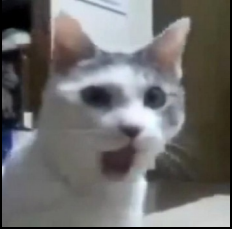




jenseits von q!





Michael Weimann

- Macht (fast alles) mit NeoVim
- Vim im fortgeschrittenen Stadium

@weeman:ccc-ffm.de
weeman@ccc-ffm.de



Vim =

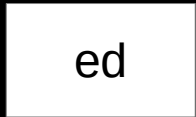




Fahrplan

- Geschichtsstunde
- NeoVim
 - Standard-Features
 - Plugins (Auswahl)
- Tipps für den Start
- Fragen

Geschichtsstunde



Zeilenbasierter
Editor

1969 -55 Jahre

ed

schlichte
Begrüßung

```
[nix-shell:~/git/krempel/gpn24]$ ed test.txt
```

```
66
```



ed

1p

erste Zeile
ausgeben

```
[nix-shell:~/git/krempel/gpn24]$ ed test.txt
66
1p
Das ist eine Zeile
2p
und noch eine Zeile
█
```

ed

a

tippen

.

dann nix

```
[nix-shell:~/git/krempel/gpn24]$ ed test.txt
```

```
66
```

```
1p
```

```
Das ist eine Zeile
```

```
2p
```

```
und noch eine Zeile
```

```
a
```

```
etwas nach der zweiten Zeile
```

```
.
```



```
I
```




Damals™

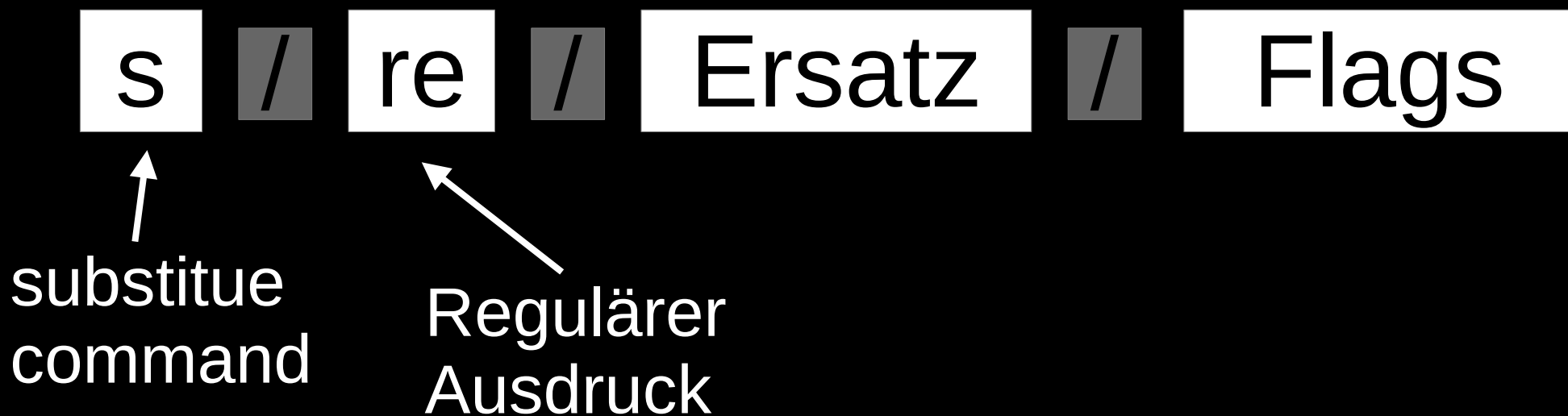
- Bildschirme waren Luxus
- Ausgabe auf Papier

Fernschreiber

EN: Teletypewriter (kurz TTY)



ed – Zeilen bearbeiten



MENSCH...

