JavaScript Runtime

Section 1

In this project, we explored the creation of a basic JavaScript runtime using Rust. To replicate JavaScript runtimes like Node.js and Deno. We learned how to create a basic javascript runtime from scratch.

A JavaScript runtime environment is a software component that executes JavaScript programs. It provides the necessary infrastructure for running JavaScript code, including the engine, memory management, and other supporting services.

Here's a detailed explanation of what a JavaScript runtime is:

### **Understanding JavaScript Runtimes**

Before we delve into creating our own runtime, it's crucial to understand what a JavaScript runtime is and why it's important.

**JavaScript Runtime**: A JavaScript runtime environment provides the necessary infrastructure to execute JavaScript code outside of a web browser. It includes a JavaScript engine, which compiles and executes JavaScript code, and additional features for interacting with the system, such as file system access and networking.

* **JavaScript Engine**: Executes JavaScript code by converting it into machine code. For example, Google Chrome uses the V8 engine.
* **Runtime Environment**: Provides additional capabilities like I/O operations and system interaction. Node.js is a popular JavaScript runtime that uses the V8 engine and adds these extra features.

**Browsers vs. Runtimes**: Browsers offer APIs for frontend development, such as document.getElementById, whereas JavaScript runtimes like Node.js provide APIs for backend operations, such as file system access and network requests.

**Reasons to Learn About Runtimes**

* Performance Optimization: Understanding how a runtime works allows developers to optimize their code for better performance.Knowledge of runtime behavior helps in identifying bottlenecks and optimizing resource usage.
* Platform Independence: Familiarity with runtimes enables developers to create cross-platform applications more easily.It helps in understanding how code behaves differently across various environments.
* Debugging and Troubleshooting:Knowing how a runtime operates aids in diagnosing issues and debugging complex problems.It helps in understanding unexpected behaviors and edge cases.
* Advanced Features Utilization: Understanding runtimes opens up opportunities to utilize advanced features and optimizations.It allows developers to take advantage of runtime-specific capabilities.
* Security Awareness: Knowledge of runtime behavior enhances security awareness, helping identify potential vulnerabilities.It aids in understanding how to prevent common security issues.
* Cloud and Distributed Systems: Understanding runtimes is crucial for developing cloud-native and distributed applications.It helps in managing resources efficiently in containerized environments.
* Multithreading and Parallelism: Knowledge of runtimes facilitates effective use of multi-core processors and parallel computing.It aids in designing concurrent and scalable systems.
* Memory Management:Understanding runtime memory management is essential for efficient resource utilization.It helps in avoiding memory leaks and optimizing memory usage.

**Core Components of a JavaScript Runtime**

* Engine: The core of the runtime, responsible for parsing and executing JavaScript code 3.
* Memory Management: Handles allocation and deallocation of memory for variables and objects 3.
* Type System: Manages data types and type conversions 3.
* Object Model: Defines how objects are structured and interact 3.
* DOM (Document Object Model): Provides an interface for interacting with HTML elements 3.

### **Getting Started with the Project**

### **1. Install Rust and Cargo:**

### **What are they?**

* Rust is a powerful programming language known for its speed, memory safety, and concurrency features.
* Cargo is Rust's package manager, used to download and manage dependencies for your projects.

### 

### **How to install:**

* Visit the official Rust installation page:<https://www.rust-lang.org/learn/get-started>
* Follow the instructions for your operating system (Windows, macOS, or Linux).
* This should install both Rust and Cargo automatically.

### **Verification:**

* Open your terminal or command prompt.
* Type cargo --version and press Enter.
* If you see a version number (e.g., cargo 1.64.0), the installation was successful.

