

1  **Scripting Languages**

Prof. Dr. Debora Weber-Wulff

2  **Table of Contents**

- Difference Compiler / Interpreter
- Popular scripting languages

3  **Compilers**

4  **Compilers**

5  **Interpreter**

6  **Java**

7  **Compilers**


8  **Scripting programming languages**

- Also called scripting languages or script languages
- Are computer programming languages designed for "scripting" the operation of a computer.
- Early script languages were often called *batch languages* or *job control languages (JCL)*.

9 

10  **Script**

- A script is a computer program that automates the sort of task that a user might otherwise do interactively at the keyboard.

11  **Shell Script**

- A shell script consists largely of the sort of commands that might be typed at a command prompt, or a sequence of tasks that the user expects to perform repeatedly.
- One can write quite elaborate programs in many languages that are still called scripts, even if they do far more than just automate things.

12 

bash shell script that lists all HTML files in a directory and

writes the 1st line of each to a file called File_Heads:

13  **Why Use Scripts?**

- rapid development vs. efficiency of execution
- strong at communication with program components written in other languages
- component "glue"

14  **Advantages of Scripting**

- Often very high level commands
- Automatic memory management
- Automatic bounds checking
- Faster to get something working
- Often platform independence

15  **Disadvantages of Scripting**

- Hard to optimize
- Longer running times
- Uses more memory
- Type checking only at run-time
- Easy to make big errors
- Hard to maintain

16  **Types of Scripting Languages**

- Application-specific
- Text-processing
- JCL / shell
- General purpose
- Extension/embeddable languages
- non-categorizable

17  **Application specific**

- Tailored to the needs of the application user
- Computer games often have their own script languages for

describing actions

18  **Typical languages**

- Action Code Script (ACS) HeXen, Doom
- Autolt (<http://www.autoitscript.com/autoit3/>) for Windows
- ActionScript for Flash
- BlobbieScript (<http://www.wocmud.org/Carnage/blobbieScript/>)
MUD scripting language
- GameMonkeyScript game and tool applications (<http://www.somedude.net/gamemonkey/>)

19  **Typical languages**

- IRC script
- Lingo for Macromedia Director
- QuakeC for customizing Quake
- UnrealScript for authoring game code
- ZZT-oop early gaming script language

20  **JavaScript**

- Not Java
- Not HTML
- Interpreted in a browser at runtime in order to offer advanced computing for web pages
- Netscape introduced it in 1995
- Microsoft then offered JScript

21  **JavaScript**

- And then back and forth, each gets new things (like the Document Object Model - DOM)
- And then this works and that and NOTHING is compatible with anything else...

22  **ActionScript**

- An ECMAScript-based programming language used for controlling Macromedia Flash movies and applications.

- Similar to JavaScript, but while JavaScript deals with windows, documents and forms, ActionScript deals with movie-clips, text fields and sounds.

23  **ActionScript**

- Flash 5 = ActionScript 1.0
- Flash 6 = enhanced possibilities
- Flash 7 (MX2004) = ActionScript 2.0 with strong typing and OO concepts
- Flash 8 = additional class libraries (Bitmap data, file upload)

24  **ActionScript**

- Flash 9 = ActionScript 3.0 (new virtual machine)
- Flash 10 = Basic 3D manipulation and a 3D drawing API; Uses a GPU; New sound API; P2P

25  **ActionScript**

- Everything is designed to be asynchronous; callbacks are ubiquitous, but Event objects do not exist.
- The XML implementation has been solid since Flash 5. Flash can send and receive XML asynchronously.

26  **ActionScript**

- Can be edited and saved as a frame in the movie
- Can also be in an external file, 2.0 also has movie scripts attached to a file
- Shared Objects! Remote Objects!
- Notoriously SLOOOOOOOOW

27  **ActionScript**

- on (press)

```

{ startDrag (this, true); }
on (release)
{ stopDrag ();
  if (this._droptarget == "/green_box")
    { _root.green_box.gotoAndStop(2); }
}

```

28  **Lingo**

- Lingo is a scripting language developed by John H. Thompson for use in Macromedia Director.
- HyperTalk is said to have been one of the inspirations for Lingo.

29  **Lingo**

- When Lingo was created, the syntax was designed to mimic spoken language, so it would be easy for beginners to get started.
- if the visible of sprite 5 then go to the frame

30  **Lingo**

- The latest version of the language fully supports dot syntax, so that the code looks more like in standard programming languages.
- if sprite(5).visible
then _movie.go(_movie.frame)

31  **Lingo**

- Lingo is embedded into Macromedia Director.
- You don't need to care much about hardware or details of getting things on the screen - the program does this for you.

