

# Comparative Evaluation: Ruby vs Python in Modern Programming Environments

Faiza Mehram  
faizamehram786@gmail.com

## I. ABSTRACT

This paper provides an in-depth comparison between Ruby and Python, two widely utilized programming languages, encompassing a range of aspects including features, performance metrics, and industry adoption. It goes beyond syntax to delve into semantics, ease of learning, code maintainability, and community support. Through meticulous evaluation, this analysis seeks to highlight the strengths and weaknesses inherent in each language, serving as a guide for developers, educators, and decision-makers in selecting languages aligned with the demands of 2024. Additionally, it explores emerging trends and advancements in Ruby and Python, offering valuable insights into the evolving landscape of modern software development and aiding stakeholders in making optimal language choices.

## II. INTRODUCTION

In today's rapidly evolving programming landscape, the choice of programming language plays a crucial role in determining the success and efficiency of software development projects. Among the excess of options available, Python and Ruby stand out as dynamic, versatile languages that cater to diverse programming needs. This introduction delves into a comparative analysis of Python and Ruby, exploring their features, strengths, and applications in modern software development.

Python, famous for its simplicity, readability, framework, and extensive libraries, has become a dominant force in AI, data science, machine learning, and scientific programming. Python's clean and easy-to-understand syntax, ability to work across different platforms without issues, and its wide range of built-in tools and resources make it a top pick for developers dealing with complex programming tasks and creating apps that can grow consistent.

In contrast, Ruby's strength lies in web development, particularly with its powerful Ruby on Rails framework. Ruby's expressive syntax, object-oriented paradigm, and focus on code flexibility make it ideal for rapid web application development. Its community support, security features, and built-in protections against common web vulnerabilities enhance its appeal for web-apps projects.

In conclusion, while Python excels in complex programming tasks and scalable application development, Ruby's prowess in web development and rapid application deployment

makes it a strong contender, catering to distinct needs in the dynamic programming landscape of today.

## III. FEATURES AND USE CASES

### A. Exception handling

Exception handling is a fundamental feature supported by Python and Ruby, with some syntactic differences between the two. In Python, exception handling is done using "try/except/finally" and "raise" to trigger new exceptions. On the other hand, Ruby utilizes "begin/rescue/ensure" and "raise" for exception handling. Python introduced an additional feature in version 2.5, allowing objects to define standard cleanup actions. This feature reduces the need for "finally" blocks, streamlining source code and ensuring proper cleanup, which can have a positive impact on performance in larger programs. For more information, refer to PEP 343.

### B. Duck Typing

Duck typing, a polymorphism paradigm in dynamically typed object-oriented languages, emphasizes object properties and methods over explicit type definitions. Both Ruby and Python promote and encourage the use of duck typing, providing appropriate exceptions to catch errors. This approach allows developers to focus on object behavior rather than strict type definitions. Programmers embracing the duck typing philosophy may find Ruby and Python more suitable languages compared to others that rely heavily on type declarations and inheritance for polymorphism.

### C. Performance Comparison

Performance is crucial for web applications, influenced by factors like network speed, database connectivity, and server hardware. Ruby and Python, being scripting languages, are interpreted at runtime, affecting their benchmark rankings. Ruby notably improved from version 1.8 to 1.9. Both languages offer caching systems like memcached for speed optimization. Overall, Ruby and Python perform comparably in terms of performance for web applications.

### D. Popularity

According to data from GitHub report, there has been a noticeable decline in Ruby's popularity, dropping from the fifth most used language to the tenth spot. On the other hand, Python has shown consistent growth, climbing from the fourth position to the third spot.

### E. Learning curve

Python is a “quick to learn” programming language. The syntax is easier to understand, and it’s easier for beginners. But beyond that, we’re going to need to make decisions on what framework to use beyond bare-bones Python. Ruby might take more time to get used to, but Ruby on Rails has Plagiarism Check built-in features—like scaffolding and Active Record—to accelerate development.

