

**G22.3033-002**  
**Scripting Languages**  
 5/22/2008  
 Introduction  
 End User Programming (VBA)

© Martin Hirzel      G22.3033-002 NYU 5/22/2008      1

## Outline

- Introduction to Scripting Languages
- Course Mechanics
- End User Programming and VBA

© Martin Hirzel      G22.3033-002 NYU 5/22/2008      2

**Concepts**

## What is Scripting

- Practice
  - Rapid prototyping (Perl, Tcl, Python)
  - End-user programming (shells, macros, config files, spreadsheets)
- Domain
  - Text processing (awk, Perl)
  - Application extension (VBA, AppleScript)
  - Web (JavaScript, PHP, Ruby)

© Martin Hirzel      G22.3033-002 NYU 5/22/2008      3

**Concepts**

## Scripting Language

Language whose features favor scripting

Low boiler-plate	<code>print "Ice cream!\n"</code>
Dynamic typing	<code>\$amount = 20 . " grams";</code>
Interpretation	<code>eval "print 'egg\n'";</code>
String manipulation	<code>\$x = "food"; \$x =~ s/o/e/g;</code>
Associative arrays	<code>\$group{pasta} = 'carbs';</code>
Properties	<code>document.iml.src="meal.jpg";</code>
Call-backs	<code>&lt;input ... onClick="stir()"&gt;</code>

© Martin Hirzel      G22.3033-002 NYU 5/22/2008      4

**Concepts**

## Why Scripting

- Simplicity
- Productivity
- Domain

[Ousterhout'98] <http://www.tcl.tk/doc/scripting.html>

© Martin Hirzel      G22.3033-002 NYU 5/22/2008      5

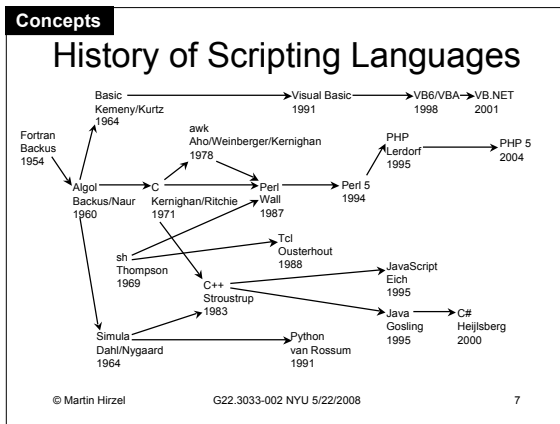
**Concepts**

## Language Popularity Index

Position May 2008	Position May 2007	Delta in Position	Programming Language	Rating May 2008	Delta May 2007	Status
1	1	▲	Java	20.77%	+1.0%	A
2	2	▲	C	15.29%	+0.14%	A
3	5	▲▲	(Visual) Basic	10.76%	+2.3%	A
4	4	▲	PHP	10.57%	+1.9%	A
5	3	▲▲	C++	10.48%	+0.27%	A
6	6	▲	Perl	5.86%	-0.2%	A
7	7	▲	Python	4.61%	+0.83%	A
8	8	▲	C#	3.96%	+0.31%	A
9	10	▲	Ruby	2.85%	+0.22%	A
10	11	▲	Delphi	2.64%	+0.51%	A
11	9	▲▲	JavaScript	2.41%	-0.6%	A
12	14	▲▲	D	1.31%	-0.03%	A
13	12	▲	SAS	0.73%	-1.34%	A
14	13	▲	PL/SQL	0.62%	-1.3%	B

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

© Martin Hirzel      G22.3033-002 NYU 5/22/2008      6



**Perl**

## Text Processing Example

```
#!/usr/bin/perl -w
%cup2g = ( flour => 110, sugar => 225, butter => 225 );
%volume = ( cup => 1, tbsp => 16, tsp => 48, ml => 236 );
%weight = ( lb => 1, oz => 16, g => 453 );
while (<>) {
    my ($qty, $unit, $ing) = /([0-9.]+) (\w+) (\w+)/;
    if ($cup2g{$ing} && $volume{$unit}) {
        $qty = 1.0 * $qty * $cup2g{$ing} / $volume{$unit};
        $unit = 'g';
    } elsif ($volume{$unit}) {
        $qty = 1.0 * $qty * $volume{ml} / $volume{$unit};
        $unit = 'ml';
    } elsif ($weight{$unit}) {
        $qty = 1.0 * $qty * $weight{g} / $weight{$unit};
        $unit = 'g';
    }
    printf("%d $unit $ing\n", $qty + .5);
}

