**Golang VS Rust**

***Golang***

Golang is an open source programming language that Google developed.

Golang is best for the back end side of a web application or website.

**Why is Go best for backend programming?**

* **Go has built-in concurrency called “goroutines”**. This means it can **multitask**, which is important for implementing high-performance servers and networked applications.
* **Networking**. Go has a robust standard library that offers support for plenty of common backend networking tasks, like HTTP handling and TLS (Transport Layer Security) encryption.
* **Go goes fast.**Go is famous for its **rapid compilation times and efficient runtime performance**, which is great for back end applications that need to handle a high volume of requests.

**Can Go be used on the front end?**

One way to use Go on the front-end is by compiling it to **WebAssembly** (Wasm). WebAssembly is a binary format that is supported by modern web browsers, and it allows developers to run code in the browser that was written in languages other than JavaScript.

### ***Rust***

Rust is a very low-level/systems-level programming language like C++ or C. It focuses on safety, speed, and concurrency

**Comparison Table:**

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Golang** | **Rust** |
| Defination | Its a open source statically typed compiled language. | Rust is a multi paradigm general purpose and open-source systems programming language . |
| Developed By and Year | Go language has been developed by Robert Griesemer, Rob Pike, and Ken Thompson at Google. It was introduced back in 2007 by Google and first launched in 2009 | developed by Graydon Hoare at Mozilla research and it came into existence back in 2010. |
| Focus | Stronger focus on **building API's** which is scale endlessly. It comes with goroutines, strong security and some standard libraries | Works well for processing , large volume of data. It's focused on safety, stability, performance and safe concurrency |
| Performance | Lacks in runtime speed | Rust has the upper hand in delivering output over Go due to its **great runtime speed.** |
| Compilation Time | It's better in Compilation time | Lacks in Compilation time |
| Concurrency | Go uses goroutines to utilize CPU's efficiency and handle **multiple tasks simultaneously.** These are light weight, **Call fewer resources** and are executed independently. They run concurrently with other functions and eliminated need to create threads | Rust was recently introduced with the native syntax in the form of async/await and guarantees memory stability. |
| Memory management and Security | Go offers various paradigms such as memory allocation and **automatic garbage collection**. It automated during runtime and don’t bother the developer to assign and remove memory | Rust ensures **no memory leakage during compile time**. wheres it makes developers needs to be attentive and be aware of memory allocation else code wont be compiled |
| Learning curve | Go is relatively **easier** than rust, it has steeper learning curve than other language like java script and python | Simplest syntax than Go and easy alternative to C++, But little hard to learn compare with Go. |
| Scalability | Go is a perfect programming language to develop **large scale applications** and handle large code base and teams | Rust favor applications that requires speed such as **game development** |
| Companies | American Express, Uber , Paypal | Dropbox, Cloud flare and Figma |

Both Rust and Golang can spell benefits galore in designing a **web service, which** *requires dealing with a high volume of traffic*. However, as per most of the modern developers, **Golang is a lot faster than Rust**. It is slightly better than the latter especially if you are working with a large team of developers or have an array of services to write.

**When to choose Golang?**

Golang is the perfect choice when you want to write code faster. Go is the best fit when you focus on simplicity and safety is your most priority. Go work great for micro services and for typical “DevOps” tasks.

**When to choose Rust?**

Rust is second to none for certain projects in which *high performance and concurrency are required*. You can choose Rust when you need total control over the hardware.

**Is Golang faster than Rust?**

Yes, the best asset of Golang is its incredible speed at which it compiles to machine code. So, Golang is faster than Rust.

Reference Links:-

<https://www.geeksforgeeks.org/difference-between-golang-and-rust/>

<https://www.mindinventory.com/blog/go-vs-rust-which-one-to-choose/>

<https://www.simplilearn.com/tutorials/golang-tutorial/guide-to-golang-vs-rust#:~:text=Go%20gets%20automated%20during%20the,be%20aware%20of%20memory%20allocation.>