### F. Reusable code

Publicly available and ready-to-use code is a relevant factor when you need to decide on a programming language. Python calls them “modules,” and they’re available via PyPI where you can search more than 150,000 modules. On the other hand, reusable code in Ruby is called Gems, and there are close to 150,000 gems. But the differentiating factor is filtering; PyPI allows filtering by categories like “development status,” which is more straightforward than comparing many libraries and manually evaluating their code.

### G. Flexibility

In this aspect, Ruby has inherited Perl’s philosophy: “There’s more than one way to do it.” This will always find many different methods to achieve a task in Ruby. Depending on who’s writing the code, this might lead to unnecessary complexity and obfuscation. On the other hand, Python follows an approach where simplicity has more value than complexity (“The Zen of Python”) Hence its philosophy is “There should be one and preferably only one—obvious way to do it.” So, although Python code probably won’t be the most flexible, it has a good chance of being more readable to an inexperienced programmer.

### H. Job Opportunities

Python boasts numerous job opportunities across the USA, Pakistan, China, UAE, India, Russia, Germany, France, and Japan. It’s particularly prevalent in data science, machine learning, web development, and automation roles, with demand growing in various industries. Ruby, while less prevalent than Python, still offers job prospects in web development across these countries. In the USA, startups and tech firms seek Ruby on Rails (RoR) developers. India, Pakistan, and other nations also have openings, albeit fewer than Python, mainly in web development roles.

8

## IV. COMPARISON OF RUBY AND PYTHON ASPECTS

### A. Syntax

#### Variable Declaration in Ruby

```
kitty = "Mr.Kitty"
i\_am\_ten = 10
puts kitty
# => "Mr.Kitty"
```

### Variable declaration in Python

```
kitty = "Mr.Kitty"
i_am_ten = 10

print(kitty)
# Output: "Mr.Kitty"
```

### B. Dictionaries

#### In Ruby

```
cat_hash = {'Jason' => 'Mr.Kitty',
'Lauren'
=> 'Cooper', 'Antony' => 'Manfred' }
cat_hash['Jason']
# => "Mr.Kitty"
```

#### In Python

```
cat_dict = {'Jason' : 'Mr.Kitty', 'Lauren'
: 'Cooper', 'Antony' : 'Manfred' }
cat_dict['Jason']
# => "Mr.Kitty"
```

### C. Indentation

In python,

```
def greet(name):
    if name:
        greeting = "Hello, " + name
        print(greeting)
    else:
        print("Hello, what's your name?")

greet("Alice")
greet("")
```

In Ruby,

```
def greet(name)
    if name != ""
        greeting = "Hello, " + name
        puts greeting
    else
        puts "Hello, what's your name?"
    end
end

greet("Alice")
greet("")
```

### D. Lists

In python,

```
cat_list = []
cat_list.append("Mr.Kitty")
cat_list.append("Meowbama")
print(cat_list)
```

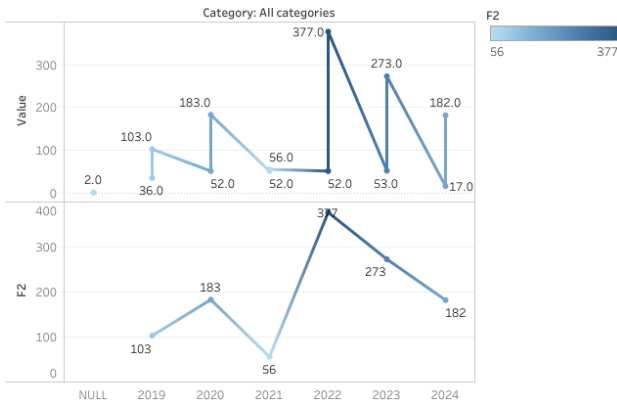
In Ruby,

```
cat_list = ["Mr.Kitty", "Meowbama",
"Cooper", "Manfred"]
cat_list.each do |cat|
  puts cat
end

puts "That's a lot of cats"
```