DAS IST JA PRAKTISCH

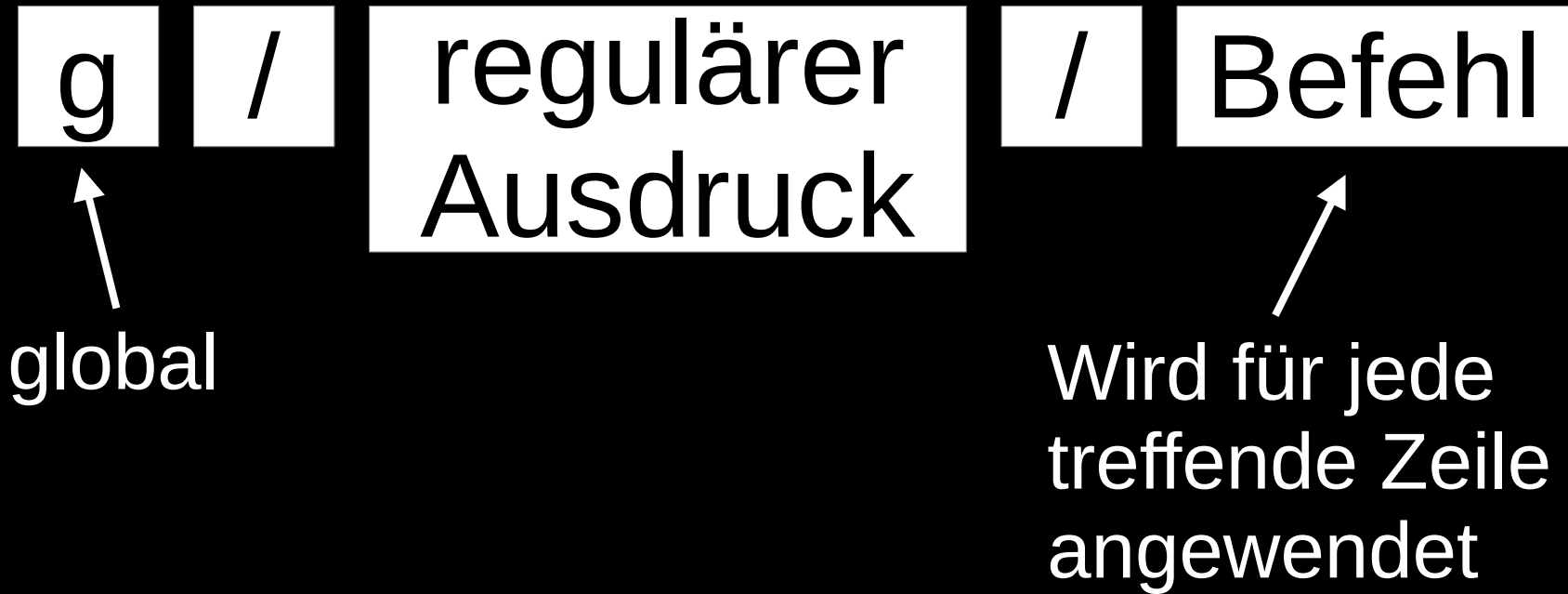
imgflip.com

/

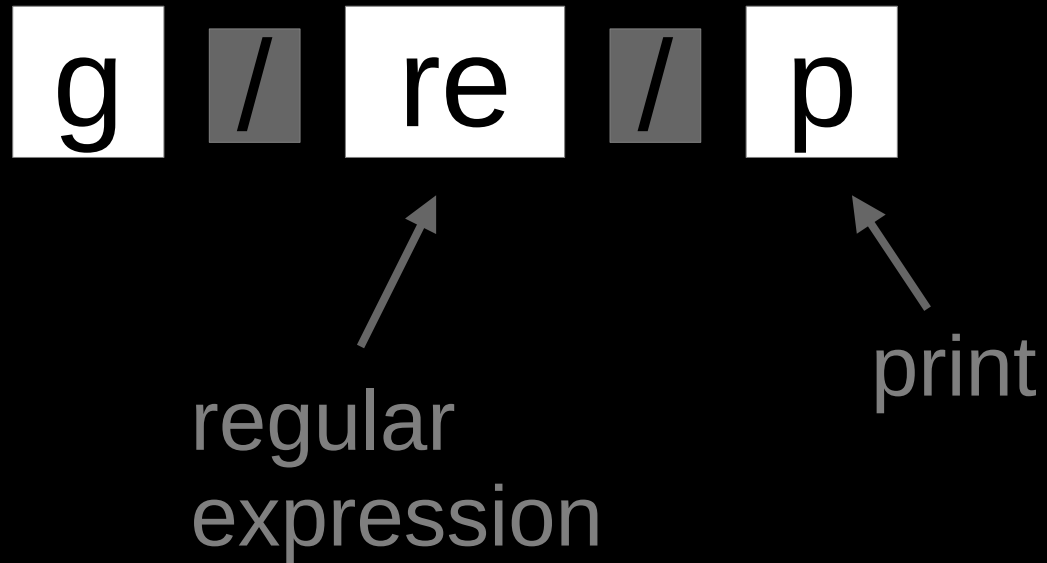
Flags

→ sed

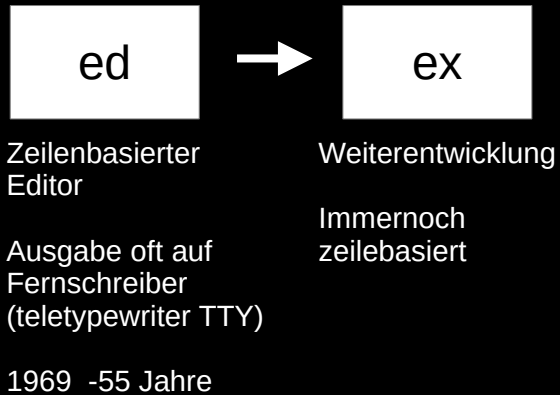
ed – Global Command



ed – Zeilen finden



Geschichtsstunde



ex

- ed mit Doppelpunkt

```
[nix-shell:~/git/krempel/gpn24]$ ex test.txt
```

I

```
test.txt: unmodified: line 3
```

```
: |
```


ex

:q!

```
test.txt: unmodified: line 3
```

```
:3p
```

```
alle guten Dinge sind drei
```

```
:a
```

```
oder vier!
```

```
.
```

```
:q
```

```
File modified since last complete write; write or use  
! to override
```

```
:q!
```

Die Zukunft™

- Computer Terminals mit Bildschirm!

➔ Zeilen bearbeiten nervt!





ex

:vi

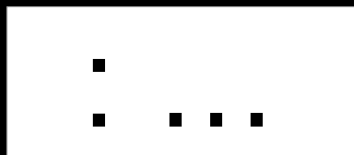
hjkl

cc

```
Das ist eine Zeile  
und noch eine Zeile  
a|lle guten Dinge sind drei
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

ex



- „ex“
cmds

```
Das ist eine Zeile
und noch eine Zeile
alle guten Dinge sind drei
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
:s/und/hier
```

```
I
```

vi = ex -v

Geschichtsstunde

ed

Zeilenbasierter
Editor

Ausgabe oft auf
Fernschreiber
(teletypewriter TTY)

1969 -55 Jahre



ex

Weiterentwicklung

Immernoch
zeilebasiert

Später mit :vi Mode



vi

Visueller Editor

Basiert auf ex vi
Mode

kann :ex commands



Vim – vi improved (1991)

- Syntax Highlighting
- Fenster
- Folding
- VimScript
- Plugins
- Diff mode
- ...

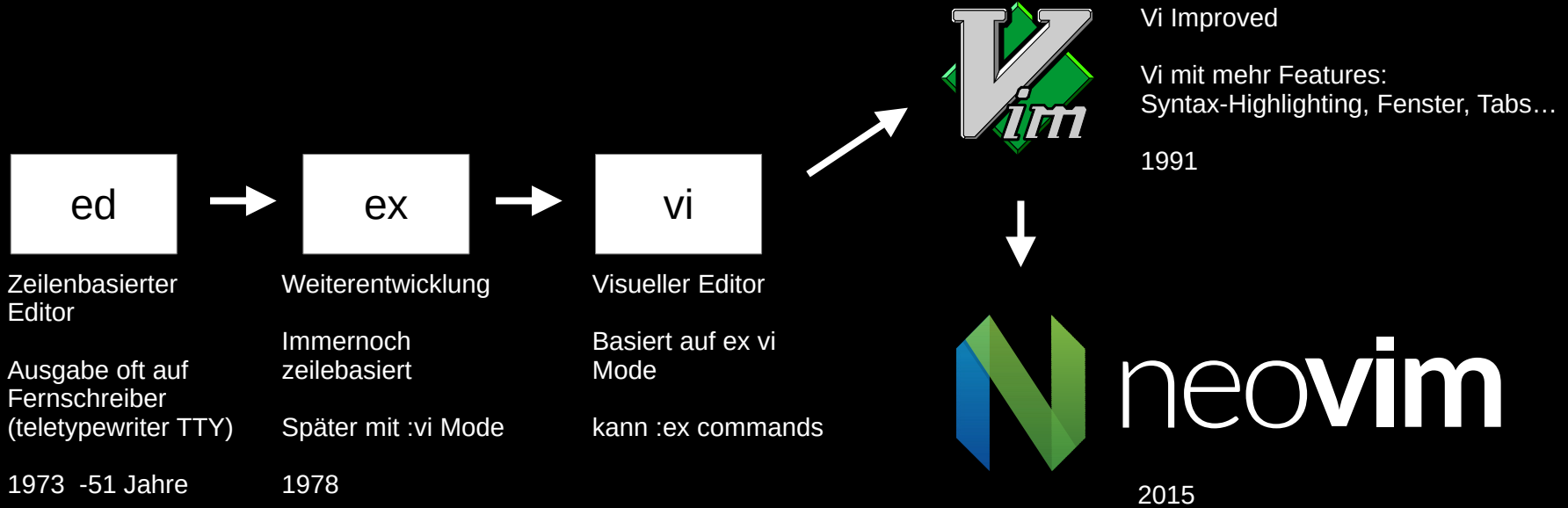


Vim – not improved

- One man show
- Durchwachsene Dokumentation
- Wartbarkeitsprobleme
- Stagnation
 - (z.B. kein async i/o)



Geschichtsstunde





- Vim Fork 2015
- Entwickelnden-Community
- Wartbarkeit und Dokumentation verbessern
- Größere Umbauten



Features

Language Server Protocol

