

Language Server と LSP

**VS Code にエラーが
表示されるとき、
裏で何が起きているのか**

キーワード

1. Language Server

言語機能を提供するサーバー

2. LSP

Language Server Protocol

**まずは
キーワード解説です**

Language Server ってなに？

プログラミング言語のサポート機能を提供してくれる**サーバー**です。

- エラーや警告の検出
- 定義へのジャンプ
- 関数名などのリネーム

LSP ってなに？

IDE（VS Code）と Language Server がやり取りする際のプロトコルです。

IDE（クライアント）

VS Code / Cursor / Vim …

LSP



Language Server

TypeScript / Go / Rust …

LSP があることで、様々な IDE と Language Server が
同じようにやりとりできます。

好きな IDE で好きな言語を書けちゃいます♥

やり取りの形式は JSON-RPC

JSON-RPC という形式で、メッセージを送り合います。
たとえば「この位置にカーソルを合わせた」——

```
{  
  "jsonrpc": "2.0",  
  "method": "textDocument/hover",  
  "params": { "position": { "line": 10, "character": 4 } }  
}
```

※ JSON が読めるならたぶん読めます。

やり取りの流れ



※ LSP という規約に準拠して IDE と Language Server がやりとりします。

VS Code の「言語」拡張機能

大まかな中身は、次の 4 つです。

1. その言語を使うための設定
2. コードに色をつける仕組み
3. **Language Server**
4. **Language Client**

補足：ファイルを開くと拡張機能が Language Server を起動し、VS Code との通信を始めます。

ここから
デモを使います

デモ①

VS Code（Cursor）にエラーが出るときの
Language Server の出力を見えます 🙌

エラーが表示されるまで

タイトル回収です

VS Code

1. コードを変更する

↓ `textDocument/didChange`

Language Server

2. 変更を受け取り、コードを解析する

↓ `textDocument/publishDiagnostics`

VS Code

3. エラーを表示する

おまけ

デモ②

次は、オリジナルの Language Server を
お披露目します 🙌

1. 独自の警告

2. 独自の Quick Fix

※ 実装も見せちゃいます (。>_<。)

LSP で広がる世界

ぜひ LSP を絡めたツール開発をしてみてください \(^o^)/

発想次第で、開発を思いっきりブーストできるかもしれません。