## V. WORLDWIDE USAGE

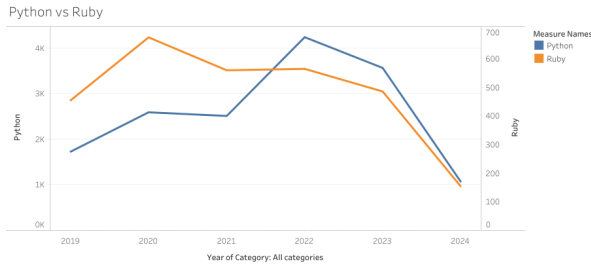
Interest over time for past 5 years



(a) Interest rate over the past 5 years with Ruby and Python



(b) Regional data Worldwide of Ruby and Python usage



(c) Ruby and Python usage in Pakistan

## VI. COMPARISON FRAMEWORKS

Frameworks is an empty platform for developing static and dynamic pages. It's a group of libraries and predefined instructions which assist to lessen the work load and time of a software developer. It is reusable and extensible platform with most recent technology and pattern. A developer can add new

functionality on every occasion he needs based totally on their necessities. Frameworks always keep a fashionable course for growing and deploying programs with a re-usability mode. Here, in case of Python and Ruby they have got a massive series of frameworks in special regions.

### FRAMEWORK COMPARISON

ID	Frameworks	Release Date
Ruby	1 Ruby 3.3.1	2024-04-23
	2 Ruby 3.2.4	2024-04-23
	3 Ruby 3.1.5	2024-04-23
	4 Ruby 3.0.7	2024-04-23
	5 Ruby 3.2.3	2024-01-18
	6 Ruby 3.3.0	2023-12-25
	7 Ruby 3.3.0-rc1	2023-12-11
	8 Ruby 3.3.0-preview3	2023-11-12
	9 Ruby 3.3.0-preview2	2023-09-14
	10 Ruby 3.3.0-preview1	2023-05-12
	11 Ruby 3.2.2	2023-03-30
	12 Ruby 3.1.4	2023-03-30
	13 Ruby 3.0.6	2023-03-30
	14 Ruby 2.7.8	2023-03-30
	15 Ruby 3.2.1	2023-02-08
	16 Ruby 3.2.0	2022-12-25
	17 Ruby 3.2.0-rc1	2022-12-06
	18 Ruby 3.1.3	2022-11-24
	19 Ruby 3.0.5	2022-11-24
	20 Ruby 2.7.7	2022-11-24
	21 Ruby 3.2.0-preview3	2022-11-11
	22 Ruby 3.2.0-preview2	2022-09-09
	23 Ruby 3.1.2	2022-04-12
	24 Ruby 3.0.4	2022-04-12
	25 Ruby 2.7.6	2022-04-12
	26 Ruby 2.6.10	2022-04-12
	27 Ruby 3.2.0-preview1	2022-04-03
	28 Ruby 3.1.1	2022-02-18
	29 Ruby 3.1.0	2021-12-25
	30 Ruby 3.0.3	2021-11-24
Python	1 Python 3.12.3	April 9, 2024
	2 Python 3.11.9	April 2, 2024
	3 Python 3.10.14	March 19, 2024
	4 Python 3.9.19	March 19, 2024
	5 Python 3.8.19	March 19, 2024
	6 Python 3.11.8	Feb. 6, 2024
	7 Python 3.12.2	Feb. 6, 2024
	8 Python 3.12.1	Dec. 8, 2023
	9 Python 3.11.7	Dec. 4, 2023
	10 Python 3.12.0	Oct. 2, 2023
	11 Python 3.11.6	Oct. 2, 2023
	12 Python 3.11.5	Aug. 24, 2023
	13 Python 3.10.13	Aug. 24, 2023
	14 Python 3.9.18	Aug. 24, 2023
	15 Python 3.8.18	Aug. 24, 2023
	16 Python 3.10.12	June 6, 2023
	17 Python 3.11.4	June 6, 2023
	18 Python 3.7.17	June 6, 2023
	19 Python 3.8.17	June 6, 2023
	20 Python 3.9.17	June 6, 2023
	21 Python 3.10.11	April 5, 2023
	22 Python 3.11.3	April 5, 2023
	23 Python 3.10.10	Feb. 8, 2023
	24 Python 3.11.2	Feb. 8, 2023
	25 Python 3.11.1	Dec. 6, 2022
	26 Python 3.10.9	Dec. 6, 2022
	27 Python 3.9.16	Dec. 6, 2022
	28 Python 3.8.16	Dec. 6, 2022
	29 Python 3.7.16	Dec. 6, 2022
	30 Python 3.11.0	Oct. 24, 2022

