RUST PROGRAMMING LANGUAGE

VIJAYESVAR.BA

CB.EN.U4CYS22074

RUST PROGRAMMING LANGUAGE

A language empowering everyone to build reliable and efficient software.Rust is a multi-paradigm system programming language focused on safety, especially safe concurrency. Rust is syntactically similar to C++, but is designed to provide better memory safety while maintaining high performance

HISTORY OF RUST

The language grew out of a personal project begun in 2006 by Mozilla employee Graydon Hoare.Project was possibly named after the rust family of fungi. Mozilla began sponsoring the project in 2009, and announced it in 2010. Rust 1.0, the first stable release, was released on May 15,2015.

GOALS OF RUST

The three goals on which Rust focuses include

- Speed
- Concurrency
- Safety

These goals are maintained through the eradication of garbage collector, which makes this language useful for a number of cases.

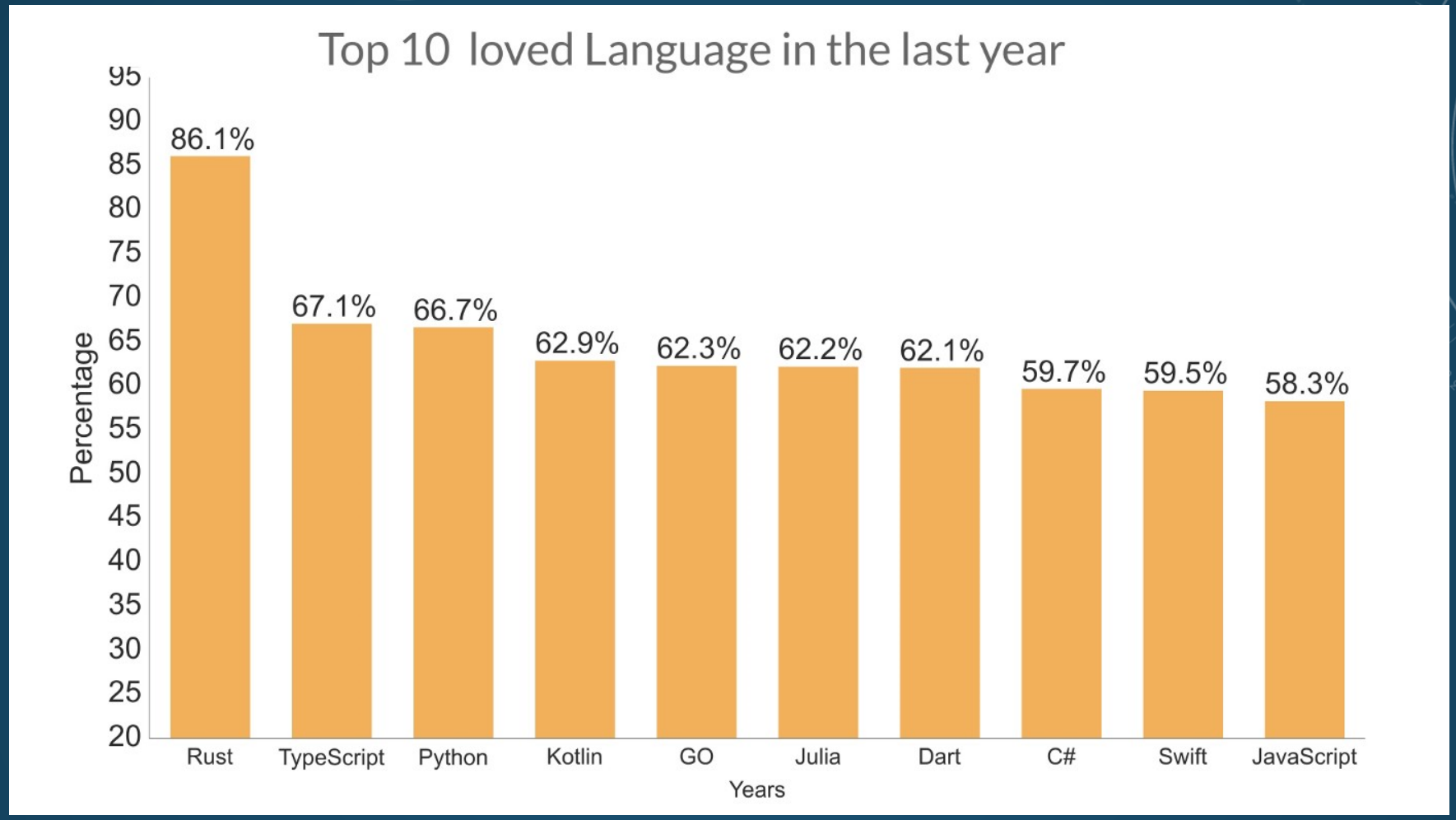
SYNTAX OF RUST

The concrete syntax of Rust is similar to C and C++, with blocks of code delimited by curly brackets, and control flow keywords such as if, else, while, and for. Not all C or C++ keywords are implemented, however, and some Rust functions (such as the use of the keyword match for pattern matching) will be less familiar to those versed in these languages.

