_výraz: Jakýkoliv číselný výraz mezi 0 a 255, který určuje, kde bude program pokračovat. Je-li číselný výraz 0, příkaz se nespustí. Je-li číselný výraz větší než 0, program skočí na návěstí, které je na daném pořadovém místě (1 = Návěstí1, 2 = Návěstí2...).
Návěstí: Cílový řádek podle struktury **GoTo** nebo **GoSub**.

Jsou platné konvence pro **GoTo** nebo **GoSub**
.

Příklad:

```
Sub ExampleOnGosub
Dim iVar As Integer
Dim sVar As String
iVar = 2
sVar =""
On iVar GoSub Sub1, Sub2
On iVar GoTo Line1, Line2
Exit Sub
Sub1:
sVar =sVar & " From Sub 1 to" : Return
Sub2:
sVar =sVar & " From Sub 2 to" : Return
Line1:
sVar =sVar & " Návěstí 1" : GoTo Ende
Line2:
sVar =sVar & " Label 2"
Ende:
MsgBox sVar,0,"On...Gosub"
End Sub
```

Skoky

Následující příkazy provádějí skoky.

Příkaz GoSub...Return [Runtime]

Zavolá podprogram, který je určen návěstím. Příkazy následující po názvu se spouští, dokud program nenarazí na příkaz Return. Poté program pokračuje příkazem následujícím po příkazu **GoSub** .

Příkaz GoTo [Runtime]

Pokračuje v provádění programu v proceduře nebo funkci na řádku, který je označen návěstím.

Příkaz On...GoSub; příkaz On...GoTo [Runtime]

Přejde na jeden ze zadaných řádků v programovém kódu, v závislosti na hodnotě číselného výrazu.

Příkaz GoSub...Return [Runtime]

Zavolá podprogram, který je určen návěstím. Příkazy následující po názvu se spouští, dokud program nenarazí na příkaz Return. Poté program pokračuje příkazem následujícím po příkazu **GoSub** .

Syntaxe:

viz Parametry

Parametry:

Sub/Function
blok příkazů
Návěstí
blok příkazů
GoSub Návěstí
Exit Sub/Function
Návěstí:
blok příkazů
Return
End Sub/Function

Příkaz **GoSub** zavolá místní podprogram určený návěstím. Návěstí musí končit dvojtečkou (":").

Pokud program narazí na příkaz Return, kterému nepředchází **GoSub**, OpenOffice.org Basic vrátí chybovou zprávu. Pomocí **Exit Sub** nebo **Exit Function** zajistíte, že program opustí Sub nebo Function před dosažením příkazu Return.

Následující příklad demonstruje použití **GoSub** a **Return**. Dvojitým spuštěním části programu se vypočte odmocnina dvou zadaných čísel.

Příklad:

```
Sub ExampleGoSub
dim iInputa as Single
dim iInputb as Single
dim iInputc as Single
iInputa = Int(InputBox$ "Zadejte první číslo: ","NumberInput"))
iInputb = Int(InputBox$ "Zadejte druhé číslo: ","NumberInput"))
iInputc=iInputa
GoSub SquareRoot
Print "Odmocnina z ";iInputa;" is";iInputc
iInputc=iInputb
GoSub SquareRoot
Print "Odmocnina z ";iInputb;" is";iInputc
Exit Sub
SquareRoot:
iInputc=sqr(iInputc)
Return
End Sub
```

Příkaz GoTo [Runtime]

Pokračuje v provádění programu v proceduře nebo funkci na řádku, který je označen návěstím.

Syntaxe:

viz Parametry

Parametry:

Sub/Function
blok příkazů
Návěstí1
Návěstí2:
blok příkazů
Exit Sub
Návěstí1:
blok příkazů
GoTo Návěstí2
End Sub/Function

Pomocí příkazu GoTo určíte, že OpenOffice.org Basic má pokračovat ve spouštění programu na dalším místě v proceduře. Místo musí být určeno návěstím. Návěstí určíte názvem, které končí dvojtečkou (":").

Příkazem GoTo nelze opustit proceduru nebo funkci.

Příklad:

viz Parametry