TABLE I: Release Versions

Library is a set of predefined operations or code to perform an activity while it's far invoked. It is a non-volatile reusable

characteristic that decide the overall performance of a language. A language having a terrific library [8],[40] help will certainly flip to a strength complete language. Because library is a fundamental building block of coding. Here are different libraries of Ruby and Python.

#### LIBRARIES COMPARISON

No.	Libraries
Python	<b>Admin Panels</b>
	Ajenti, django-grappelli, django-jet, django-suit, django-xadmin, flask-admin, flower, wooye
	<b>Authentication</b>
	OAuth (authlib, django-allauth, django oauth2, python-social-auth)
	JWT (pyjwt, python-jose, python-jwt)
	<b>Caching</b>
	Beaker, django-cache-machine, python-diskcache, django-cache, dogpile.cache, HermesCache, pylibmc
	Code Analysis (coala, code2flow, prospector, pycall-graph)
	Code Linters (flake8, pylint, pylama)
	Code Formatters (black, yapf)
	Static Type Checkers (mypy, pyre-check)
	Static Type Annotations Generators
	<b>Command-line Tools</b>
	Productivity (cookiecutter, doitlive, howdoi, Path-Picker, percol, thefuck, tmuxp, try)
	CLI Enhancements (httplib, kube-shell, mycli, pgcli, saws)
	<b>Concurrency and Parallelism</b>
	Concurrent features ,multiprocessing , Evenlet,scoop
	<b>Configurations</b>
	Configobj,configparser
	<b>Database Drivers</b>
	Mysql, PostgreSQL, Other Relational Databases, NoSQL Databases, Asynchronous Clients
	<b>Database</b>
	tpxipckosletgDrBes.. ttinxRydebd.ixZ)ODB
Ruby	<b>Admin Interface</b>
	Active Admin, ActiveScaffold, Typus Adminstrate, bhf, Trestle, Rails Admin
	<b>Authentication and OAuth Authorization</b>
	Authlogic, Clearance, Device, JWT, Knock Monban, OmniAuth, Rodath, Sheild, Sorcey, warden OAuth (Doorkeeper, OAuth2), Authorization
	<b>Caching</b>
	Actioncaching for Action Pack, Dalli, Garner, IdentityCache, Kashmir, Readthis, Record Cache
	<b>Code Analysis and Metrics</b>
	Barkeep, Brakeman, Cane, Sorbet, Coverband, Fasterer, Flay, Scientist, FLog, Fukuzatsu, Metric Fu, Reek, Pippi, rails_best_practices, Pronto, RuboCop, RubyCritic, SimpleCov, Traceroute
	<b>CLI Builder</b>
	Clamp, cmdparse, Commander, GLI, Hanami CLI, Main, Optimist, Rake, Slop, Terrapi, Thor, TTY
	<b>CLI Utilities</b>
	Awesome Print, Betty, colorize, coloris, formatador, Paint, Tabulo, Pastel, Ru, Ruby, ProgressBar
	<b>Concurrency and Parallelism</b>
	Celluloid, Concurrent Ruby, forkoff, EventMachine, Parallel
	<b>Configuration</b>
	Champer, Configatron, Configs, detenv, Econfig, EN-Vied, Figaro, Global, Sail, RailsConfig
	<b>Data Visualization</b>
	Chartkick, GeoPattern, LazyHighChart, RailRoady, Rails Erd, Ruby, GraphViz

