

|  |
| --- |
|  |

**Python-Ruby-Perl comparison**

Document Control

|  |  |
| --- | --- |
| **Title**: | **Git Flow Implementation** |

|  |  |  |
| --- | --- | --- |
| **Revision history**: | | |
| ***Version No.*** | ***Release Date*** | ***Author*** |
| 0.1 | 26-July-2016 | Saikrishna Gogula |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
| **Significant changes**: | |
| ***Version No.*** | ***Details*** |
| 0.1 | Initial Draft Version |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Availability**: | |
| ***Copy*** | ***Location*** |
| Soft copy |  |
| Hard copy |  |

|  |  |
| --- | --- |
| **General**: | |
| **Approved by**: |  |
| **Released by**: |  |

|  |
| --- |
| **Distribution list**: |

|  |  |  |
| --- | --- | --- |
| **Name** | **Role** | **Approver/Reviewer** |
| Chandra Allaka | Strategy Architect | Reviewer |
| Daniel Lane | Chief Engineer | Approver |
| Mikkal Sveum | Portfolio Testing Infrastructure Manager | Approver |
| Sunil Kothyala | Senior Manager | Approver |

**Table of Contents**

Table of Contents

[1 Overview 5](#_Toc457320924)

[2 Purpose 6](#_Toc457320925)

[3 Python 7](#_Toc457320926)

[4 Perl 8](#_Toc457320927)

[5 Ruby 9](#_Toc457320928)

[6 Differences between Python, Perl and Ruby 10](#_Toc457320929)

[7 Comparison between Python, Perl and Ruby 11](#_Toc457320930)

# Overview

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. Like Perl, Python source code is also available under the GNU General Public License (GPL).

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, and GUI development.

Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write.

# Purpose

The purpose of this document is to provide an overview on Python, Perl, Ruby programming languages and compare the three programming languages features.

# Python

Python is a general purpose programming language created in the late 1980s, and named after Monty Python, that’s used by thousands of people to do things from testing microchips at Intel, to powering Instagram, to building video games with the PyGame library. It’s small, very closely resembles the English language, and has hundreds of existing third-party libraries.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and UNIX shell and other scripting languages.

**Features:**

**Readability:** Python very closely resembles the English language, using words like ‘not’ and ‘in’ to make it to where you can very often read a program, or script, aloud to someone else and not feel like you’re speaking some arcane language. This is also helped by Python’s very strict punctuation rules which means you don’t have curly braces ({ }) all over your code.

**Libraries:** Python has been around for over 20 years, so a lot of code written in Python has built up over the decades and, being an open source language, a lot of this has been released for others to use. Almost all of it is collected on https://pypi.python.org, pronounced “pie-pee-eye” or, more commonly called “the CheeseShop”. You can install this software on your system to be used by your own projects. For example, if you want to use Python to build scripts with command line arguments, you’d install the “click” library and then import it into your scripts and use it. There are libraries for pretty much any use case you can come up with, from image manipulation, to scientific calculations, to server automation.

**Portable:** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

**Extendable:** You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

**Scalable:** Python provides a better structure and support for large programs than shell scripting.

Python has a big list of good features, few are listed below:

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* It supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

# Perl

Perl is a script programming language that is similar in syntax to the C language and that includes a number of popular UNIX facilities such as sed, awk, and tr. Perl is an interpreted language that can optionally be compiled just before execution into either C code or cross-platform byte code. When compiled, a Perl program is almost (but not quite) as fast as a fully precompiled C language program. Perl is regarded as a good choice for developing common gateway interface (CGI) programs because it has good text manipulation facilities (although it also handles binary files).

Traditional compilers convert programs into machine language. When you run a Perl program, it's first compiled into a byte code, which is then converted (as the program runs) into machine instructions. So it is not quite the same as shells, or Tcl, which are strictly interpreted without an intermediate representation.

Perl was originally developed by Larry Wall in 1987 as a general-purpose UNIX scripting language to make report processing easier.

**Features:**

* Perl takes the best features from other languages, such as C, awk, sed, sh, and BASIC, among others.
* Perl’s database integration interface DBI supports third-party databases including Oracle, Sybase, Postgres, MySQL and others.
* Perl works with HTML, XML, and other mark-up languages.
* Perl supports Unicode.
* Perl is Y2K compliant.
* Perl supports both procedural and object-oriented programming.
* Perl interfaces with external C/C++ libraries through XS or SWIG.
* Perl is extensible. There are over 20,000 third party modules available from the Comprehensive Perl Archive Network
* The Perl interpreter can be embedded into other systems.