```
public async create(): Promise<GroupCall> {
  this.creationTs = Date.now();
  this.client.groupCallEventHandler!.groupCalls.set(this.room.roomId, this);
  this.client.emit(GroupCallEventHandlerEvent.Outgoing, this);
  this.client.emit
}

emit Method (method)
emitPromised Method TypedEventEmitter<EmittedEvents,
reEmitter Property ClientEventHandlerMap,
requestEmailToken Method ClientEventHandlerMap>.
requestPasswordEmailToken Method emit<T>(event: T, ...args:
requestRegisterEmailToken Method Parameters<ClientEventHandlerMap[T]>
requestAdd3pidEmailToken Method : boolean (+1 overload)
getEventTimeline Method Synchronously calls each of the
paginateEventTimeline Method listeners registered for the event
getDomain Method named
setRoomTopic Method event, in the order they were
getMediaConfig Method registered, passing the supplied @@@
```

Language Server Protocol

Netbeans

- C
- PHP
- JavaScript

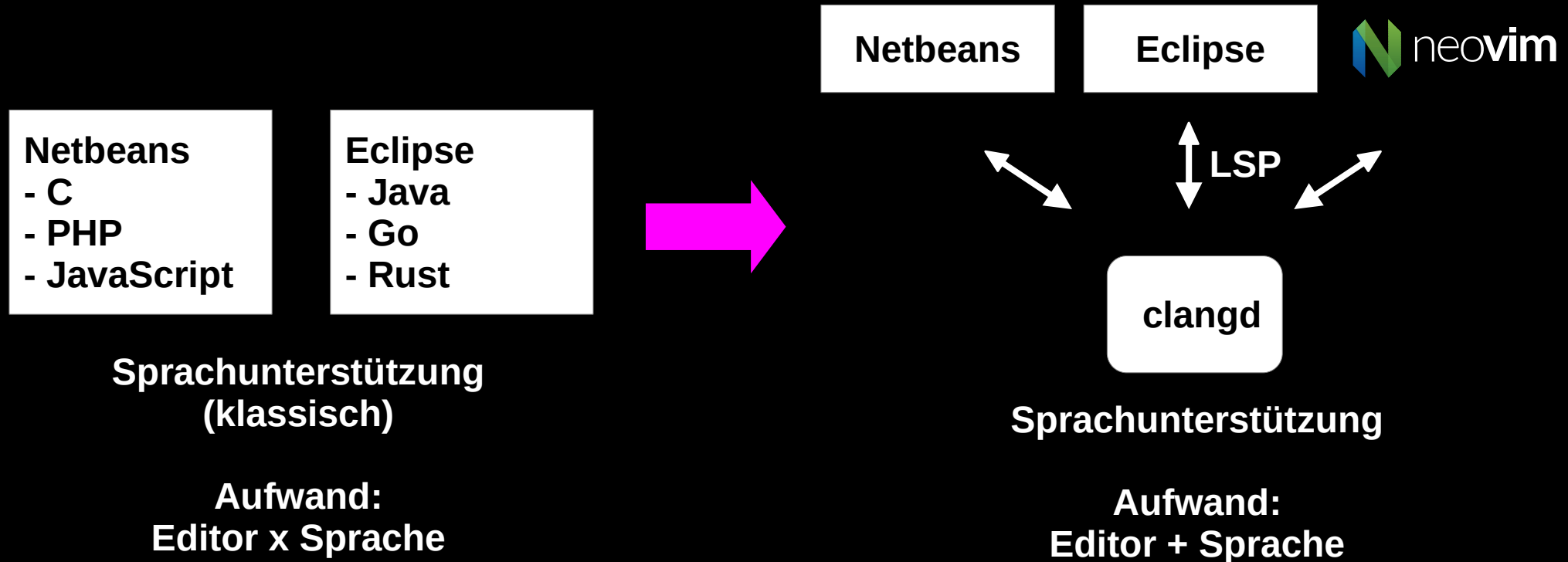
Eclipse

- Java
- Go
- Rust

**Sprachunterstützung
(klassisch)**

**Aufwand:
Editor x Sprache**

Language Server Protocol



NeoVim als LSP Client



textDocument.didChange

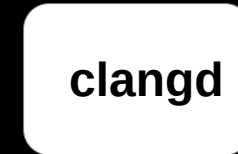


[Protokoll hier](#)

NeoVim als LSP Client



textDocument.didChange



LSP Server


textDocument.publishDiagnostics



[Protokoll hier](#)


LSP – Diagnostic

```
public async create(): Promise<GroupCall> {  
    this.creationTs = Date.now (); ■ Unexpected whitespace between function name and paren.  
    this.client.groupCallEventHandler!.groupCalls.set(this.room.roomId, this);  
    this.client.emit(GroupCallEventHandlerEvent.Outgoing, this);  
  
    await this.sendCallStateEvent();  
  
    return this;  
}
```



LSP – Code Actions

```
2
1   public async create(): Promise<GroupCall> {
E 349   this.creationTs = Date.now (); ■ Unexpected whitespace between function name and paren.
1     this.client.groupCallEventHandler!.groupCalls.set(this.room.roomId, this);
2     this.client.emit(GroupCallEventHandlerEvent.Outgoing, this);
3
4     await this.sendCallStateEvent();
5
6     return this;
7   }
```

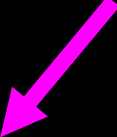
Code actions: 

- 1: Fix this @typescript-eslint/func-call-spacing problem
- 2: Disable @typescript-eslint/func-call-spacing for this line
- 3: Disable @typescript-eslint/func-call-spacing for the entire file
- 4: Show documentation for @typescript-eslint/func-call-spacing
- 5: Fix all auto-fixable problems
- 6: Move to a new file

Type number and <Enter> or click with the mouse (q or empty cancels):

LSP – Inlay Hints

