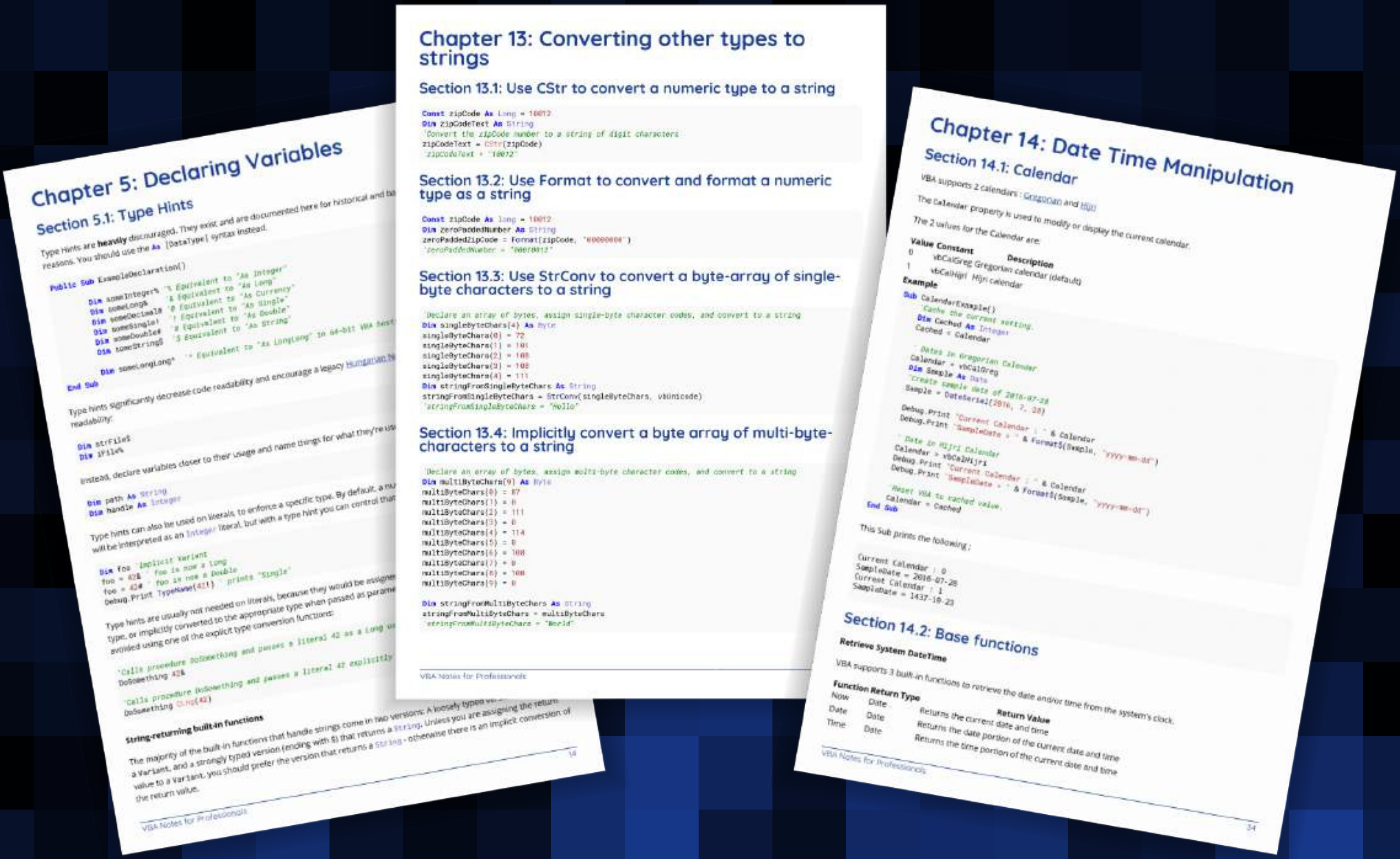
VBA

Notes for Professionals



Chapter 5: Declaring Variables

Section 5.1: Type Hints

Type Hints are **heavily** discouraged. They exist and are documented here for historical and ba reasons. You should use the `As [DataType]` syntax instead.

Type hints significantly decrease code readability and encourage a legacy Hungarian N

Instead, declare variables closer to their usage and name things for what they're us

Type hints can also be used on literals, to enforce a specific type. By default, a nu will be interpreted as an `Integer` literal, but with a type hint you can control that

Type hints are usually not needed on literals, because they would be assigne type, or implicitly converted to the appropriate type when passed as parame avoided using one of the explicit type conversion functions:

String-returning built-in functions

The majority of the built-in functions that handle strings come in two versions: A loosely typed ve a `Variant`, and a strongly typed version (ending with `$`) that returns a `String`. Unless you are assigning the return value to a `Variant`, you should prefer the version that returns a `String` - otherwise there is an implicit conversion of the return value.

