LibreOffice Basic Overview

Overview
- Development time: Coding 20% - Maintaining 80%
- Entities Naming
  - Variables, constants, subs, functions, modules and libraries must be identified.
  - Allowed chars: unaccented chars, numbers, underscore (_).
  - An identifier can't start with a number nor contain a space.
- Don't use any Basic keyword to name an entity!
- Easy to read names: CamelCase, Name_with_separators
- Explicit names: IsCell(), SaveSpreadsheet()
- Comments
  - (apostrophe) or REM. What follows is a comment.
  - Comments are as important as code! They apply to the current line only.

Code Indent
- Indented code is easier to read. Indent each block level with Space / Tabulation.
- Several Instructions On The Same Line
  - Use the ":=" character (semicolon) to separate instructions on the same line.
  - Dim MyVar As Integer : MyVar = 123

Continuing An Instruction On The Next Line
- Last two chars of the first line: _ (space + underscore).

Variables
- Variable: a memory place. A variable contents may be modified at run-time.
  - By default, variable declares are not mandatory but this is dangerous (types lead to double declares).
- Adding Option Explicit on top of a module forces variable declaration.

Declaring Variables
- Simple Variables
  - Dim MyVar As AType
  - Dim A As Byte, B As String
  - Dim Nav As Integer : Nav = 123
- Array Variables
  - Dim Mat(2) As Integer

Setting Non-Object Variables
- MyVar = SomeValue
- Basic often automatically typecasts when SomeValue is not of the same type as MyVar. Prefer explicit typecasts using the CXXX functions (RefCard #5).

Creating/Setting Object Variables
- See RefCard #9: Structured Types

Variables Visibility
- Declaring...
  - Dim MyVar As AType
  - Static MyVar As AType
  - Private MyVar As AType
  - Public MyVar As AType
  - Global MyVar As AType

Types
- Specifies the value set a variable can carry or a function return.
- Predefined Simple Types

<table>
<thead>
<tr>
<th>Type Name</th>
<th>Description</th>
<th>Initial to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Logical values True / False.</td>
<td>False</td>
</tr>
<tr>
<td>Byte</td>
<td>Integer numbers (8 bits), from 0 to 255.</td>
<td>0</td>
</tr>
<tr>
<td>Currency (Decimal)</td>
<td>Integer numbers (4 decimals).</td>
<td>0.0000</td>
</tr>
<tr>
<td>Variant subtype, returned by CDec(string)</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>28 digits (int. part + dec. part). From 1 x 10^-28 to 7,9 x 10^28</td>
<td>(max precision 28 decimals)</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Dates and hours. In fact, doubles.</td>
<td>0.0</td>
</tr>
<tr>
<td>Double</td>
<td>Floating numbers (64 bits).</td>
<td>0.0</td>
</tr>
<tr>
<td>Integer</td>
<td>Integer numbers (16 bits), from -32768 to +32767.</td>
<td>0</td>
</tr>
<tr>
<td>Long</td>
<td>32 bits int numbers, -2 147 483 648 to +2 147 483 647.</td>
<td>0</td>
</tr>
<tr>
<td>Object</td>
<td>Objects. Allow to manipulate LibreOffice API objects.</td>
<td>Null</td>
</tr>
<tr>
<td>Single</td>
<td>Floating numbers (32 bits).</td>
<td>0.0</td>
</tr>
<tr>
<td>String</td>
<td>Text (0 to 65 545 characters).</td>
<td>&quot;null&quot;</td>
</tr>
<tr>
<td>Variant</td>
<td>Any type, incl. object.</td>
<td>Empty</td>
</tr>
</tbody>
</table>

LibOBasic_02_Overview_Flat_A4_EN_v200.odt
Control Structures

Loops
Repeat a sequence of instructions.
- **Premature exit possible using Exit End If** Do according to situation.
- **For** ... **Next**
  - For each index value ...
  - For i = Start To End [Step Increment] 
    - Instructions
  - Next i
- **For Each** ... **Next**
  - For each item...
  - For Each Item In SomeObject Do something with item
  - Next
  - The number of items is unknown.
- Do While Loop
  - Do While Condition 
    - Instructions 
  - Loop or...
  - While Condition 
    - instructions
  - Wend
- Do Loop ... Until
  - Do Loop Until Condition 
    - Instructions
  - Loop Until Condition
  - The number of items is unknown.

A branch that allows to take action according to a given state/situation.

- **If (alone)**
- **If Then [Else]**
- **ElseIf** OtherCondition Then
- **If** Condition Then
  - **Else**
  - **ElseIf**
  - **Else**

Condition is evaluated first.
- **infinite loops (condition never met)**
- **Older syntax, for compatibility only. Doesn't support Exit**
- **Do not use!**
- **Case** SomeVariable
  - **Case** Else
  - **Case** Value1, Value2
  - **Case** Value : DoThat()
  - **Case** Value

Choose among several possibilities, according to SomeVariable actual value.

Error Management
In Basic, error management relies on:
- **On Error** ... to check for error presence.
- **On Error Goto** MyLabel
- **On Error Resume Next**
- **On Error Goto** 0

Error Information Functions
- **Err** The message that describes the error.
- **Err** The line number where the error occurred.
- **Error** The error code.
  - An error code of (zero) means "no error".
  - Use If Err Then...

Error interception using **On Error Goto** as long as it has not been canceled.
- **On Error** Goto MyLabel
- **On Error Resume Next**
- **On Error** Goto 0
- **On Error Local** – Locally Intercepting Errors

In a subprogram, you might prefer the **On Local Error** syntax. This doesn't require calling **On Error** Goto 0 to cancel error interception: canceling is automatically performed when leaving the Sub or Function.

Different Ways Of Running A Macro

**Method**
- **LibreOffice**
- **Document Type**
- **Current Document**
- **Using a toolbar button**
- **Using a menu**
- **Using a shortcut**
- **Through an event**

Main Types Compatibility Chart

**Target**
- **Source**
  - **Integer**
  - **Long**
  - **Single**
  - **Double**
  - **Currency**
  - **Date**
  - **String**
  - **Object**
  - **Boolean**

- **Variant**
  - **Compatibility**
    - **compatible**
    - **not compatible**
  - **possible loss**
  - **possible overflow**

Reading The Chart
- **You may assign a source variable contents of type**
- **Double to a target variable**

Credits
Author: Jean-François Nifenecker — jean-francois.nifenecker@laposte.net

License
This refcard is placed under the CreativeCommons BY-SA v4 (intl) license.
More information:
https://creativecommons.org/licenses/by-sa/4.0/