```
public async create(): Promise<GroupCall> {  
    this.creationTs = Date.now();  
    this.client.groupCallEventHandler!.groupCalls.set(key: this.room.roomId, value: this);  
    this.client.emit(event: GroupCallEventHandlerEvent.Outgoing, call: this);  
  
    await this.sendCallStateEvent();  
  
    return this;  
}
```



LSP – Navigation

- Zur Deklaration
- Referenzen
- Implementierungen
- ...

LSP – Navigation

```
src/client.ts:8214:12
src/client.ts:7704:18
src/client.ts:6490:28
src/client.ts:6208:36
src/client.ts:6131:69
src/client.ts:6121:45
src/client.ts:6072:65
src/client.ts:6065:65
spec/unit/room.spec.ts:3057:25
spec/unit/room.spec.ts:2961:25
> spec/unit/room.spec.ts:2792:25
spec/integ/matrix-client-rela...
```

```
public fetchRelations(
  this.fetchRelations(roomId,
  promise = this.fetchRelation
  const res = await this.fetch
  const resNewer: IRel
  const resOlder = await t
  const resNewer: IRelatio
  const resOlder: IRelatio
  room.client.fetchRelations =
  room.client.fetchRelations =
  room.client.fetchRelations =
  const response = client!.fetchRe
```

LSP References

12 / 12

```
});
});
room.client.fetchRelations = (
  roomId: string,
  eventId: string,
  relationType?: RelationType | stri
  eventType?: EventType | string | n
  opts: IRelationsRequestOpts = { di
) =>
  Promise.resolve({
    chunk: [threadResponse.event]
    next_batch: "start_token",
  });
const prom = emitPromise(room, ThreadE
await room.addLiveEvents([randomMessag
const thread: Thread = await prom;
await emitPromise(room, ThreadEvent.Up
expect(thread.initialEventsFetched).to
```

Language Server



- Rust
- TypeScript
- C
- Markdown
- Cobol
- ...

Configurations

LSP configs provided by nvim-lspconfig are listed below. To view this file in Nvim by running `:help lspconfig-`

- [agda_ls](#)
- [aliken](#)
- [als](#)
- [anakin_language_server](#)
- [angularls](#)
- [ansiblels](#)
- [antlersls](#)
- [apex_ls](#)
- [arduino_language_server](#)
- [asm_lsp](#)
- [ast_grep](#)
- [astro](#)
- [autotools_ls](#)
- [awk_ls](#)
- [azure_pipelines_ls](#)
- [bacon_ls](#)

LSP

- Nicht nur Programmiersprachen

- Rechtschreibprüfung
(cspell)

```
console.log('Froscon')
```

■ Unknown word (Froscon)

- Linting
(z.B. ESLint, ruff...)

Tree-sitter

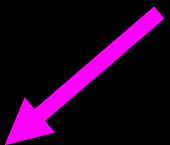
- Fehlertoleranter, schneller Parser
- Konkreter Syntaxbaum
- In NeoVim eingebaut
- `:InspectTree`




<https://tree-sitter.github.io/tree-sitter/>

Tree-sitter – :InspectTree

```
1 function add (a, b) {  
  1   return a + b  
  2 }
```



```
2 (program ; [0, 0] - [3, 0]  
  1 (function_declaration ; [0, 0] - [  
  2   name: (identifier) ; [0, 9] - [0,  
  3   parameters: (formal_parameters  
  4     (identifier) ; [0, 14] - [0, 1  
  5     (identifier)) ; [0, 17] - [0,  
  6   body: (statement_block ; [0, 20]  
  7     (return_statement ; [1, 2] - [  
  8       (binary_expression ; [1, 9]  
         left: (identifier) ; [1, 9]  
         right: (identifier)))))) ;
```



Tree-sitter - Syntax Highlighting

Regex

Tree
sitter

```
12 int tslua_add_language(lua_State *L)
11 {
10     if (lua_gettop(L) < 2 || !lua_isstring(L, 1) || !lua_isstring(L, 2)) {
9         return luaL_error(L, "string expected");
8     }
7
6     const char *path = lua_tostring(L, 1);
5     const char *lang_name = lua_tostring(L, 2);
4
3     if (pmap_has(cstr_t)(langs, lang_name)) {
2         return 0;
1     }

```

```
140
1 #define BUFSIZE 128
2 char symbol_buf[BUFSIZE];
3 snprintf(symbol_buf, BUFSIZE, "tree_sitter_%s", lang_name);
4 #undef BUFSIZE
5
6 uv_lib_t lib;
7 if (uv_dlopen(path, &lib)) {
8     snprintf((char *)IObuff, IOSIZE, "Failed to load parser: uv_dlopen: %s",
9             uv_dlerror(&lib));
10    uv_dlclose(&lib);
11    lua_pushstring(L, (char *)IObuff);
12    return lua_error(L);
13 }
14
15 TSLanguage *(*lang_parser)(void);
16 if (uv_dlsym(&lib, symbol_buf, (void **)&lang_parser)) {
17     snprintf((char *)IObuff, IOSIZE, "Failed to load parser: uv_dlsym: %s",
18             uv_dlerror(&lib));
19    uv_dlclose(&lib);
20    lua_pushstring(L, (char *)IObuff);
21    return lua_error(L);
22 }
23
24 TSLanguage *lang = lang_parser();
25 if (lang == NULL) {
26     return luaL_error(L, "Failed to load parser: internal error");
27 }
28
29 pmap_put(cstr_t)(langs, xstrdup(lang_name), lang);
30
31 lua_pushboolean(L, true);
32 return 1;
33 }
34
```

```
12 int tslua_add_language(lua_State *L)
11 {
10     if (lua_gettop(L) < 2 || !lua_isstring(L, 1) || !lua_isstring(L, 2)) {
9         return luaL_error(L, "string expected");
8     }
7
6     const char *path = lua_tostring(L, 1);
5     const char *lang_name = lua_tostring(L, 2);
4
3     if (pmap_has(cstr_t)(langs, lang_name)) {
2         return 0;
1     }

```

