

The screenshot shows a web browser window with a navigation menu at the top. The menu includes links for 'Developer Site', 'Home', 'Getting', 'Tcl', 'Software', 'Documents', 'Commur', and 'Links'. Below the menu is a search bar with the text 'site:tcl.tk' and two search buttons. The main content area displays the title 'History of Scripting'.

Books | F.A.Q. | Manual Pages | Tutorials | **White Papers** | Y2K | How To Guides | Tcl Extension Architecture

Scripting | Integration | Compiler Effects | Tcl History | **Scripting History**

[JCL](#) | [Unix shells](#) | [Rexx](#) | [Perl](#) | [Tcl](#) | [Visual Basic](#) | [Python](#) | [JavaScript](#) | [Related information](#)

Scripting languages are not new: they have existed since at least the 1960s. However, the power and sophistication of scripting languages have improved dramatically in recent years. When coupled with tremendous increases in computer speed, it has become possible to use scripting languages for a much broader range of applications than was possible previously.

Nearly every major computing platform over the last four decades has supported both system programming languages (for creating applications from scratch) and scripting languages (for integrating components and applications). Here are a few examples of significant scripting languages, in rough chronological order.

JCL (Job Control Language)

Used to sequence job steps and arrange the flow of data in card decks for OS/360, starting in the 1960s. JCL may have been the first widely used scripting language, though it is a relatively primitive one by today's standards.

Unix shells (sh, csh, ksh, ...)

The first shell, sh, was written for the Unix operating system in the early 1970's and a series of others followed over the next 20 years. Shell programs are used both for typing interactive commands and for writing scripts that automate common tasks. One of the most unique and powerful aspects of Unix was the ability to write shell scripts that create new applications by composing existing applications; it is perhaps the single most important reason for Unix's popularity as a platform for application developers.

[Rexx](#)

Michael Cowlshaw initially conceived the Rexx language in 1979 to simplify programming tasks on IBM's CMS timesharing system. It became popular as a macro language for arbitrary application programs, and its usage spread to many other platforms, including PCs and Unix.

[Perl](#)

Created by Larry Wall in the late 1980s as a way to bring together in one place the functions of many popular Unix text processing applications such as sh, sed, and awk, Perl quickly became a favorite tool of system administrators. With the arrival of the World-Wide Web, Perl achieved even greater fame as a convenient way to write CGI scripts for dynamic Web pages.

[Tcl](#)

Created by John Ousterhout in the late 1980s as an embeddable command language for interactive tools. When supplemented with the Tk toolkit, it became popular as the fastest way to build graphical user interfaces on Unix. Tcl and Tk were ported to Windows and the Macintosh in the mid 1990s, producing an outstanding cross-platform development environment. Today Tcl is used for a wide variety of integration applications

including Web content generation, financial applications, electronic design automation, automated testing, and system management.

[Visual Basic](#)

This Microsoft product lies somewhere between a scripting language and a system programming language. It became popular in the early 1990s as the easiest way to create graphical user interfaces under Windows. The combination of Visual Basic and VBX (later ActiveX) components is probably the most successful component framework in existence, due in large part to the ease of integration provided by Visual Basic.

[Python](#)

Python is a dynamic object-oriented language, created by Guido van Rossum in the early 1990s to bridge the gap between shell and C programming. Its elegant, easy to learn syntax, high level data types, elaborate library, portability, and ease of extending and embedding in C/C++ all contribute to its popularity. Originally designed as an advanced scripting language, it found new uses as a rapid application development language for web, database and GUI applications, as well as for distributed systems and mobile code.

[JavaScript](#)

Created in the mid 1990s by Netscape Corporation to perform scripting functions in Web browsers, such as simple form validation. JavaScript has become the de facto standard for client-side Web scripting, although it doesn't have much to do with Java.

Other scripting languages

There are many other important scripting languages that are not described here. Apologies to the authors and advocates of those languages.

Related information

- [More information on the history of Tcl](#): Written by the creator of Tcl, John Ousterhout, this article explains how the idea for Tcl was developed and why it became so popular.
- [Tcl Advocacy](#): Tcl is the leading scripting language for a wide variety of integration application needs. A comparison chart gives you an overview of the features available in each of the most popular scripting languages today.

[Top of page](#)

[Developer Home](#) | [Getting Started](#) | [Tcl Advocacy](#) | [Software Resources](#) | [Documents](#) | [Community](#) | [Links](#)
[Site Map](#) | [Feedback](#) | webmaster@-SPAM-.tcl.tk
[Increase page width](#)

Last modified: April 25, 2001