### **2. Create a New Rust Project:**

### Rust projects use Cargo for managing dependencies and building the final program.

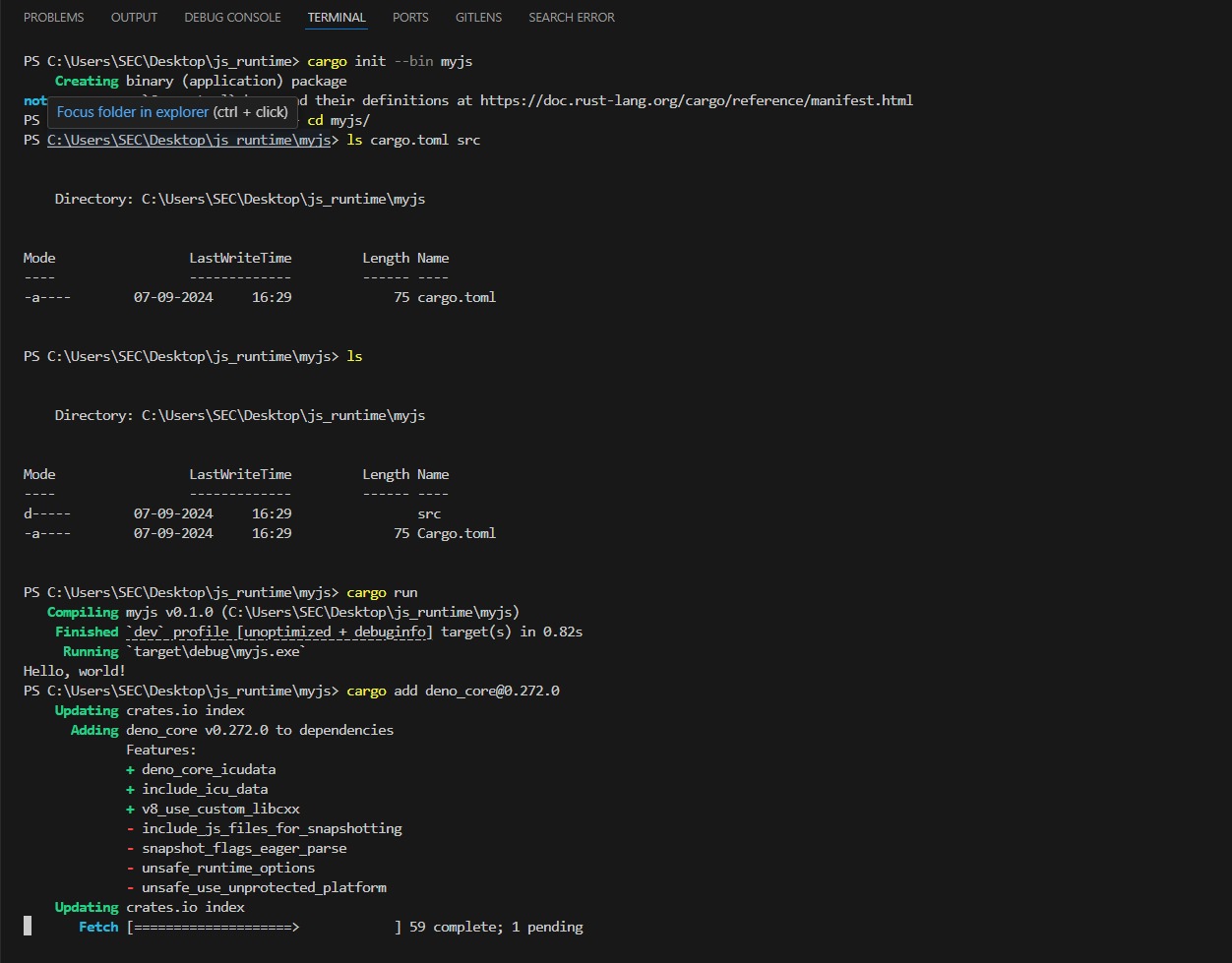
### **Steps:**

### Open your terminal/command prompt.

### Navigate to the directory where you want your project to reside.

### Run the command cargo new my\_project\_name. Replace "my\_project\_name" with your desired name.

### This creates a new directory with the chosen name and essential files like Cargo.toml (project configuration) and src/main.rs (the main Rust source file).



### **3. Add Dependencies:**

### Our project interacts with JavaScript code, so we need the deno\_core crate.

### Crates are Rust's term for reusable libraries.

### **Steps:**

### Open Cargo.toml in your code editor.

### Locate the [dependencies] section (or add it if it doesn't exist).

### Add a line like this, replacing the version number if necessary:

### **4. Modify the Project**

### We'll edit the main.rs file to incorporate the deno\_core crate and our custom logic.

### **Steps:**

### Open src/main.rs in your code editor.

### **5. Import Necessary Functions:**

### **What are imports?**

### Imports allow your code to use functions and data structures defined in other crates.

### 

### **6. Set Up the JavaScript Runtime and run\_js Function:**

### **JavaScript Runtime Creation:**

### The JsRuntime object represents the JavaScript runtime environment.

### **run\_js Function:**

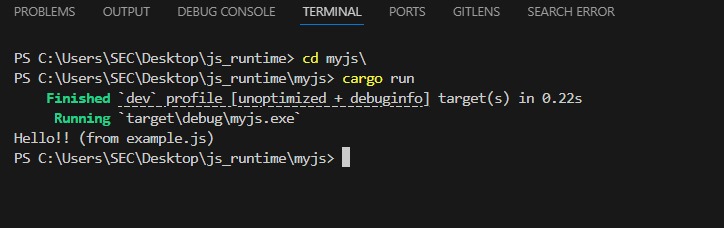
### This function will be responsible for executing JavaScript code using the runtime.

### **Explanation:**

* The main function is the entry point of our program.
* We create a new JsRuntime instance with default options.
* The run\_js function is asynchronous (indicated by async), meaning it might involve waiting for operations to complete. We'll fill in the details of this function later.
* Inside main, we call run\_js with the path to your JavaScript file (replace "your\_script.js" with the actual path).

### **7. Create a Tokio Runtime (Optional):**

### This step is optional but might be necessary if your JavaScript code involves asynchronous operations. Tokio is a popular library for asynchronous programming in Rust.



### 