Chapter 13: Converting other types to strings

Section 13.1: Use CStr to convert a numeric type to a string

```
Const zipCode As Long = 10012
Dim zipCodeText As String
'Convert the zipCode number to a string of digit characters
zipCodeText = CStr(zipCode)
'zipCodeText = "10012"
```

Section 13.2: Use Format to convert and format a numeric type as a string

```
Const zipCode As Long = 10012
Dim zeroPaddedNumber As String
zeroPaddedZipCode = Format(zipCode, "00000000")
'zeroPaddedNumber = "00010012"
```

Section 13.3: Use StrConv to convert a byte-array of single-byte characters to a string

```
'Declare an array of bytes, assign single-byte character codes, and convert to a string
Dim singleByteChars(4) As Byte
singleByteChars(0) = 72
singleByteChars(1) = 101
singleByteChars(2) = 108
singleByteChars(3) = 108
singleByteChars(4) = 111
Dim stringFromSingleByteChars As String
stringFromSingleByteChars = StrConv(singleByteChars, vbUnicode)
'stringFromSingleByteChars = "Hello"
```

Section 13.4: Implicitly convert a byte array of multi-byte-characters to a string

```
'Declare an array of bytes, assign multi-byte character codes, and convert to a string
Dim multiByteChars(9) As Byte
multiByteChars(0) = 87
multiByteChars(1) = 0
multiByteChars(2) = 111
multiByteChars(3) = 0
multiByteChars(4) = 114
multiByteChars(5) = 0
multiByteChars(6) = 108
multiByteChars(7) = 0
multiByteChars(8) = 100
multiByteChars(9) = 0

Dim stringFromMultiByteChars As String
stringFromMultiByteChars = multiByteChars
'stringFromMultiByteChars = "World"
```

VBA Notes for Professionals

Chapter 14: Date Time Manipulation

Section 14.1: Calendar

VBA supports 2 calendars : Gregorian and Hijri

The Calendar property is used to modify or display the current calendar.

The 2 values for the Calendar are:

Value	Constant	Description
0	vbCalGreg	Gregorian calendar (default)
1	vbCalHijri	Hijri calendar

Example

This Sub prints the following ;

```
Current Calendar : 0
SampleDate = 2016-07-28
Current Calendar : 1
SampleDate = 1437-10-23
```

Section 14.2: Base functions

Retrieve System DateTime

VBA supports 3 built-in functions to retrieve the date and/or time from the system's clock.

Function	Return Type	Return Value
Now	Date	Returns the current date and time
Date	Date	Returns the date portion of the current date and time
Time	Date	Returns the time portion of the current date and time

VBA Notes for Professionals

100+ pages

of professional hints and tricks

GoalKicker.com
Free Programming Books

Disclaimer
This is an unofficial free book created for educational purposes and is not affiliated with official VBA group(s) or company(s).
All trademarks and registered trademarks are the property of their respective owners

Contents

About ... 1

Chapter 1: Getting started with VBA ... 2

- Section 1.1: Accessing the Visual Basic Editor in Microsoft Office ... 2
- Section 1.2: Debugging ... 3
- Section 1.3: First Module and Hello World ... 4

Chapter 2: Comments ... 6

- Section 2.1: Apostrophe Comments ... 6
- Section 2.2: REM Comments ... 6

Chapter 3: String Literals - Escaping, non-printable characters and line-continuations ... 7

- Section 3.1: Escaping the " character ... 7
- Section 3.2: Assigning long string literals ... 7
- Section 3.3: Using VBA string constants ... 7

Chapter 4: VBA Option Keyword ... 9

- Section 4.1: Option Explicit ... 9
- Section 4.2: Option Base {0 | 1} ... 10
- Section 4.3: Option Compare {Binary | Text | Database} ... 12

Chapter 5: Declaring Variables ... 14

- Section 5.1: Type Hints ... 14
- Section 5.2: Variables ... 15
- Section 5.3: Constants (Const) ... 18
- Section 5.4: Declaring Fixed-Length Strings ... 19
- Section 5.5: When to use a Static variable ... 20
- Section 5.6: Implicit And Explicit Declaration ... 22
- Section 5.7: Access Modifiers ... 22