```
140
1 #define BUFSIZE 128
2 char symbol_buf[BUFSIZE];
3 snprintf(symbol_buf, BUFSIZE, "tree_sitter_%s", lang_name);
4 #undef BUFSIZE
5
6 uv_lib_t lib;
7 if (uv_dlopen(path, &lib)) {
8     snprintf((char *)IObuff, IOSIZE, "Failed to load parser: uv_dlopen: %s",
9             uv_dlerror(&lib));
10    uv_dlclose(&lib);
11    lua_pushstring(L, (char *)IObuff);
12    return lua_error(L);
13 }
14
15 TSLanguage *(*lang_parser)(void);
16 if (uv_dlsym(&lib, symbol_buf, (void **)&lang_parser)) {
17     snprintf((char *)IObuff, IOSIZE, "Failed to load parser: uv_dlsym: %s",
18             uv_dlerror(&lib));
19    uv_dlclose(&lib);
20    lua_pushstring(L, (char *)IObuff);
21    return lua_error(L);
22 }
23
24 TSLanguage *lang = lang_parser();
25 if (lang == NULL) {
26     return luaL_error(L, "Failed to load parser: internal error");
27 }
28
29 pmap_put(cstr_t)(langs, xstrdup(lang_name), lang);
30
31 lua_pushboolean(L, true);
32 return 1;
33 }
34
```

Tree-sitter - Text Objects

- Funktion

v

if

inner function

```
function add (a, b) {  
  // return the result of a + b  
  return a + b  
}
```

Tree-sitter Text Objects

- Blöcke { ... }
- Parameter (a, b)
- Kommentare //
- Klassen class { ...
- ...

Tree-sitter Module

- `nvim-treesitter`
 - Syntax Highlighting
 - Incremental Selection
 - Einrückung
 - Folding
 - Text Objects
 - ...

EditorConfig



- Seit 0.9 eingebaut
- Regeln für Zeichensatz, Tabs vs. Spaces
- Von vielen Editoren unterstützt
- <https://editorconfig.org/>

EditorConfig



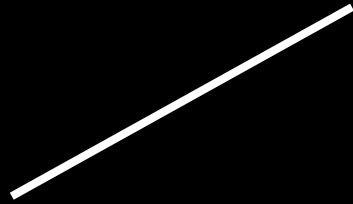
```
root = true
```

```
[*]  
end_of_line = lf  
insert_final_newline = true  
charset = utf-8
```

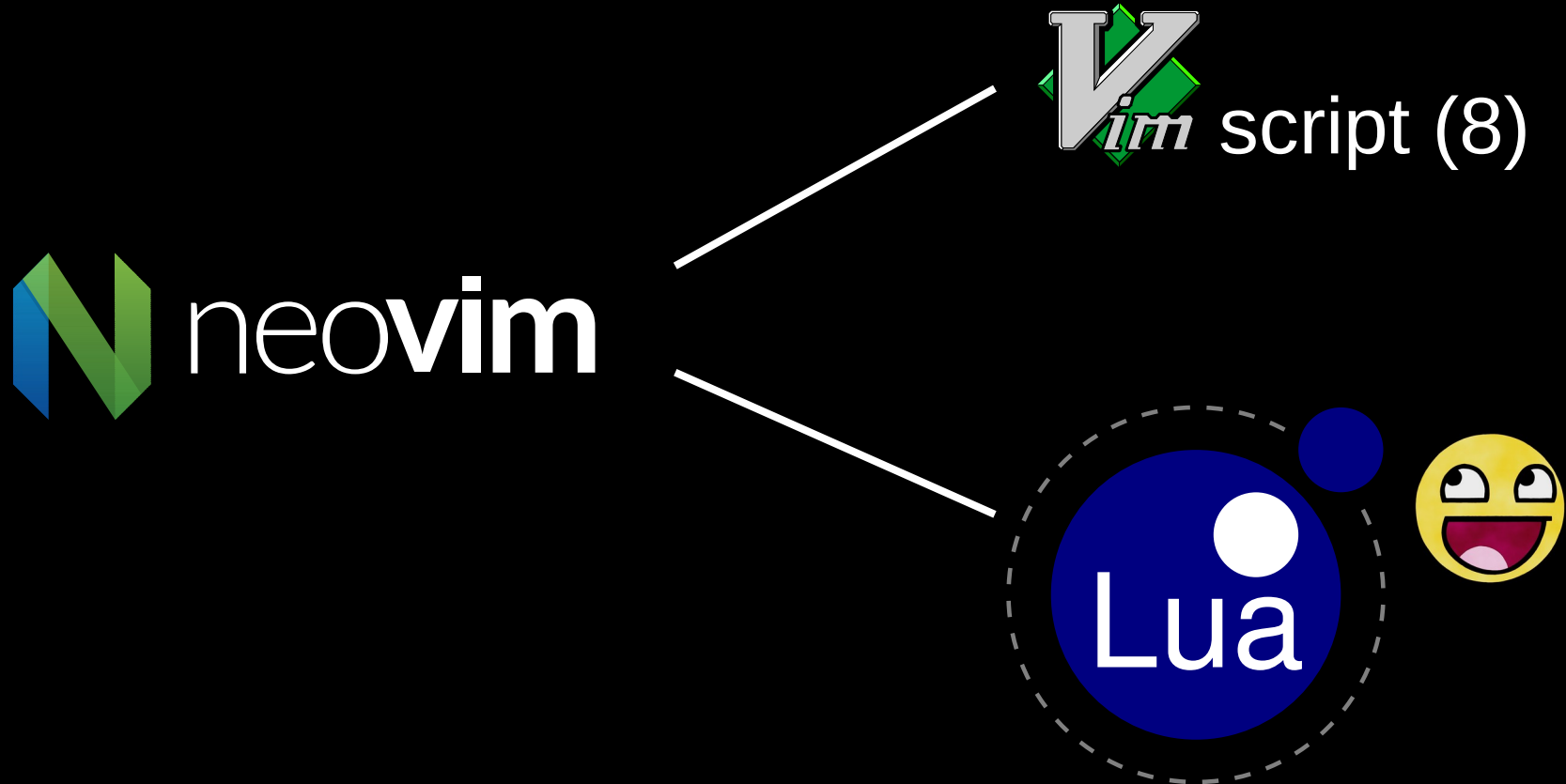
```
[*.py]  
indent_style = space  
indent_size = 4  
trim_trailing_whitespace = true
```