```

© Martin Hirzel G22.3033-002 NYU 5/22/2008 8


**VBA**

## Application Extension Example

Option Explicit

```
Sub LemonStar ()
    Dim S As PowerPoint.Slide
    Set S = ActivePresentation.Slides( _
        ActivePresentation.Slides.Count)
    Dim I As Integer
    For I = 0 To 8
        Dim L As PowerPoint.Shape
        Const Dpi As Integer = 72 ' 72 dots per inch
        Set L = S.Shapes.AddLine( _
            BeginX:=Dpi*5, BeginY:=Dpi*3.75+I*Dpi/8, _
            EndX :=Dpi*6, EndY :=Dpi*4.75-I*Dpi/8)
        L.Line.ForeColor.RGB = RGB(I * 31, I * 31, 0)
    Next I
End Sub

```



© Martin Hirzel G22.3033-002 NYU 5/22/2008 9

**PHP**

## Server-Side Scripting Example

```
<?php
$d = sqlite_open("data/sqlite2", 0666, $err);
if ($err) { die($err); }
sqlite_query($d, "select * from T", SQLITE_BOTH, $err);
if ($err) {
    echo "table does not yet exist, creating it ..<br>";
    $q = "create table T(I integer, S char(10))";
    sqlite_query($d, $q, SQLITE_BOTH, $err);
    if ($err) { die($err); }
    sqlite_query($d, "insert into T values(0, 'n')");
}
$rows = sqlite_query($d, "select I from T where S='n'");
$row = sqlite_fetch_array($rows, SQLITE_BOTH);
echo "T[S=n][I]=" . $row['I'] . " "; reload for ++<br>";
sqlite_query($d, "update T set I = I+1 where S='n'");
echo "delete data/sqlite2 to start over<br>";
?>

```

© Martin Hirzel G22.3033-002 NYU 5/22/2008 10

**JavaScript**

## Client-Side Scripting Example

```
<html>
<head><title>Form validation example</title>
<script language="JavaScript">
function chk() {
    var v = document.myFm.num.value;
    if (v>=1 && v<=10) return true;
    alert("bad input " + v);
    return false; //abort commit
}
</script>
</head><body>
<form name="myFm" method="post" action="otherpage.htm">
Enter a number: <input size="4" type="text" name="num">
<input type="submit" value="OK" onClick="return chk()">
</form></body>
</html>

```

© Martin Hirzel G22.3033-002 NYU 5/22/2008 11

## Outline

- Introduction to Scripting Languages
- Course Mechanics
- End User Programming and VBA

© Martin Hirzel G22.3033-002 NYU 5/22/2008 12

## Administrative

## Course Goals

- Short-term
  - Have skills in Perl, VBA, PHP, JavaScript
  - Understand scripting PL concepts
- Long-term
  - Use languages effectively
  - Quickly learn new languages on your own
  - Design or improve scripting languages

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

13

## Administrative

## Course Schedule

Date	Lecture topic	Homework	Quiz
Th 5/22	Introduction, end user programming (VBA)	hw01	
Th 5/29	Dynamic typing, objects, properties	hw02	
Th 6/5	Textual data processing (Perl)	hw03	quiz1
Th 6/12	Dynamic scoping, context, objects	hw04	
Th 6/19	Server-side scripting (PHP)	hw05	
Th 6/26	Client-side scripting (JavaScript)	hw06	quiz2
Th 7/3	Web applications and databases	hw07	
Th 7/10	<a href="#">Marco Pistoia</a> : Security for web languages	hw08	
Th 7/17	Debugging	hw09	quiz3
Th 7/24	Other scripting languages	hw10	
Th 8/7	Final exam		

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

14

## Administrative

## Grading

- 25% hw + 35% quizzes + 40% final
- Homeworks
  - Due each Friday at 4pm
  - >= 1 minute late: 50% points
  - >= 3 hours late: 0% points
  - You can collaborate, but must indicate so

<http://www.cs.nyu.edu/courses/summer08/G22.3033-002/>

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

15

## Administrative

## Contact

- Instructor: Martin Hirzel  
hirzel@cs.nyu.edu
- Office hours: Thursday 5-6pm,  
Room CIWW 328, or by appointment
- Guest lecturer: Marco Pistoia  
[pistoia@us.ibm.com](mailto:pistoia@us.ibm.com)