TABLE II: Python and Ruby Libraries

#### VII. ADVANTAGES AND DISADVANTAGES

##### A. Ruby

###### Advantages

- Ruby's syntax is designed to be readable and easy to write, making it popular among developers.
- Ruby has a strong community and a wide range of libraries and frameworks like Ruby on Rails for web development.
- Ruby's metaprogramming features allow for dynamic code generation and flexibility.
- Ruby is purely object-oriented, which promotes good design practices and code organization.
- Ruby's concurrency model, especially with tools like Sidekiq, supports efficient handling of concurrent tasks.

###### Disadvantages

- Ruby can be slower compared to languages like Python or C++ due to its interpreted nature.
- Ruby applications can consume more memory compared to some other languages.
- While Ruby's syntax is elegant, mastering its full potential, especially with metaprogramming, can take time.
- While Ruby has a strong community, it's not as large as some other languages, which can impact support and resources for less common tasks.

##### B. Python

###### Advantages

- Python is highly versatile, used for web development, data science, artificial intelligence, scripting, and more.
- Python has a massive community, leading to extensive documentation, libraries like NumPy, pandas, and TensorFlow, and strong support.
- Python's syntax emphasizes readability and clean code, making it beginner-friendly and great for collaboration.
- Python scales well, from small scripts to large-scale applications and systems.
- Python integrates well with other languages and platforms, enhancing its capabilities in diverse environments.

###### Disadvantages

- While improvements have been made, Python can still be slower than compiled languages for certain tasks.
- The GIL in Python can limit true parallelism in multi-threaded applications, although this is being addressed in newer versions.
- Compared to languages like Swift or Java, Python's presence in mobile app development is more limited.
- While Python's dynamic typing can be flexible, it can also lead to runtime errors if not managed carefully.

#### VIII. COMPARISON THROUGH APPS

##### Admin Panels

- **Python:** Ajenti, django-grappelli, django-jet, django-suit, django-xadmin, flask-admin, flower, wooye
- **Ruby:** Active Admin, ActiveScaffold, Typus Adminstrate, bhf, Trestle, Rails Admin



Fig. 2: Apps with Ruby programming language

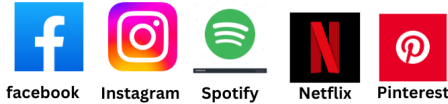


Fig. 3: Apps with Python Programming language

#### Authentication and OAuth Authorization

- **Python:** OAuth (authlib, django-allauth, django oauth2, python-social-auth), JWT (pyjwt, python-jose, python-jwt)
- **Ruby:** Authlogic, Clearance, Devise, JWT, Knock Monban, OmniAuth, Rodath, Sheild, Sorcey, warden OAuth (Doorkeeper, OAuth2), Authorization

#### Caching

- **Python:** Beaker, django-cache-machine, python-diskcache, django-cache, dogpile.cache, HermesCache, pylibmc
- **Ruby:** Actioncaching for Action Pack, Dalli, Garner, IdentityCache, Kashmir, Readthis, Record Cache

#### Code Analysis and Metrics

- **Python:** Code Analysis (coala, code2flow, prospector, pycallgraph), Code Linters (flake8, pylint, pylama), Code Formatters (black, yapf), Static Type Checkers (mypy, pyre-check), Static Type Annotations Generators
- **Ruby:** Barkeep, Brakeman, Cane, Sorbet, Coverband, Fasterer, Flay, Scientist, FLog, Fukuzatsu, Metric Fu, Reek, Pippi, rails\_best\_practices, Pronto, RuboCop, RubyCritic, SimpleCov

#### Database Tools

- **Python:** tpxipckosletgDrBes.. ttinxRydebd.ixZ)ODB
- **Ruby:** Connection\_pool, Database Cleaner, Foreigner, Large Hadron, Lol DBA, Polo, PgHero, Rails DB, Sencic, SchemaPlus, Seed dump, Upsert, Migrator

In 2024, Python continues to dominate the programming landscape due to its versatility, extensive libraries, and straightforward syntax. Its widespread adoption in fields like data science, machine learning, and web development stems from its ease of use and powerful frameworks such as Django and Flask. Python's robust community support and continuous development ensure that it remains at the forefront of technological advancements, making it a reliable choice for a wide range of applications.