```
[*.json]  
indent_style = space  
indent_size = 2  
trim_trailing_whitespace = true
```


Lua API



Lua API



Lua API

- Leichtgewichtige Skriptsprache
- Gut einzubetten
- NeoVim stellt die API bereit
- Weniger schlimm als VimScript
- <https://neovim.io/doc/user/lua-guide.html>
- Zum Plugins bauen



Lua – Beispiel Diagnostic API

- [API Docs](#)
- `print(vim.inspect(vim.diagnostic.get(0)))`
- `vim.diagnostic.setqflist()`

Lua – Diagnostic API

```
vim.diagnostic.set(23, 0, {{  
  lnum = 57,  
  end_lnum = 57,  
  col = 20,  
  end_col = 30,  
  severity = vim.diagnostic.severity.ERROR,  
  message = "Hier ist was kaputt!",  
}}, {  
  underline = true,  
})
```





Plugins

- „Frisches“ Plugin Ökosystem (Lua > Vimscript)
- Es gibt für fast alles ein Plugin
- Plugin-Manager (z.B. Lazy)

lazy.nvim

- Plugin-Manager
- Hat eine Lock-Datei mit genauer Version
- Lädt Plugins „lazy“ z.B. nach Filetype
- Abhängigkeiten definieren
- ...

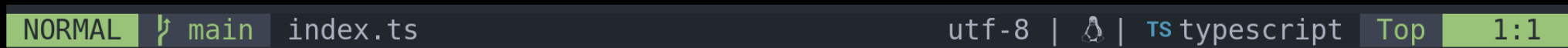
lazy.nvim

- Beispiel Plugin Konfig für lua-line.nvim

```
{  
  "nvim-lualine/lualine.nvim",  
  dependencies = { "kyazdani42/nvim-web-devicons" },  
  config = function()  
    require("lualine").setup({})  
  end  
}
```

lualine.nvim

- Statuszeile
- Batteries included



which-key.nvim



`0` → Start of line
`b` → Prev word
`B` → Prev WORD
`e` → Next end of word
`E` → Next end of WORD
`f` → Move to next char
`F` → Move to prev char
`G` → Last line
`h` → Left

`$` → End of line
`%` → Matching (){}[]
`,` → Prev ftFT
`/` → `gg` Search forward
`;` → Next ftFT
`?` → `gg` Search backward
`^` → Start of line (non ws)
`{` → Prev empty line
`}` → Next empty line

+Delete

`ESC` close `^D/^U` scroll

nvim-tree.lua

- Datei-Browser
- Zur Datei springen
- Git Status

```
8 ~/git/matrix/matrix-js-sdk/..
7 > █ .git
6 > █ .github
5 > █ .husky
4 > █ docs
3 > █ examples
2 > █ git-hooks
1 > █ ○ node_modules
  ▾ █ scripts
    1 > █ release
      2   🐍 changelog_head.py
      3   📄 switch_package_to_release.js
    4 > █ spec
    5 > █ x ★ src
    6   🌀 .babelrc
    7   🖱️ .editorconfig
    8   🌀 .eslintignore
    9   📄 .eslintrc.js
   10  📄 .git-blame-ignore-revs
```

Finden – telescope.nvim

Results

```
🔒 composer.lock:1730:60: "illuminate/con
🔒 composer.lock:1728:61: "ext-filter": "
🔒 composer.lock:1708:65: "url": "https:/
🔒 composer.lock:1703:55: "url": "https:/
🔒 composer.lock:1699:33: "name": "illuminate
🔒 composer.lock:375:18: "database",
🔒 composer.lock:371:42: "description": "Powe
🐞 tests/Feature/Database/DatabaseTest.php:25:31:
🐞 tests/Feature/Database/DatabaseTest.php:12:20: *
🐞 tests/Feature/Database/DatabaseTest.php:9:16:abstract
🐞 tests/Feature/Database/DatabaseTest.php:5:36:namespac
🐞 config/app.php:20:22: \Engelsystem\Database\Da
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
> 🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 tests/Feature/Database/DatabaseServiceProviderTest.ph
🐞 config/config.default.php:11:10: 'database' =>
🐞 config/config.default.php:9:6: 'database'
```

Grep Preview

```
use Engelsystem\Config\Config;
use Engelsystem\Database\Database;
use Engelsystem\Database\DatabaseService

class DatabaseServiceProviderTest extend
{
    /**
     * @covers \Engelsystem\Database\Dat
     */
    public function testRegister(): void
    {
        $this->app->instance('config', n
        'database' => $this->getDbCo
        'timezone' => 'UTC',
    ]]);

        $serviceProvider = new DatabaseS
        $serviceProvider->register();
        $this->assertTrue($this->app->ha
    }
}
```

Live Grep (Args)

> database

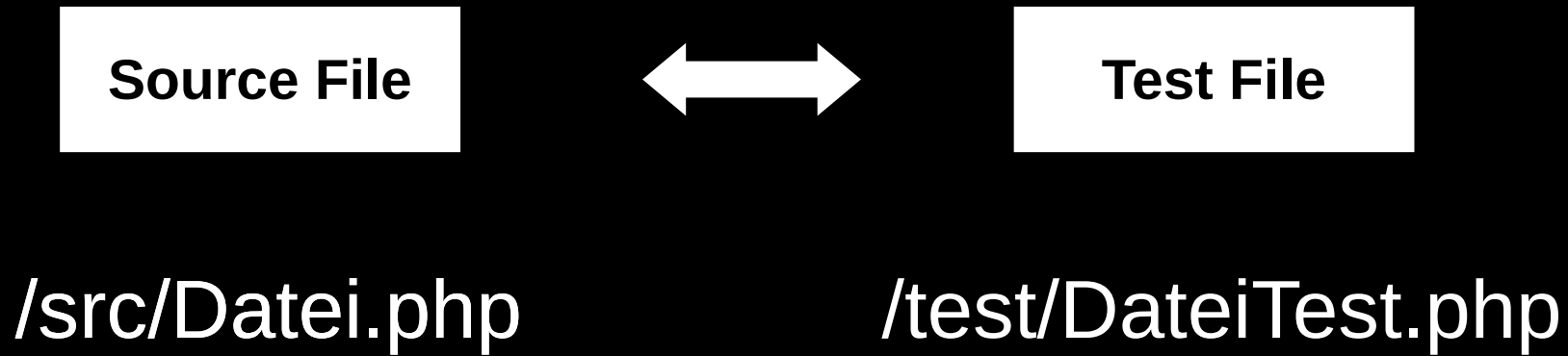
818 / 818

Telescope findet alles

- (offene) Dateien, Grep
- Diagnostic
- Git Status, Branches...
- Vim Commands, Hilfe, Options...
- ... [Liste hier](#)

other.nvim

- Zwischen verbundenen Dateien wechseln



gitsigns.nvim

- Git Status in der Sign Column

```
13  const widgetApi = useWidgetApi(),  
12  const [lastOwnDice, setLastOwnDice] = useState<  
11  // wichtiger Kommentar  
10  // ...  
9   const [dices, setDices] = useState<number[]>([])  
8  
7   useEffect(() => {  
6     const subscription = widgetApi  
5     .observeRoomEvents(STATE_EVENT_THROW_DICE)
```


diffview.nvim

```
e/s/D/DicePage.tsx | d///p/0/DiffviewFilePanel
~/git/matrix-widget-toolkit
Help: g?