Chapter 6: Declaring and assigning strings ... 24

- Section 6.1: Assignment to and from a byte array ... 24
- Section 6.2: Declare a string constant ... 24
- Section 6.3: Declare a variable-width string variable ... 24
- Section 6.4: Declare and assign a fixed-width string ... 24
- Section 6.5: Declare and assign a string array ... 24
- Section 6.6: Assign specific characters within a string using Mid statement ... 25

Chapter 7: Concatenating strings ... 26

- Section 7.1: Concatenate an array of strings using the Join function ... 26
- Section 7.2: Concatenate strings using the & operator ... 26

Chapter 8: Frequently used string manipulation ... 27

- Section 8.1: String manipulation frequently used examples ... 27

Chapter 9: Substrings ... 29

- Section 9.1: Use Left or Left$ to get the 3 left-most characters in a string ... 29
- Section 9.2: Use Right or Right$ to get the 3 right-most characters in a string ... 29
- Section 9.3: Use Mid or Mid$ to get specific characters from within a string ... 29
- Section 9.4: Use Trim to get a copy of the string without any leading or trailing spaces ... 29

Chapter 10: Searching within strings for the presence of substrings ... 30

- Section 10.1: Use InStr to determine if a string contains a substring ... 30
- Section 10.2: Use InStrRev to find the position of the last instance of a substring ... 30
- Section 10.3: Use InStr to find the position of the first instance of a substring ... 30

Chapter 11: Assigning strings with repeated characters 31
Section 11.1: Use the String function to assign a string with n repeated characters 31
Section 11.2: Use the String and Space functions to assign an n-character string 31
Chapter 12: Measuring the length of strings 32
Section 12.1: Use the Len function to determine the number of characters in a string 32
Section 12.2: Use the LenB function to determine the number of bytes in a string 32
Section 12.3: Prefer `If Len(myString) = 0 Then` over `If myString = "" Then` 32
Chapter 13: Converting other types to strings 33
Section 13.1: Use CStr to convert a numeric type to a string 33
Section 13.2: Use Format to convert and format a numeric type as a string 33
Section 13.3: Use StrConv to convert a byte-array of single-byte characters to a string 33
Section 13.4: Implicitly convert a byte array of multi-byte-characters to a string 33
Chapter 14: Date Time Manipulation 34
Section 14.1: Calendar 34
Section 14.2: Base functions 34
Section 14.3: Extraction functions 36
Section 14.4: Calculation functions 37
Section 14.5: Conversion and Creation 39
Chapter 15: Data Types and Limits 41
Section 15.1: Variant 41
Section 15.2: Boolean 42
Section 15.3: String 42
Section 15.4: Byte 43
Section 15.5: Currency 44
Section 15.6: Decimal 44
Section 15.7: Integer 44
Section 15.8: Long 44
Section 15.9: Single 45
Section 15.10: Double 45
Section 15.11: Date 45
Section 15.12: LongLong 46
Section 15.13: LongPtr 46
Chapter 16: Naming Conventions 47
Section 16.1: Variable Names 47
Section 16.2: Procedure Names 50
Chapter 17: Data Structures 52
Section 17.1: Linked List 52
Section 17.2: Binary Tree 53
Chapter 18: Arrays 54
Section 18.1: Multidimensional Arrays 54
Section 18.2: Dynamic Arrays (Array Resizing and Dynamic Handling) 59
Section 18.3: Jagged Arrays (Arrays of Arrays) 60
Section 18.4: Declaring an Array in VBA 63
Section 18.5: Use of Split to create an array from a string 64
Section 18.6: Iterating elements of an array 65
Chapter 19: Copying, returning and passing arrays 67
Section 19.1: Passing Arrays to Proceedures 67
Section 19.2: Copying Arrays 67
Section 19.3: Returning Arrays from Functions 69