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

16

## Reference

## Recommended Books

- Programming Language Pragmatics, 2nd edition. Michael Scott. Morgan Kaufmann, 2006.
- Programming Perl, 3rd edition [safari]. Larry Wall, Tom Christiansen, and Jon Orwant. O'Reilly, 2000.
- Mastering VBA, 2nd edition. Guy Hart-Davis. Wiley, 2005.
- Programming PHP, 2nd edition [safari]. Rasmus Lerdorf, Kevin Tatroe, and Peter MacIntyre. O'Reilly, 2006.
- JavaScript: The Definitive Guide, 5th edition [safari]. David Flanagan. O'Reilly, 2006.

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

17

## Outline

- Introduction to Scripting Languages
- Course Mechanics
- End User Programming and VBA

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

18

**VBA**

## About VBA

- Visual Basic for Applications
  - Visual = create GUI by drag'n'drop
  - Basic = simple, for end users
  - for Applications = embedded in Word, Excel, Powerpoint, AutoCAD, Corel Draw, Adobe
- Object-oriented, event-driven
- Aimed at end user programming
  - Automate task user does by hand otherwise
  - Same niche as Emacs Lisp, AppleScript

© Martin Hirzel G22.3033-002 NYU 5/22/2008 19

**VBA**

## Related Languages

- VB6
  - Same language as VBA, different libraries
  - “End of life”, not supported on new Microsoft platforms
- VBScript
  - Subset of VB 6.0, for server-side scripts
- VB.NET
  - Not backward compatible, based on CLR

© Martin Hirzel G22.3033-002 NYU 5/22/2008 20

**VBA**

## How to Write + Run Code

- Open Word, PowerPoint, or Excel
- Tools→Macros→Record
  - Recorded macros are useful code blue-print
- Tools→Macros→Visual Basic Editor
  - Auto-completion, help system, object browser
  - Edit-eval-print loop (“immediate window”)
  - Debugger
- Tools→Customize
  - Add hand-written code to toolbars or menus

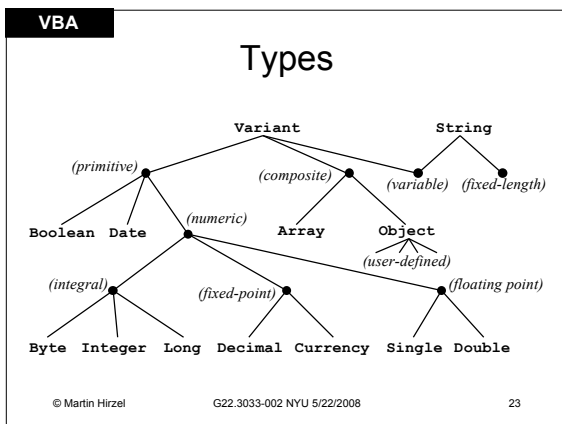
© Martin Hirzel G22.3033-002 NYU 5/22/2008 21

**VBA**

## Lexical Peculiarities

- Case insensitive
- Line break sensitive
  - One statement, multiple lines: `_`
  - One line, multiple statements: `:`
  - Single-line comments: `'`, `Rem`
- Literals
  - Boolean: `True`, `False`
  - Date: `#April 20, 2008#`, `#10:15pm#`
  - Variant: `Empty`, `Error`, `Nothing`, `Nul`
  - Character: `vbCr`, `vbTab`, `vbLf`, `Chr(149)`, ...

© Martin Hirzel G22.3033-002 NYU 5/22/2008 22



**VBA**

## Variable Declarations

Implicit, untyped	<code>fruit = "apple"</code>	Type declaration characters: % Integer, & Long, @ Currency, ! Single, # Double, \$ String
Implicit, typed	<code>fruit\$ = "apple"</code>	
Explicit, untyped	<code>Dim fruit</code>	Explicit declaration keywords: <code>Dim</code> , <code>Private</code> , <code>Public</code> , <code>Static</code>
Explicit, typed	<code>Dim f As String</code>	Enforce: <code>Option Explicit</code>
Constant	<code>Public Const pi_</code> <code>As Byte = 3</code>	
Array	<code>Dim a1()</code> <code>Dim a2(10)</code> <code>Dim a3(5 To 10)</code> <code>Dim a4(10, 5)</code>	Optional: <code>As type</code>

