

Tree-sitter is awesome and you should use it

(#eq? @author “Kian Kasad”)

Grab these slides!

<https://kasad.com/tree-sitter.pdf>

^[\t\t]ree[-_]?sitter\$

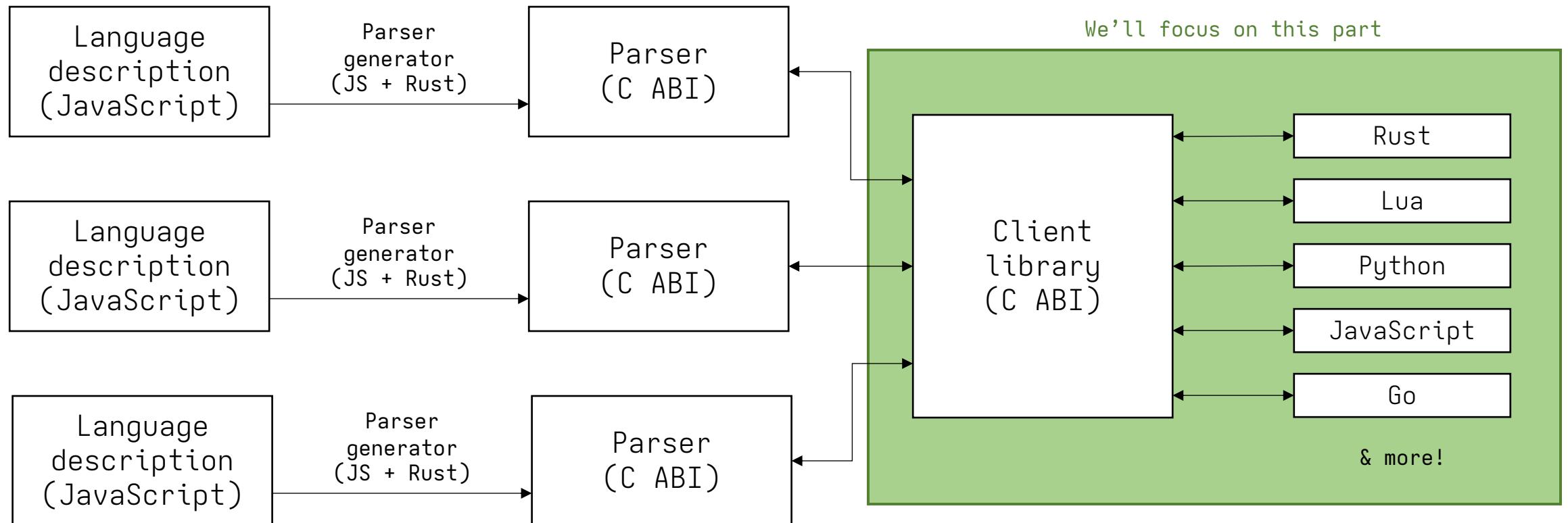
What is it?

*Tree-sitter is a **parser generator** tool and an **incremental parsing** library. It can build a **concrete** syntax tree for a source file and efficiently update the syntax tree as the source file is edited. Tree-sitter aims to be:*

- **General** enough to parse any programming language
- **Fast** enough to parse on every keystroke in a text editor
- **Robust** enough to provide useful results even in the presence of syntax errors
- **Dependency-free** so that the runtime library (which is written in pure C11) can be embedded in any application

— <https://tree-sitter.github.io/tree-sitter/index.html>

Architecture



Isn't this just another parser?

Yes, but... Tree-sitter gives us **queries**!

They look like this:

(call_expression)

Isn't this just another parser?

They look like this:

(call_expression)

And match this:

```
char *a = (*func_ptr)(get_input_stream());  
int cmp = __builtin_memcmp(a, &b, 32);
```

Isn't this just another parser?

They look like this:

```
(call_expression  
  function: (identifier))
```

And match this:

```
char *a = (*func_ptr)(get_input_stream());  
int cmp = __builtin_memcmp(a, &b, 32);
```

Isn't this just another parser?

They look like this:

```
(call_expression  
  function: (identifier) @function)
```

And match this:

```
char *a = (*func_ptr)(get_input_stream());  
int cmp = __builtin_memcmp(a, &b, 32);
```

Isn't this just another parser?

They look like this:

```
((call_expression
  function: (identifier) @function.builtin)
 (#match? @function.builtin "^__builtin_"))
```

And match this:

```
char *a = (*func_ptr)(get_input_stream());
int cmp = __builtin_memcmp(a, &b, 32);
```

Isn't this just another parser?

They look like this:

```
(init_declarator
  declarator: (identifier) @lhs
  value: ((call_expression
    function: (identifier) @function.builtin)
  (#match? @function.builtin "^_builtin_")))
```

And match this:

```
char *a = (*func_ptr)(get_input_stream());
int cmp = _builtin_memcmp(a, &b, 32);
@lhs           @function.builtin
```

What can we do with this?

Tree-sitter gives you:

- Better syntax highlighting

Demo: Better syntax highlighting

Turn off LSP

:LspStop

Toggle tree-sitter

:TSToggle highlight

What can we do with this?

Tree-sitter gives you:

- Better syntax highlighting
- Smart editing features

Demo: Smart editing features

Using these plugins:

- [nvim-treesitter/nvim-treesitter](#)
- [nvim-treesitter/nvim-treesitter-textobjects](#)
- [echasnovski/mini.ai](#)

What can we do with this?

Tree-sitter gives you:

- Better syntax highlighting
- Smart editing features
- Scriptable syntax-aware transformations

Demo: Syntax-aware scripting

Problem:

I have a lot of long C files for which I need to generate header files.

Solution: Tree-sitter!

1. Select each function definition.
2. Strip off the body.
3. Fold into one line and add a semicolon.

What can we do with this?

Tree-sitter gives you:

- Better syntax highlighting
- Smart editing features
- Scriptable syntax-aware transformations
- The ability to rewrite [eastwood-tidy](#) in Rust

Demo: Custom language injections

Query definitions

are how Neovim uses treesitter's parse tree to define highlights, text objects, injections, and more.

Language injections

are a feature of Neovim that lets us switch languages in the middle of a file.

Questions?

**(MISSING
(slide)) @end**