Changes (2)
  example-widget-mui/src/Dice
    DicePage.tsx 2, 2
    .editorconfig

+ 1  +-- 57 lines: Copyright 2022 Nordeck IT + Consul
1  );
2  };
3
4  export const DiceView = (): ReactElement => {
5    const widgetApi = useWidgetApi();
6    const [lastOwnDice, setLastOwnDice] = useState
-----
7    const [dices, setDices] = useState<number[]>([
8
9    useEffect(() => {
10     setDices([]);
11
12     const subscription = widgetApi
13       .observeRoomEvents(STATE_EVENT_THROW_DICE)
14       .pipe(
15         filter(isValidThrowDiceEvent),
16         map((r) => r.content.pips),
17       )
+ 18 +-- 66 lines: .subscribe((d) => {.....} +

+ 12 +-- 57 lines: Copyright 2022 Nordeck IT + Consul
11  );
10  };
9
8  export const DiceView = (): ReactElement => {
7    const widgetApi = useWidgetApi();
6    const [lastOwnDice, setLastOwnDice] = useStat
// wichtiger Kommentar
5    // ...
4
3    const [dices, setDices] = useState<number[]>([
2
1    useEffect(() => {
-----
69  const subscription = widgetApi
1    .observeRoomEvents(STATE_EVENT_THROW_DICE)
2    .pipe(
3      filter(isValidThrowDiceEvent),
4      map((r) => r.content.pips),
5    )
6    lines: .subscribe((d) => {.....} +
```

diffview.nvim

- Aktuelle Änderungen
- Branches
- Datei Historie
- Merge Tool

neotest

```
5
✓ 4 class StringInputLengthTest extends TestCase
3 {
2     /**
1     * @covers      \Engelsystem\Http\Validation\Rules\StringInputLength::validate
14  * @covers      \Engelsystem\Http\Validation\Rules\StringInputLength::isDateTime
1     * @dataProvider validateProvider
2     */
✓ 3 public function testValidate(mixed $input, mixed $expectedInput): void
4 {
5     $rule = new UsesStringInputLength();
6     $rule->validate($input);
7
8     $this->assertEquals($expectedInput, $rule->lastInput);
9 }
```

NORMAL main StringInputLengthTest.php utf-8 < < php

Time: 00:00.015, Memory: 8.00 MB

OK (16 tests, 16 assertions)

Neotest Output Panel [-]

Debugger

- Debug Adapter Protocol
 - (vgl. Language Server Protocol)
- nvim-dap
- nvim-dap-ui

```

locals:
> class variables
> m NoneCheckModel = NoneCheckModel(required_str_value='v1', required_str_none_
> special variables
  > __abstractmethods__ frozenset = frozenset({})
  > __annotations__ dict = {'required_str_value': <class 'str'>, 'required_str_
  > __class__ ModelMetaClass = <class 'tests.test_main.test_nullable_strings_su
  > __class_vars__ set = {}
  > __config__ type = <class 'tests.test_main.Config'>
  > __custom_root_type__ bool = False
  > __delattr__ method-wrapper = <method-wrapper '__delattr__' of NoneCheckMode
  > __dict__ dict = {'required_str_value': 'v1', 'required_str_none_value': Non
  > __dir__ builtin_function_or_method = <built-in method __dir__ of NoneCheckM
  > __doc__ NoneType = None

```

```

test_main.py:
131 assert m.required_str_value == 'v1'

```

```

MainThread:
> test_nullable_strings_success test_main.py:131
run pytest.py:118
main __init__.py:52
<module> neotest.py:20

```

```

> m._fields__ dict = {'required_str_value': ModelField(name='req...ired=True),
> special variables
> function variables
  > 'required_str_value' ModelField = ModelField(name='required_str_value', type=
  > 'required_str_none_value' ModelField = ModelField(name='required_str_none_val
  > 'required_bytes_value' ModelField = ModelField(name='required_bytes_value', t
  > 'required_bytes_none_value' ModelField = ModelField(name='required_bytes_none
  > 'existing_str_value' ModelField = ModelField(name='existing_str_value', type=
  > 'existing_bytes_value' ModelField = ModelField(name='existing_bytes_value', t
  len() int = 6

```

```

> tests > test_main.py
22
21     assert str(e.value) == "you need to set the type of field 'a' when using 'default_factory'"
20
19
18 def test_comparing():
17     m = UltraSimpleModel(a=10.2, b='100')
16     assert m == {'a': 10.2, 'b': 100}
15     assert m == UltraSimpleModel(a=10.2, b=100)
14
13
12 def test_nullable_strings_success():
11     class NoneCheckModel(BaseModel):
10         existing_str_value = 'foo'
9         required_str_value: str = ...
8         required_str_none_value: NoneStr = ...
7         existing_bytes_value = b'foo'
6         required_bytes_value: bytes = ...
5         required_bytes_none_value: NoneBytes = ...
4
3         m = NoneCheckModel(
2             required_str_value='v1', required_str_none_value=None, required_bytes_value='v2', required_bytes_none_value=None
1         )
→ 131     assert m.required_str_value == 'v1'
1     assert m.required_str_none_value is None
2     assert m.required_bytes_value == b'v2'
3     assert m.required_bytes_none_value is None
4
5
6 def test_nullable_strings_fails():
7     class NoneCheckModel(BaseModel):
8         existing_str_value = 'foo'
9         required_str_value: str = ...
10        required_str_none_value: NoneStr = ...
11        existing_bytes_value = b'foo'
12        required_bytes_value: bytes = ...
13        required_bytes_none_value: NoneBytes = ...
14
15        with pytest.raises(ValidationError) as exc_info:
16            NoneCheckModel(
17                required_str_value=None,
18                required_str_none_value=None,
19                required_bytes_value=None,
20                required_bytes_none_value=None,
21            )
22        assert exc_info.value.errors() == [


```