© Martin Hirzel G22.3033-002 NYU 5/22/2008 24

**Concepts**

## Gradual and Dynamic Typing

- *Gradual typing*: optional in declaration
- If not specified: **variant**
  - *Dynamic typing*: check at the last moment whether runtime value legal for operation
  - E.g., same variable can hold an **Integer** at one time, and a **String** at another
- If specified:
  - *Static typing*: check at compile time
  - Compiler can optimize time+space

© Martin Hirzel G22.3033-002 NYU 5/22/2008 25

**VBA**

## Arrays

- Indexing with round parentheses “ ( ) ”
- Statements
  - **ReDim [Preserve] id ( {new sizes/bounds} )**
  - “**Preserve**” keeps old values
  - **Erase id**
- Functions
  - **IsArray (id)**
  - **LBound (id [, dim] ) , UBound (id [, dim] )**
- Options
  - “**Option Base 1**” overrides default base-0 indexing

© Martin Hirzel G22.3033-002 NYU 5/22/2008 26

**Concepts**

## BNF = Backus Naur Form

Construct	Description	Example
Terminal	concrete code	<b>Courier Bold</b>
Non-terminal	placeholder	<i>Times Italic</i>
Ellipsis	omitted code	...
Rule	non-terminal definition	<i>Lhs ::= Rhs</i>
Alternative	choose one	<i>Alt1   Alt2</i>
Optional	zero or one times	[Square brackets]
Repeat	zero or more times	Kleene star*
Repeat	one or more times	Kleene plus+
Grouping	treat as unit	(parentheses)

© Martin Hirzel G22.3033-002 NYU 5/22/2008 27

**VBA**

## Input and Output

- **Debug.Print "Hello, world!"**
- **Application.StatusBar = "Hello, world!"**
- **MsgBox "Hello, world!"**
  - **MsgBox (prompt [, buttons] [, title] ) As Long**
  - Buttons: 0 vbOkOnly, 1 vbOkCancel, 2 vbAbortRetryIgnore, 3 vbYesNoCancel, 4 vbYesNo, 5 vbRetryCancel
  - Result: 1 vbOk, 2 vbCancel, 3 vbAbort, 4 vbRetry, 5 vbIgnore, 6 vbYes, 7 vbNo
- **userName = InputBox ("Who are you?")**
  - **InputBox (prompt [, title] [, default] [, xpos] [, ypos] ) As String**

© Martin Hirzel G22.3033-002 NYU 5/22/2008 28

**VBA**

## Operators

Operator	Arity	Associativity	Description
(...)			Subexpression; Call; Indexing
^	2	L	Exponentiation
+, -	1		Identity, negation
*, /	2	L	Multiplicative
\	2	L	Integer division
Mod	2	L	Modulus
+, -	2	L	Additive; String concatenation
&	2	L	String concatenation
<<, >>	2	L	Bit shift
=, <>, <, <=, >, >=, Is	2	L	Comparison
Not	1		Negation
And, Or, Xor, Eqv, Imp	2	L	Logic (not all same precedence)
Set ... = ...	2		Assignment statement

© Martin Hirzel G22.3033-002 NYU 5/22/2008 29

**Concepts**

## Operator Characterization

Operator	Arity	Associativity	Description
(...)			Subexpression; Call; Indexing
^	1 = unary	L = left	Exponentiation
+, -	2 = binary	R = right	Identity, negation
*, /			Multiplicative
\			Integer division
Mod			Modulus
+, -			Additive
&			String concatenation
<<, >>			Bit shift
=, <>, <, <=, >, >=, Is			Comparison
Not			Negation
And, Or, Xor, Eqv, Imp			Logic (not all same precedence)
Set ... = ...			Assignment statement

Precedence: from high to low

© Martin Hirzel G22.3033-002 NYU 5/22/2008 30

**Concepts**

### Arity, Precedence, Associativity

Arity	Number of operands	-2 2 - 2	unary binary
Precedence	Binding strength	2+2*2 (2+2)*2 2+(2*2)	* has higher precedence than +
Associativity	Grouping direction	2/2/2 (2/2)/2 2/(2/2)	/ is left-associative

Precedence and associativity in programming usually follows the conventions from math.