SIMPLE PROGRAM IN RUST

```
fnmain(){
println("Hello, world!");
}
```

- Above code can be save in file.rs, but it can be any namewith .rs extension.
- Compiling via rustc file.rs
- Executing by ./file on Linux and Mac or file.exe on Windows



In programming, Rust is one of the most loved languages. It is a programming language focused on memory safety, protection, high performance during the processing of large amounts of data, concurrency and a highly efficient compiler

WHY RUST?

Performance:

Rust is blazingly fast and memory-efficient with no runtime or garbage collector, it can power performance-critical services, run on embedded devices, and easily integrate with other languages.

Reliability:

Rust's rich type system and ownership model guarantee memory-safety and thread-safety--and enable you to eliminate many classes of bugs at compile-time.

Productivity:

Rust has great documentation, a friendly compiler with useful error messages, and top-notch tooling an integrated package manager and build tool, smart multi- editor support with auto-completion and type inspections, an auto-formatter, and more.

WHAT MAKES RUST DIFFERENT?

- Unlike other programming languages, Rust was built from scratch, and it incorporates elements and findings from high-level systems programming languages and modern programming language design.
- Also, as an open source project, Rust has developers who have optimized and smoothed out the user experience. For example, the compiler generates very accurate and particularly instructive error messages to help the users; it even highlights offending code and also recommends how to fix it.
- The Rust community also provides thorough documentation and immediate online help. Not just the tooling but infrastructure around Rust is commendable. Unlike many other programming languages, where you need to use many compilers, Rust can be used with ease by having a single tool, named Cargo, to compile the code, run tests, generate documentation, and upload a package to a repository and much more.
- Rust has a powerful command-line interface tool; it's an excellent tool for developing web applications and also cross-platform applications. Rust is profoundly suitable for low-resource environments. For popular development environments, there are Rust plugins, such as Visual Studio Code by Microsoft, IntelliJ by JetBrains, and Rust Playground, and it provides a rich live online environment of Rust for code experimentation.

more control,
less safety

less control,
more safety

Rust

more control,
more safety

HOW RUST COMPARES TO OTHER PROGRAMMING LANGUAGES

Developers love Rust," the latest Stack Overflow survey, released last month, proclaimed.

The Mozilla Foundation, known for the popular Firefox web browser and Thunderbird email clients, has also been working on the open source Rust programming language for a while now. Many developers have been watching Rust with anticipation, waiting for the right time to dive in and start building things. Given that Rust hit 1.0 last May, it is now better than ever to start hacking on Rust!

As with any language or tool, though, it's important to understand what problems Rust was introduced tosolve, and what types of solutions best flow from it.

MORE LIKE C++ AND GO, LESS LIKE NODE AND RUBY

While Rust is a general purpose language, you could write your next web app in Rust, but you wouldn't be best experiencing what it has to offer. Rust is a low-level language, best suited for systems, embedded. and other performance critical code. While it is conceivable that one day people will be writing the latest 3D video games in Rust-an area where high performance has historically been critical-it is unlikely ever to have a web framework that will go toe-to-toe with Ruby on Rails.

SAFER THAN C/C++:

- The biggest--and some would say most vital-difference between Rust and C++ is the emphasis writing safe code.onWith "safe code," objects are managed by the programming language from the beginning to end.
- developer doesn't do any pointer arithmetic or manage memory, as can be necessary in C or C++ programs. For a given object the proper amount of memory is promised to be allocated-or, reserved-for the object. When accessing this object, it is impossible to accidentally access a memory location that is out of bounds. And when its job is done, the object will automatically be deallocated by the system, by which 1mean the programmer will not have to manually "free" or unreserve the memory used by that object.
- With unmanaged code, not only is it harder to write code that is correct and bug-free, but leaves code far more vulnerable to security threats. A particularly common threat is a buffer overflow, where a user can enter more information than can be contained within the program's allotted memory space, allowing a malicious user to modify memory in parts of the system not under direct control of the code

SOME SHINING RUST STATS

- When it comes to popularity, Rust is on top. Since 2015, Rust has been voted the most loved programming language by developers on Stack Overflow's developer survey for four consecutive years (2016, 2017, 2018, 2019):
- Rust is at the top of the top 10 programming languages to earn the highest salaries for 2020:
- On Google trends, you can see the way Rust's search results are growing year by year:According to the survey by Stack Overflow, Rust is on the top five list of most wanted top programming languages:
- Rust broke into TIOBE's index of the top 20 most popular programming languages for the first time in 2020. And in the 2020's Sub-Reddit survey (r/adventofcode), Rust was positioned as the second most used programming language after Python

FRAMEWORK:

- Rocket is a web framework for Rust that makes it simple to write fast, secure web applications without sacrificing flexibility, usability, or type safety.
- Rust in production panies around the world are using Rust in production today for fast, low-resource, cross-platform solutions. Software you know and love, like Firefox, Dropbox, and Cloudflare, uses Rust. From startups to large corporations, from embedded devices to scalable web services, Rust is a great fit.

THANK YOU!