It is also not like most versions of C or C++, which are compiled directly into a machine dependent format. It is somewhere in between, along with Python and awk and Emacs .elc files.

# Ruby

Ruby is a pure object-oriented programming language. It was created in 1993 by Yukihiro Matsumoto of Japan.

Ruby has features that are similar to those of Smalltalk, Perl, and Python. Perl, Python, and Smalltalk are scripting languages. Smalltalk is a true object-oriented language. Ruby, like Smalltalk, is a perfect object-oriented language. Using Ruby syntax is much easier than using Smalltalk syntax.

**Features:**

* Ruby is an open-source and is freely available on the Web, but it is subject to a license.
* Ruby is a general-purpose, interpreted programming language.
* Ruby is a true object-oriented programming language.
* Ruby is a server-side scripting language similar to Python and PERL.
* Ruby can be used to write Common Gateway Interface (CGI) scripts.
* Ruby can be embedded into Hypertext Markup Language (HTML).
* Ruby has a clean and easy syntax that allows a new developer to learn Ruby very quickly and easily.
* Ruby has similar syntax to that of many programming languages such as C++ and Perl.
* Ruby is very much scalable and big programs written in Ruby are easily maintainable.
* Ruby can be used for developing Internet and intranet applications.
* Ruby can be installed in Windows and POSIX environments.
* Ruby support many GUI tools such as Tcl/Tk, GTK, and OpenGL.
* Ruby can easily be connected to DB2, MySQL, Oracle, and Sybase.
* Ruby has a rich set of built-in functions, which can be used directly into Ruby scripts.

There are libraries for building all types of applications with Ruby: XML parsers, GUI bindings, networking protocols, game libraries and more. Ruby programmers also have access to the powerful RubyGems program. Comparable to Perl's CPAN, RubyGems makes it easy to import other programmers' libraries into your own programs.

# Differences between Python, Perl and Ruby

* In Perl and Ruby, you should declare name spaces all the way. In Python, on the other hand, each source file has its own name space and the name space name and file name is the same. This way is convenient to avoid name collisions.
* The coding style of modules is substantially different from that of ordinary programs in Perl and Ruby, which means that you should modify your program considerably to use as a module. In Python, the coding style of modules and ordinary programs are same and you can use your small script as a module with a few modifications such as adding \_\_all\_\_ statement. You can use a module as a stand-alone program as well.
* Perl programs basically read file line by line using while (<>). Python and Ruby programs, on the other hand, read all file contents at once using read(). The difference may be because that memory price got cheaper in 1990s when Python and Ruby appeared and that plenty of memory space became available.
* Perl programs basically use for each block while Perl has list operating functions such as map and grep, Python programs use list comprehensive while it also has the functions such as map and filter and Ruby programs operate list using methods with block such as map, select, and grep.
* Perl uses unusual way to hand parameters to functions. Python and Ruby adopt a conventional way.
* Functions of Perl and Ruby can return a function but Python can't (basically). Python uses a class of functions instead.
* Perl and Python support higher order functions i.e. Perl can give a reference of function as a parameter, In Python; function names represent pointers to the functions. The list of parameters required to call as function. For instance, if function foo is defined, foo (...) is to call the function and foo represent the pointer to the function. Pointers to function can be used as arguments or items of for loops. Whereas Ruby does not support higher order functions
* Lazy evaluation is available on Python and Ruby
* The online document of Python is excellent. That of Ruby is not good very much.
* Ruby is an object orientated dialect of Lisp. Python takes many good features from many languages. Basically it is a weakly-typed C++ or Java. But it also borrows some features from functional programming languages such as Haskell and Lisp.
* As Python does not force the object orientated style, beginners can learn it easily.
* Writability is not so different each other. Readability, however, differs tremendously: Python is easiest to read, and then Ruby follows. Perl code cannot be read if not commented enough nor carefully decided parameter names.

# Comparison between Python, Perl and Ruby

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Python** | **Ruby** | **Perl** |
| Writability | A | A | A |
| Readability | A+ | A | C |
| Library | A | B | A |
| Run time | A | B | A |
| Online Documentation | A | B | A+ |
| Number of users | A | B | A+ |

**Note:** A+ is considered to be the highest grade and C is the lowest for the mentioned parameters