© Martin Hirzel G22.3033-002 NYU 5/22/2008 31

**VBA**

### Control Statements

Conditional	If ... Then ... ElseIf ... Else ... End If Select Case expr ... Case Else ... End Select
Fixed-iteration loops	For id = expr To expr Step expr ... Next id For Each id In expr ... Next id
Indefinite loops	Do ... Loop While expr ... Wend Do While expr ... Loop Do ... Loop While expr Do Until expr ... Loop Do ... Loop Until expr
Unstructured control	GoTo id / line number Exit Do Exit For Exit Function On Error GoTo id

© Martin Hirzel G22.3033-002 NYU 5/22/2008 32

**VBA**

### Writing Subroutines

- Declaration
  - `mods* Sub id [(arg*)] ... End Sub`
  - `mods* Function id [(arg*)] [As type] ... End Function`
  - To return a value, assign to function name
- Function modifiers
  - **Public, Private**: visibility outside module
  - **Static**: make all locals static
- Arguments: `mods* id [( ) ] [As type] [= expr]`
- Argument modifiers
  - **Optional**: after first, rest must also be **Optional**
  - **ByRef, ByVal**: default **ByRef**
  - **ParamArray**: must be last, allows for var-args

© Martin Hirzel G22.3033-002 NYU 5/22/2008 33

**Concepts**

### Positional vs. Named Parameters

```
Sub bake(Optional Amount=1, Optional Dish="pizza") ... End Sub
```

<code>bake( 2, "cakes")</code>	<i>positional</i>
<code>bake( 2 )</code>	
<code>bake( , "cakes")</code>	
<code>bake(Amount:=2, Dish:="cakes")</code>	<i>named</i>
<code>bake(Amount:=2 )</code>	
<code>bake( Dish:="cakes")</code>	
<code>bake(Dish:="cakes", Amount:=2)</code>	

Call syntax:

- How to omit optional parameters
- Whether or not order matters
- Note: parentheses not required

© Martin Hirzel G22.3033-002 NYU 5/22/2008 34

**VBA**

### Library Functions

- Simple data conversion: CBool, CByte, CCur, CDate, CDbt, CInt, CLng, CSng, CStr, CVar
- Complex data conversion: Asc, Chr, Format, Hex, Oct, RGB, QBColor, Str, Val
- String: InStr, InStrRev, LCase, UCase, Left, Len, LTrim, RTrim, Mid, Right, Space, StrComp, StrConv, StrReverse, Trim
- Math: Abs, Atn, Cos, Exp, Fix, Log, Rnd, Sgn, Sin, Sqr, Tan, IsNumeric
- And many more


© Martin Hirzel G22.3033-002 NYU 5/22/2008 35

**VBA**

### Example, Revisited

```
Option Explicit

Sub LemonStar()
    Dim S As PowerPoint.Slide
    Set S = ActivePresentation.Slides( _
        ActivePresentation.Slides.Count)
    Dim I As Integer
    For I = 0 To 8
        Dim L As PowerPoint.Shape
        Const Dpi As Integer = 72 ' 72 dots per inch
        Set L = S.Shapes.AddLine( _
            BeginX:=Dpi*5, BeginY:=Dpi*3.75+I*Dpi/8, _
            EndX :=Dpi*6, EndY :=Dpi*4.75-I*Dpi/8)
        L.Line.ForeColor.RGB = RGB(I * 31, I * 31, 0)
    Next I
End Sub
```



© Martin Hirzel G22.3033-002 NYU 5/22/2008 36

**Reference**

## VBA Documentation

- Included:
  - Macro recorder
  - Auto-completion
  - Help system
  - Object browser
- Book: Mastering VBA, 2nd edition.  
Guy Hart-Davis. Wiley, 2005.
- Online:  
MSDN Library→Development Tools+Languages  
→Visual Studio 6.0→VB 6.0

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

37

**Soap-box**

## Evaluating VBA

Strengths

- Development environment
- Simplicity
- Availability
- Popularity
- Best tool for end users of MS Office

Weaknesses

- Single platform/vendor
- Binary format
- Security
- Missing features
  - Structured exceptions
  - Implementation inheritance
  - Regular expressions
  - Associative arrays

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

38

**Administrative**

## Last Slide

- Please register
- Please subscribe to mailing list
- First homework is due on Friday 30 May
- Today's lecture
  - Introduction to scripting
  - Dynamic typing
  - Precedence and associativity
  - Basics of VBA
- Next lecture
  - Putting the V and the A in VBA
  - Application extension
  - Properties
  - Call-backs

© Martin Hirzel

G22.3033-002 NYU 5/22/2008

39