32  **Lingo**

- The basic object is a sprite
- There are 3 types of scripts in Lingo:
 - BEHAVIOR scripts tell sprites what to do
 - MOVIE scripts control the scene
 - PARENT scripts create objects and sprites

33  **Lingo Example**

```

on beginSprite (me)
  -- create a reference to the stage's image
  myOutput = (the stage).image
  -- create a 'buffer' image.
  myBuffer = myOutput.duplicate()
  -- Create a little 'helper' pixel.
  myPixel = image(1,1,myBuffer.depth)
  myPixel.setPixel(0,0,rgb(0,0,0))
  -- Create an empty list to store 'brush' objects in
  myBrushList = []
  -- Start off by moving the pixels at
  -- a moderate speed.
  mySpeed =5.0
end

```

34  **Text processing languages**

- One of the oldest uses of scripting languages
- Unix's awk and, later, Perl, were originally designed to aid system administrators in automating tasks that involved Unix text-based configuration and log files.

35  **Text processing languages**

- SED
line by line processing of files

36  **Text processing languages**

- XSLT
Transforms XML documents into other documents

37  **awk**

- AWK is a general purpose computer language designed for processing text based data, either in files or data streams.
- The name AWK is derived from the surnames of its authors — Alfred V. Aho, Peter J. Weinberger, and Brian W. Kernighan.

38  **awk**

- A typical awk program consists of a series of lines, each of the form

```
/pattern/ { action }
```

- *pattern* is a regular expression and *action* is a command.

39 **awk**

- Awk looks through the input file; when it finds a line that matches *pattern*, it executes the command(s) specified in *action*.

40 **awk**

- Sum 1st column of input

```
{ s += $1 }  
END { print s }
```

41 **awk**

- Word Frequency
(uses associative arrays)

```
{ for (i=1; i<NF; i++) words[$i]++ } END { for (i in words) print i,  
words[i] }
```

42 **Text processing languages**

- Perl -- originally a report-generation language (hence its name, *Practical Extraction and Reporting Language*) it has grown into a full-fledged applications language in its own right.

<http://www.perl.org/>

43 **Perl**

- An interpreted procedural programming language designed by Larry Wall.
- Perl has a unique set of features partly borrowed from C, shell scripting (sh), awk, sed, and many other programming languages.

44 **perlintro(1) man page**

- Perl is a general-purpose programming language originally developed for text manipulation and now used for a wide range of tasks including system administration, web development, network programming, GUI development, and more.

45 **perlintro(1) man page**

- The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal). Its major features are that it's easy to use, supports both procedural and object-oriented (OO) programming, has powerful built-in support for text processing, and has one of the world's most impressive collections of third-party modules.

46 **Perl example**

- Hello world

```
#!/usr/bin/perl -w
print "Hello, world!\n";
```

47 **Perl example**

- rot-13 encoding/decoding

```
perl -pe 'tr/A-Za-z/N-ZA-Mn-za-m/' < input_file > output_file
```

- It is entered and run directly on the command line

48 **Fun with Perl**

- CPAN: Comprehensive Perl Archive Network
- Obfuscated code competitions
- Perl Poetry
- Lingua::Romana::Perligata for using Latin keywords

49 **Job control languages and shells**

- Starting and controlling the behavior of system programs is called "job control".
- Many of these languages' interpreters double as command-line interfaces, such as the Unix shell or the MS-DOS COMMAND.COM

50  **Job control languages**

- AppleScript
- AREXX (Amiga REXX)
- BS2000
- JCL
- MS-DOS batch
- REXX

51  **Shells**

- sh (Bourne shell)
- bash (*Bourne-again shell*)
- csh (C shell)
- ksh (Korn shell)

52  **General-purpose
dynamic languages**

- Some languages have begun as scripting languages but developed into programming languages suitable for broader purposes.
- Other similar languages -- frequently interpreted, memory-managed, dynamic -- have been described as "scripting languages" for these similarities, even if they are more commonly used for applications programming.

53  **General-Purpose
Dynamic Languages**

- ColdFusion
- Dylan
- Jython
- Perl
- PHP

54  **General-Purpose
Dynamic Languages**

- Pike
- Python
- Ruby

- Scheme

55 **Extension/embeddable languages**

- A small number of languages have been designed for the purpose of replacing application-specific scripting languages, by being embeddable in application programs.
- The application programmer (working in C or another systems language) includes "hooks" where the scripting language can control the application.

56 **Extension/embeddable languages**

- Guile
- Lua
- Tcl (Tool command language)
- JavaScript

57 **Others**

- There are many other scripting languages, not many are widely used.