UNIT-1

INTRODUCTION TO SCRIPTING LANGUAGES

1.1 Scripts and programs:

Scripting is the action of writing scripts using a scripting language, distinguishing neatly between programs, which are written in conventional programming language such as C,C++,java, and scripts, which are written using a different kind of language.

We could reasonably argue that the use of scripting languages is just another kind of programming. Scripting languages are used for is qualitatively different from conventional programming languages like C++ and Ada address the problem of developing large applications from the ground up, employing a team of professional programmers, starting from well-defined specifications, and meeting specified performance constraints.

Scripting languages, on other hand, address different problems:

- Building applications from 'off the shelf' components
- Controlling applications that have a programmable interface
- Writing programs where speed of development is more important than run-time efficiency.

The most important difference is that scripting languages incorporate features that enhance the productivity of the user in one way or another, making them accessible to people who would not normally describe themselves as programmers, their primary employment being in some other capacity. Scripting languages make programmers of us all, to some extent.

1.2 Origin of scripting:

The use of the word 'script' in a computing context dates back to the early 1970s,when the originators of the UNIX operating system create the term 'shell script' for sequence of commands that were to be read from a file and follow in sequence as if they had been typed in at the keyword. e.g. an ‘AWKscript’, a ‘perl script’ etc.. the name ‘script ‘ being used for a text file that was intended to be executed directly rather than being compiled to a different form of file prior to execution.

Other early occurrences of the term ‘script’ can be found. For example, in a DOS-based system, use of a dial-up connection to a remote system required a communication package that used proprietary language to write scripts to automate the sequence of operations required to establish a connection to a remote system.

Note that if we regard a scripts as a sequence of commands to control an application or a device, a configuration file such as a UNIX ‘make file’ could be regard as a script.

However, scripts only become interesting when they have the added value that comes from using programming concepts such as loops and branches.

1.3 Scripting today:

SCRIPTING IS USED WITH 3 DIFFRENT MEANINGS:

1. A new style of programming which allows applications to be developed much faster than traditional methods allow,and maks it possible for applications to evolve rapidly to meet changing user requirements.This style of programming frequently uses a scripting language to interconnect ‘off the shelf ‘ components that are themselves written in conventional language.Applications built in this way are called ‘glue applications’ ,and the language is called a ‘glue language’.

A **glue language** is a programming language (usually an interpreted scripting language) that is designed or suited for writing glue code – code to connect software components. They are especially useful for writing and maintaining:

- Custom commands for a command shell
- Smaller programs than those that are better implemented in a compiled language

- "Wrapper" programs for executables, like a batch file that moves or manipulates files and does other things with the operating system before or after running an application like a word processor, spreadsheet, data base, assembler, compiler, etc.
- Scripts that may change
- Rapid prototypes of a solution eventually implemented in another, usually compiled, language.

Glue language examples:

- AppleScript
- ColdFusion
- DCL
- Embeddable Common Lisp
- ecl
- Erlang
- JCL
- JScript and JavaScript
- Lua
- m4
- Perl
- PHP
- Pure
- Python
- Rebol
- Rexx
- Ruby
- Scheme
- Tcl
- Unix Shell scripts (ksh, csh,bash, sh and others)
- VBScript
- Work Flow Language
- Windows PowerShell
- XSLT

2.Using a scripting language to ‘manipulate,customize and automate the facilities of an existing system’,as the ECMAScript definition puts it.Here the script is used to control an application that privides a programmable interface:this may be an API,though more commonly the application is construted from a collection of objects whose properties and methods are exposed to the scripting language.Example: use of Visual Basic for applications to control the applications in the Microsoft Office Suite.

3.Using a scripting language with its rich funcationaliy and ease of use as an alternate to a conventional language for general programming tasks ,particularly system programming and administration.Examples: are UNIX system adminstrators have for a long time used scripting languages for system maintenace tasks,and administrators of WINDOWS NT systems are adopting a scripting language ,PERL for their work.

1.4 Characteristics of scripting languages:

These are some properties of scripting languages which differentiate SL from programming languages.

- Integrated compile and run:SL's are usually characterized as interpreted languages,but this is just an oversimplification.They operate on an immediate execution,without need to issue separate commond to compile the program and then to run the resulting object file,and without the need to link extensive libraries into he object code.This is done automatically.A few SL'S are indeed implemented as strict interpreters.
- Low overheads and ease of use:

  1.variables can be declared by use

  2.the number of different data types is usually limited
  3.everything is string by context it will be converted as number(vice versa)
  4.number of data strucures is limited(arrays)

- Enhanced functionality:SL's usually have enhanced functionality in some areas.For example ,most languages provide string manipulation based on the use of regular expressions,while other languages provide easy access to low-level operating system facilities,or to the API,or object exported by an application.
- Efficiency is not an issue:ease of use is achieved at the expense of effeciency,because efficiency is not an issue in the applications for which SL'S are designed.

- A scripting language is usually interpreted from source code or bytecode. By contrast, the software environment the scripts are written for is typically written in a compiled language and distributed in machine code form.