Chapter 20: Collections ... 71
Section 20.1: Getting the Item Count of a Collection ... 71
Section 20.2: Determining if a Key or Item Exists in a Collection ... 71
Section 20.3: Adding Items to a Collection ... 72
Section 20.4: Removing Items From a Collection ... 73
Section 20.5: Retrieving Items From a Collection ... 74
Section 20.6: Clearing All Items From a Collection ... 75
Chapter 21: Operators ... 77
Section 21.1: Concatenation Operators ... 77
Section 21.2: Comparison Operators ... 77
Section 21.3: Bitwise \ Logical Operators ... 79
Section 21.4: Mathematical Operators ... 81
Chapter 22: Sorting ... 82
Section 22.1: Algorithm Implementation - Quick Sort on a One-Dimensional Array ... 82
Section 22.2: Using the Excel Library to Sort a One-Dimensional Array ... 82
Chapter 23: Flow control structures ... 85
Section 23.1: For loop ... 85
Section 23.2: Select Case ... 86
Section 23.3: For Each loop ... 87
Section 23.4: Do loop ... 88
Section 23.5: While loop ... 88
Chapter 24: Passing Arguments ByRef or ByVal ... 89
Section 24.1: Passing Simple Variables ByRef And ByVal ... 89
Section 24.2: ByRef ... 90
Section 24.3: ByVal ... 91
Chapter 25: Scripting.FileSystemObject ... 93
Section 25.1: Retrieve only the path from a file path ... 93
Section 25.2: Retrieve just the extension from a file name ... 93
Section 25.3: Recursively enumerate folders and files ... 93
Section 25.4: Strip file extension from a file name ... 94
Section 25.5: Enumerate files in a directory using FileSystemObject ... 94
Section 25.6: Creating a FileSystemObject ... 95
Section 25.7: Reading a text file using a FileSystemObject ... 95
Section 25.8: Creating a text file with FileSystemObject ... 96
Section 25.9: Using FSO.BuildPath to build a Full Path from folder path and file name ... 96
Section 25.10: Writing to an existing file with FileSystemObject ... 97
Chapter 26: Working With Files and Directories Without Using FileSystemObject ... 98
Section 26.1: Determining If Folders and Files Exist ... 98
Section 26.2: Creating and Deleting File Folders ... 99
Chapter 27: Reading 2GB+ files in binary in VBA and File Hashes ... 100
Section 27.1: This have to be in a Class module, examples later referred as "Random" ... 100
Section 27.2: Code for Calculating File Hash in a Standard module ... 103
Section 27.3: Calculating all Files Hash from a root Folder ... 105
Chapter 28: Creating a procedure ... 109
Section 28.1: Introduction to procedures ... 109
Section 28.2: Function With Examples ... 109
Chapter 29: Procedure Calls ... 111
Section 29.1: This is confusing. Why not just always use parentheses? ... 111
Section 29.2: Implicit Call Syntax ... 111

Section 29.3: Optional Arguments ... 112
Section 29.4: Explicit Call Syntax ... 112
Section 29.5: Return Values ... 113
Chapter 30: Conditional Compilation ... 114
Section 30.1: Changing code behavior at compile time ... 114
Section 30.2: Using Declare Imports that work on all versions of Office ... 115
Chapter 31: Object-Oriented VBA ... 117
Section 31.1: Abstraction ... 117
Section 31.2: Encapsulation ... 117
Section 31.3: Polymorphism ... 121
Chapter 32: Creating a Custom Class ... 124
Section 32.1: Adding a Property to a Class ... 124
Section 32.2: Class module scope, instancing and re-use ... 125
Section 32.3: Adding Functionality to a Class ... 125
Chapter 33: Interfaces ... 127
Section 33.1: Multiple Interfaces in One Class - Flyable and Swimable ... 127
Section 33.2: Simple Interface - Flyable ... 128
Chapter 34: Recursion ... 130
Section 34.1: Factorials ... 130
Section 34.2: Folder Recursion ... 130
Chapter 35: Events ... 132
Section 35.1: Sources and Handlers ... 132
Section 35.2: Passing data back to the event source ... 134
Chapter 36: Scripting.Dictionary object ... 136
Section 36.1: Properties and Methods ... 136
Chapter 37: Working with ADO ... 138
Section 37.1: Making a connection to a data source ... 138
Section 37.2: Creating parameterized commands ... 138
Section 37.3: Retrieving records with a query ... 139
Section 37.4: Executing non-scalar functions ... 141
Chapter 38: Attributes ... 142
Section 38.1: VB_PredeclaredId ... 142
Section 38.2: VB_[Var]UserMemId ... 142
Section 38.3: VB_Exposed ... 143
Section 38.4: VB_Description ... 144
Section 38.5: VB_Name ... 144
Section 38.6: VB_GlobalNameSpace ... 144
Section 38.7: VB_Createable ... 145
Chapter 39: User Forms ... 146
Section 39.1: Best Practices ... 146
Section 39.2: Handling QueryClose ... 148
Chapter 40: CreateObject vs. GetObject ... 150
Section 40.1: Demonstrating GetObject and CreateObject ... 150
Chapter 41: Non-Latin Characters ... 151
Section 41.1: Non-Latin Text in VBA Code ... 151
Section 41.2: Non-Latin Identifiers and Language Coverage ... 152
Chapter 42: API Calls ... 153
Section 42.1: Mac APIs ... 153

Click here to download full PDF material