Titile of the project : **Scripting & Automation**

Name : Lavanya Chilukamari

Inter ID : 308

🛠️ **Tool Name :** “Perl Ruby”.

Description : “PERL” is a high-level, interpreted scripting language designed for text processing, automation, and web development. **“**Ruby” is a dynamic, object-oriented, reflective, general-purpose programming language. It follows a flexible syntax that allows both procedural and object-oriented programming.

📌 **What is this about?**

- With “PERL” scripting, developers can automate tasks, manipulate text files, interact with databases, and even develop web applications. Learning PERL opens up numerous career opportunities, especially in fields like system administration, cybersecurity, and data analytics.
- “Ruby” Automation Framework is a collection of tools and practices used to automate the process of testing software applications using the Ruby programming language. It helps developers and testers create and execute automated test cases efficiently.

🔑 **Key Characteristics & features of Perl Ruby.**

- Mission critical. Used for mission critical projects in the public and private sectors.
- Object-oriented, procedural and functional. Supports object-oriented, procedural and functional programming.
- Easily extendible.
- Text manipulation.
- Unicode support.
- Database integration.
- C/C++ library interface.
- Embeddable.
- Elegant and Readable Syntax. Ruby's syntax is designed to be easily understood.
- Pure Object-Oriented Design. Ruby is a purely object-oriented language.
- Dynamic Typing.
- Duck Typing.
- Metaprogramming.
- Flexibility.
- Blocks.
- Web Development with Ruby on Rails.

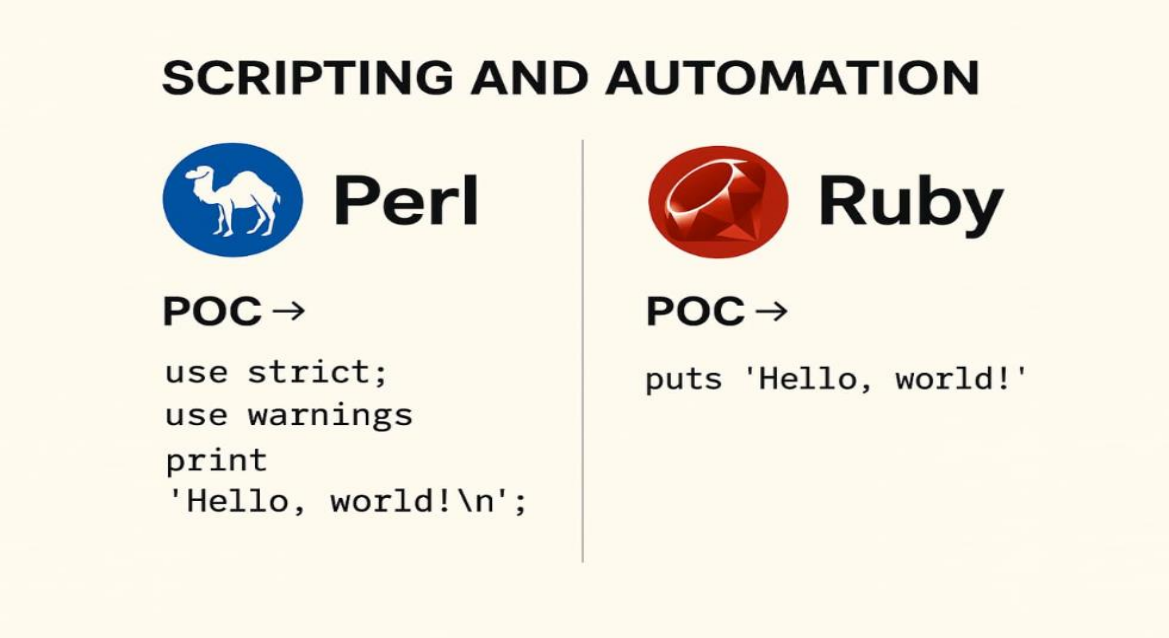
🧩 Types & Modules Available in Perl Ruby.

- Archive-Zip-1.68.
- autovivification-0.18.
- Business-ISBN-3.012.
- Business-ISMN-1.205.
- Business-ISSN-1.008.
- Class-Accessor-0.51.
- Data-Compare-1.29.
- Data-Dump-1.25.

🔄 How will this Tool Help.

- Desktop apps, static websites, data processing services, automation tools.
- Ruby is a dynamic, open source, and reflecting programming language.
- Web servers, DevOps, web scraping and crawling.
- Rails app framework = database-driven apps.
- Perl is complex and flexible.
- It has comprehensive text processing capabilities.

📸 Proofs of Concept (PoC) Images.