```

Test session starts (platform: linux, Python 3.10.5, pytest 7.1.1, pytest-sugar
0.9.4)
rootdir: /home/ronan/Dev/repos/pydantic, configfile: setup.cfg
plugins: mock-3.7.0, cov-3.0.0, sugar-0.9.4
collecting ...

```

cutlass.nvim

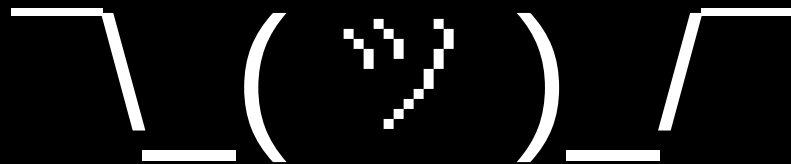
- z.B. dd (löschen) schneidet quasi aus
 - „Zwischenablage“ weg
- 
- cutlass setzt bei d, x, etc. keine Register
- Definiert dafür eine Ausschneidenaktion



Unendliche Möglichkeiten

- <https://neovim.io/>
→ Get Plugins
- [awesome-neovim](#)
→ Kuratierte Liste

Warum?





- Langsam (klebrig)
- Instabil
- RAM Vernichter
- Arbeitet gegen mich



- Langsam (klebrig)
- Instabil
- RAM Vernichter
- Arbeitet gegen mich



- Keine (ordentliche) Autocompletion
- Wenig(er) Features



- Langsam (klebrig)
- Instabil
- RAM Vernichter
- Arbeitet gegen mich



Vim mode



- Keine (ordentliche) Autocompletion
- Wenig(er) Features



- „Vim Sprache“
- hjkl
- ciw



- Langsam (klebrig)
- Instabil
- RAM Vernichter
- Arbeitet gegen mich



Vim mode



- Language Server
- Plugins
- Handgeklöppelte Konfig



- Keine (ordentliche) Autocompletion
- Wenig(er) Features



- „Vim Sprache“
- hjkl
- ciw

Vim „Sprache“

dd

Zeile löschen



5

dd

Fünf Zeilen löschen...

ciw

klackerdie -
klack

Wort ändern
(change inner word)

ESC



.

nochmal!

⚠ Sicherheitshinweis

- Kann Spuren von Rabbit Hole enthalten



Vim überall



Im Browser



Datei-Manager

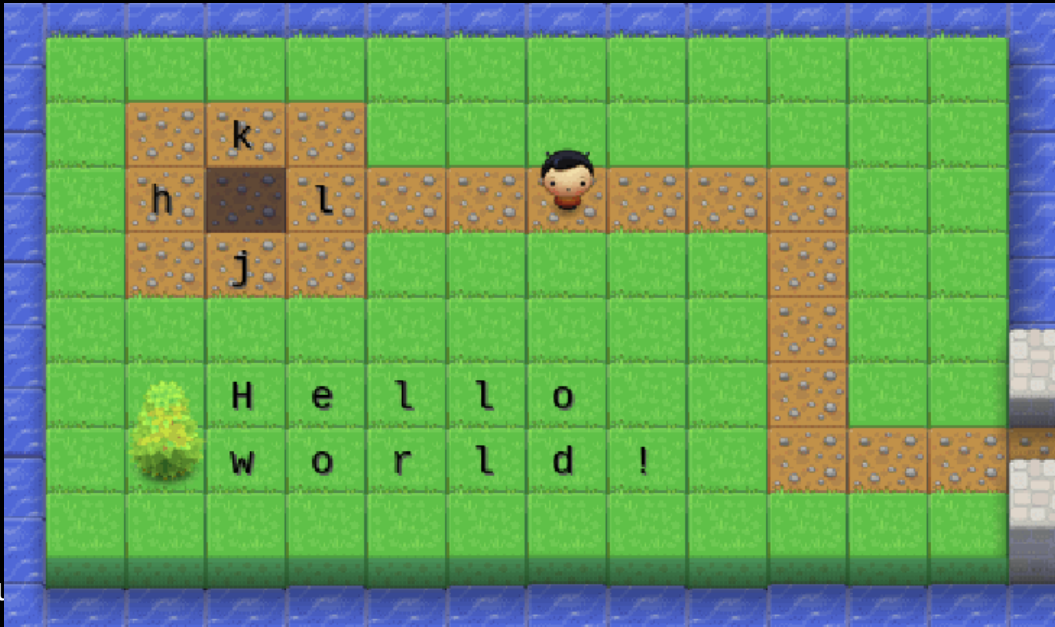


iamb – Matrix Client



Tipps – Vim lernen

- :Tutor
- <https://vim-adventures.com/>



IDE

vs

PDE

- Integrated Development Environment
- Kann ootb einiges
- Ich lerne die IDE
- Begrenzt anpassbar

- Personalised Development Environment
- Kann ootb weniger
- PDE macht, was ich will
- Sehr anpassbar

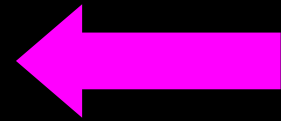
Restekiste

Migration Vim → NeoVim

- Die `init.vim` kann weiter verwendet werden
- **NeoVim installieren**
- `init.vim` nach `~/.config/nvim` kopieren
- `mkdir ~/.config/nvim/lua`
- `echo "print(,Hallo Lua´)" > ~/.config/nvim/lua/neovim.lua`
- `lua require(,neovim´)` in die `init.vim`

Konfig Einstiegspunkte

- Start
 - Zu Fuß: [neovim-konfig-workshop](#)
 - Mehr drin: [kickstart.nvim](#)
 - Komplettpaket: [LazyVim](#)




Changelog

- <https://neovim.io/roadmap/>
- 0.10 (aktuell)
 - Farbschema
 - LSP inlay hints
 - LSP snippets
 - Aus-/Einkommentieren
 - ...

Roadmap

- <https://neovim.io/roadmap/>
- 0.11 (nächste)
 - Eingebaute Autocompletion
 - `vim.lsp.server()`
 - Async refactoring

NeoVim – Ab ins Rabbit Hole

- Kommt gestern zum Vim Basics und NeoVim Workshop 
- <https://codeberg.org/weeman>
- <https://neovim.io/>
- <https://github.com/rockerBOO/awesome-neovim>

BOOT TO NEOVIM!