- Scripting languages may be designed for use by end users of a program – end-user development – or may be only for internal use by developers, so they can write portions of the program in the scripting language.

- Scripting languages typically use abstraction, a form of information hiding, to spare users the details of internal variable types, data storage, and memory management.
- Scripts are often created or modified by the person executing them, but they are also often distributed, such as when large portions of games are written in a scripting language.
- The characteristics of ease of use,particularly the lack of an explicit compile-link-load sequence,are sometimes taken as the sole definition of a scripting language.

1.5 Users For Scripting Lanuages:

Users are classified into two types

1. Modern applications
2. Traditional users

Modern applications of scripting languages are:
1.**Visual scripting**: A collection of visual objects is used to construct a graphical interface.This process of constructing a graphical interface is known as visual scripting.the properties of visual objects include text on button,background and foreground colors.These properties of objects can be changed by writing program in a suitable language.
The outstanding visual scripting system is visual basic.It is used to develop new applications.Visual scripting is also used to create enhanced web pages.

2.**Scripting components**:In scripting languages we use the idea to control the scriptable objects belonging to scripting architecture.Microsoft's visual basic and excel are the first applications that used the concept of scriptable objects.To support all the applications of microsoft the concept of scriptable objects was developed.3.Web scripting:web scripting is classified into three forms.they are processing forms,dynamic web pages,dynamically generating HTML.

Applications of traditional scripting languages are:

1. system administration,
2. experimental programming,
3. controlling applications.

Application areas :
Four main usage areas for scripting languages:
1. Command scripting languages
2.Application scripting languages
3.Markup language
4. Universal scripting languages

1.**Command scripting languages** are the oldest class of scripting languages. They appeared in 1960, when a need for programs and tasks control arised. The most known language from the first generation of such languages is JCL (Job Control Language), created for IBM OS/360 operating system. Modern examples of such languages include shell language, described above, and also text-processing languages, such as sed and awk. These languages were one of the first to directly include support for regular expression matching - a feature that later was included into more general-purpose languages, such as Perl.

2.**Application scripting languages** :Application scripting languages were developed in 1980s, in the era of personal computers, when such important applications as spreadsheets and database clients were introduced, and interactive session in front of the PC became the norm. One of the prime examples of these languages is Microsoft-created Visual Basic language, and especially it's subset named Visual Basic for Applications, designed explicitly for office applications programming

3.**Markup languages** are a special case in the sense that they are not a real programming languages, but rather a set of special command words called 'tags' used to mark up parts of text documents, that are later used by special programs called processors, to do all kinds of transformations to the text, such as displaying it in a browser, or converting it to some other data format. The basic idea of markup languages is the separation of contents and structure, and also including formatting commands and interactive objects into the documents. The first markup language named GML (Generic Markup Language) was created in 1969 by IBM. In 1986, ISO created a standard called SGML, based on GML ideas.

4.**Universal scripting languages** :The languages that belong to this class are perhaps the most well-known. The very term "scripting languages" is associated with them. Most of these languages were originally created for the Unix environment. The goals however were different. The Perl programming language was made for report generation, which is even reflected in its name (Practical Extraction and Report Language). It is commonly said that the primary reason for it's enormous popularity is the ability to write simple and efficient CGI scripts for forming dynamic web pages with this language. Perl was there in the right place at the right time. The Python language was originally made as a tool for accessing system services of the experimental operating system Amoeba. Later it became a universal object-oriented scripting language. Implementations exist for the Java Virtual Machine and also for Microsoft Intermediate Language used on Microsoft .NET platform.

Unlike Perl and Python, which make it easy to write completely standalone programs, Tcl relies heavily on C and C++ extension modules.

1.6 web scripting:

Web is the most fertile areas for the application of scripting languages. Web scripting divides into three areas

a. processing forms

b. creating pages with enhanced visual effects and user interaction and
c. generating pages ’on the fly’ from material held in database.

Processing Web forms:

In the original implementation of the web , when the form is submitted for processing, the information entered by the user is encoded and sent to the server for processing by a CGI script that generates an HTML page to be sent back to the Web browser.

This processing requires string manipulation to construct the HTML page that constitutes the replay, and may also require system access , to run other processes and to establish network connections. Perl is also a language that uses CGI scripting.

Alternatively for processing the form with script running on the server it possible to do some client –side processing within the browser to validate form data before sending it to the server by using JavaScript, VBScript etc.

Dynamic Web pages:

‘Dynamic HTML’ makes every component of a Web page (headings, anchors, tables etc.) a scriptable object. This makes it possible to provide simple interaction with the user using scripts written in JavaScript/Jscript or VBScript, which are interpreted by the browser.

Microsoft’s ActiveX technology allows the creation of pages with more elaborate user interaction by using embedded visual objects called ActiveX controls. These controls are scriptable objects, and can in fact be scripted in a variety languages. This can be scripted by using Perl scripting engine.

Dynamically generated HTML:

Another form of dynamic Web page is one in which some or all of the HTML is generated by scripts executed on the server. A common application of the technique is to construct pages