On the other hand, Ruby retains its appeal, particularly in web development, thanks to the Ruby on Rails framework.

Known for its developer-friendly nature and convention over configuration approach, Ruby on Rails enables rapid prototyping and streamlined development processes. While its usage may not be as ubiquitous as Python's, Ruby's emphasis on programmer happiness and elegant code design continues to attract developers looking for an expressive and efficient language for building web applications and services.

#### IX. CONCLUSION

In conclusion, the comparison between Ruby and Python reveals distinct strengths and areas of specialization for each language. Python's extensive libraries and versatility make it a top choice for data science, machine learning, and a wide range of application domains. Its simplicity and widespread adoption contribute to its continued growth and relevance in the industry.

On the other hand, Ruby, particularly when paired with the Ruby on Rails framework, excels in web development by offering a developer-friendly environment and rapid development capabilities. Its emphasis on programmer happiness and elegant code design makes it well-suited for building web applications and services efficiently. The decision between Ruby and Python ultimately hinges on project-specific requirements, team expertise, and long-term maintenance considerations. For data-intensive tasks and diverse application needs, Python's ecosystem and flexibility stand out. Conversely, for web development projects focusing on quick deployment and code elegance, Ruby with Ruby on Rails provides a compelling solution. In professional practice, understanding the unique strengths of each language and aligning them with project goals is crucial for making informed decisions that lead to successful software development outcomes.

#### REFERENCES

- [1] International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958 (Online), Volume-9 Issue-3, February 2020
- [2] KR Srinath, Python- "The fastest Growing Programming Language", International Research Journal of Engineering and Technology (IRJET), p-ISSN: 2395-0072, Volume: 04, Issue: 12, Dec-2017
- [3] Fankar Armash Aslam, Hawa Nabeel Mohammed, Jummal Musab Mohd, Munir, Murade Aaraf Gulamgaus, Prof. P. S. Lokhande, Case study and report in "Efficient Way Of Web Development Using Python And Flask", International Journal of Advanced Research in Computer Science, ISSN No.0976-5697, Volume 6, No. 2, March-April 2015
- [4] Patrick Hill, Uma Kanagaratnam, "Python Machine Learning Sebastian Raksha", Institute of Electrical and Electronics Engineers (IEEE), Print ISSN: 1746-5702, Volume 58, Issue 3, Autumn 2016, page 64, Doi: 10.1093/itnow/bww088, 31 August-2016
- [5] Leon Thurner, Alexander Scheidler, FLorain Schafer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, Martin Braun, "Pandapower- An open source Python tool for Convenient Modeling, Analysis and Optimization of Electric power system", Institute of Electrical and Electronics Engineers (IEEE), Print ISSN: 0885-8950, Volume: 33, Issue 6, Nov. 2018.
- [6] Miti S Bhat, Deepthi G Nair, Devyani Bansal, J. Vaishnavi, Data Structure based Performance Evaluation of Emerging Technologies A Comparison of Scala, Ruby, Groovy, and Python, Institute of Electrical and Electronics Engineers (IEEE), Print ISBN: 978-1-4673-2177-8, Doi: 10.1109/CONSEG.2012.6349515, 12 November 2012.
- [7] Prof. P. Karthikeyan and Rama Pandian M, Research paper in "Robust Application with RoR", International Journal of Advanced Research in Computer Science, ISSN No. 0976-5697, Volume 1, No. 4, Nov-Dec 2010.
- [8] A Bogdanchikov1, M Zhaparov1 and R Suliyev, "Python to learn programming", IOP Publishing Ltd, Journal of Physics: Conference Series 423 (2013) 012027, doi: 10.1088/1742-6596/423/1/012027 (2013).