💻 Sample Perl Snippet (Eyer-style Minimalist)

```perl
use LWP::UserAgent;
use HTTP::Request::Common qw(POST);
use JSON;

my $ua = LWP::UserAgent->new;

# Login
my $res = $ua->request(POST 'http://localhost/api/login', [
  user => 'admin',
  pass => 'secret'
]);

die "Login failed" unless $res->is_success;

# GET status
my $status_res = $ua->get("http://localhost/api/status");
my $json = decode_json($status_res->decoded_content);

print "System A: $json->{system_a}{status}\n";
print "System B: $json->{system_b}{status}\n";
```

Automated Login + Fetch System Status from a Web Page

```perl
#!/usr/bin/perl
use strict;
use warnings;
use LWP::UserAgent;
use HTTP::Request::Common qw(POST);
use HTML::TreeBuilder;

# Setup user agent
my $ua = LWP::UserAgent->new;
$ua->agent("Mozilla/5.0");

# Step 1: Login to the site
my $login_url = "http://example.com/login";
my $response = $ua->request(POST $login_url, [
    username => 'admin',
    password => 'secret123'
]);
```

```
die "Login failed!" unless $response->is_success;

# Step 2: Access protected page
my $dashboard_url = "http://example.com/dashboard";
my $dashboard_res = $ua->get($dashboard_url);
die "Failed to load dashboard!" unless $dashboard_res->is_success;

# Step 3: Parse HTML content
my $tree = HTML::TreeBuilder->new;
$tree->parse($dashboard_res->decoded_content);

# Example: Extract page title and system status
my $title = $tree->look_down(_tag => 'title')->as_text;
my $status = $tree->look_down(id => 'system-status')->as_text;

print "Page Title: $title\n";
print "System Status: $status\n";

$tree->delete;
```

ruby

```ruby
require 'httparty'
require 'nokogiri'

# Step 1: Login
login_url = "http://example.com/login"
login_data = { username: 'admin', password: 'secret123' }

response = HTTParty.post(login_url, body: login_data)
cookies = response.headers['set-cookie']

# Step 2: Access dashboard with cookies
dashboard_url = "http://example.com/dashboard"
dashboard_res = HTTParty.get(dashboard_url, headers: { 'Cookie' => cookies })

# Step 3: Parse with Nokogiri
doc = Nokogiri::HTML(dashboard_res.body)
title = doc.at('title').text
status = doc.at_css('#system-status').text

puts "Title: #{title}"
puts "System Status: #{status}"
```

```ruby

source 'https://rubygems.org'
gem 'httparty'
gem 'nokogiri'
```

📝 15-Liner Summary.

1. Perl stands for *Practical Extraction and Report Language*, originally developed for text processing and report generation.
2. Ruby is an object-oriented programming language designed for simplicity and productivity, created by Yukihiro Matsumoto.
3. Perl became popular in the 1990s for its powerful regular expressions and system admin scripting.
4. Ruby gained traction in the 2000s with the rise of the web framework Ruby on Rails.
5. Both are high-level, interpreted languages.
6. Perl is known for its motto *"There's more than one way to do it"*, leading to flexible but sometimes unreadable code.
7. Ruby emphasizes readability and elegance with the motto *"Optimized for developer happiness."*
8. Perl uses a mix of procedural and object-oriented styles; Ruby is purely object-oriented—everything is an object.
9. Perl scripts often use sigils ($, @, %) to denote variable types, while Ruby does not.
10. Ruby has cleaner syntax, making it easier for beginners compared to Perl's complex, cryptic style.
11. CPAN is Perl's vast library of modules, while RubyGems serves the same purpose for Ruby.
12. Perl excels in text processing, system administration, and bioinformatics.
13. Ruby shines in rapid web development and agile applications.
14. Community-wise, Perl is more niche today, while Ruby still holds strong, especially with Rails.
15. Both remain open-source and are suitable for scripting, but Ruby often wins in maintainability and readability.

⏱️ Time to Use / Best Case Scenarios.

- Perl shines in text processing, one-liners, and legacy systems.
- Text processing / regex-heavy tasks.
- Log file parsing.
- System administration scripting.
- Legacy codebases.
- Bioinformatics & scientific scripting.

- Ruby excels in web development, developer-friendly scripting, and DSLs (Domain-Specific Languages).
- Web development.
- Readable, elegant scripts.
- Automation / DevOps tooling.
- Test scripting.
- Creating DSLs.

🕵️ When to Use During Investigation.

- Parsing and extracting data from log files (e.g., /var/log/auth.log, web server logs).
- Regex-heavy tasks like scanning logs for suspicious IPs, URLs, or patterns.
- Automating repetitive tasks, like renaming or moving large volumes of files or emails.
- Legacy system compatibility — many older investigation tools and scripts are written in Perl.
- It working with Metasploit Framework for exploitation/investigation.
- It can readable, object-oriented scripting.
- They can building or extending investigation tools quickly.
- There are automating API interactions (e.g., querying a SIEM or threat intel platform).

🎯 Best Person to Use This Tool &Required Skills.

- **Best User :** Understands the Domain of the Tool, is Comfortable With CLI Tools, it can Read and Modify Scripts, is Resourceful, it knows Version Management.
- **Required Skills :**
    - Basic Perl Scripting
    - Regular Expressions (Regex)
    - CPAN Usage
    - File Handling & Text Processing
    - Shebang, Permissions, Execution
    - Basic Ruby Programming
    - Gems & Bundler
    - Scripting for Automation
    - Understanding of Rake Tasks
    - Using IRB & Debuggers

⚠️ Flaws / Suggestions to Improve.

- Adopt Modern Perl (v5.36+) Practices
- Move to Moose / Moo / Modern::Perl for OOP
- Standardize Code Style
- Replace with Raku (Perl 6)
- Upgrade to Ruby 3+
- Use Sorbet or RBS for Type Safety

- Limit Rails Magic
- Optimize Memory Usage
- Explore Alternatives When Needed

🌟 **Good About The Tool.**

In Perl :

- Common in old infra
- Used in bioinformatics

In Ruby :

- Ruby on Rails is top-notch
- Easy to maintain and elegant