- [9] Prajakta Deshpande, Vikrant Shaga, Surabhi Thorat, "Review of sentiment analysis on twitter data using Python", International Journal of Advanced Research in Computer Science, ISSN No. 0976-5697, Volume 8, No. 9, November-December 2017.
- [10] Aji Joy, "Design of a Computer Numeric Control System with Open Software Tools", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, ISSN (Print) : 2320 – 3765, Vol. 3, Issue 2, February 2014.
- [11] Mohd. Hamzah Khan and Ihtiram Raza Khan, Review article in "Malware Detection and Analysis", International Journal of Advanced Research in Computer Science, ISSN No. 0976-5697, Volume 8, No. 5, May – June 2017.
- [12] L. Prechelt, "An empirical comparison of seven programming languages", Institute of Electrical and Electronics Engineers (IEEE), Print ISSN: 0018-9162, Volume:33, Issue: 10, Doi:10.1109/2.876288, October 2000.
- [13] Douglas Blank, Lisa Meeden, Deepak Kumar, "Python robotics: an environment for exploring robotics beyond LEGOs", SIDCSE '03 Proceeding of the 34th SIGCSE technical symposium on Computer science education, ISBN: 1-58113-648-X Order Number: 457030, 1902-2003.
- [14] Michael H. Goldwasser, David Letscher, "Teaching an object-oriented CS1-with Python", ITiCSE Innovation and Technology in Computer Science Education, ISBN: 978 -1 - a. 60558-078-4, Order number: 457083, doi: 10.1145/1384271.1384285, 30-06-2008.
- [15] M. Sheeran, S Singh, "Ruby as a basis for hardware/ software code-sign", Institute of Electrical and Electronics Engineers (IEEE), INSPEC Accession Number: 5184000, Doi: 10.1049/ic:19951042, 06 August-2002.
- [16] Scott Chamberlain and Carl Boettiger, "R Python, and Ruby clients for GBIF species occurrence data", PeerJ Preprints, <https://doi.org/10.7287/peerj.preprints.3304v1>, CC BY 4.0 Open Access, rec: 29 Sep 2017, publ: 29 Sep 2017.
- [17] Julio Vega and Jose M. Canas, article in "PiBot: An Open Low-Cost Robotic Platform with Camera for STEM Education", applied sciences, Electronics 2018, 7, 430; doi:10.3390/electronics7120430.
- [18] Ning Yu and Kyle Darling, article in "A Low-Cost Approach to Crack Python CAPTCHAs Using AI-Based Chosen-Plaintext Attack", Appl. Sci. 2019, 9, 2010; doi:10.3390/app9102010.
- [19] Tom Radcliffe, article in "Python vs. Ruby: Which is best for web development", <https://opensource.com/article/17/4/python-or-ruby-web-development>, 11-April-2017.
- [20] Jeffrey Elkner, Allen B. Downey and Chris Meyers, "The way of program", <http://www.openbookproject.net>
- [21] <http://thinkcs.python/english2e/ch01.html>, 21 April 2012
- [22] <https://github.com/ga-wdi-boston/python-vs-ruby-study>
- [23] <https://jaydevs.com/ruby-vs-python/>
- [24] <https://www.ijeat.org/wp-content/uploads/papers/v9i3/C6435029320.pdf>
- [25] <https://hackr.io/blog/ruby-vs-python>
- [26] <http://old.hiit.fi/files/admin/publications/TechnicalReports/hiit-tr-2007-2.pdf>
- [27] <https://www.academia.edu/download/93350695/C6435029320.pdf>
- [28] <https://www.academia.edu/download/58163916/117.pdf>
- [29] <http://rie-eto.topaz.web.id/IT/1922-1815/Ruby3881rie-eto-topazngetesumum.html>
- [30] <https://folk.ntnu.no/haugwarb/Programming/Ruby/rubyvspython.pdf>