

Sublime Text 3 para Programar em Python

Scripts de Configuração

1. Alterar a execução para o comando python3, ao invés do python

```
{
    "cmd": ["python3", "-i", "-u", "$file"],
    "file_regex": "^[ ]File \"(...?)\", line ([0-9]*)",
    "selector": "source.python"
}
```

2. Executar linhas selecionadas do Código

2.1 Criação do plugin

```
import sublime, sublime_plugin

class RunSelectionCommand(sublime_plugin.TextCommand):
    def run(self, edit, **kwargs):
        import re, os
        import tempfile

        chunks = []
        for region in self.view.sel():
            chunks.append(self.view.substr(region))

        if self.view.file_name():
            working_dir = os.path.dirname(self.view.file_name())
        else:
            working_dir = os.getcwd()

        chunks = "\n".join(chunks)
        lines = filter(
            None, [l for l in chunks.split("\n") if l.strip() != ""]
        )
        source_code = "\n".join(lines)

        with tempfile.NamedTemporaryFile(suffix='.py', mode='w',
delete=False) as f:
            f.write(source_code)
            window = sublime.active_window()
            window.run_command("exec", {
                "shell_cmd": "python3 {}".format(f.name)
            })

# Run the build now
```

```
self.window.run_command("build", {"variant": "Selection"})

def is_enabled(self):
    return len(self.view.sel()) > 0
```

2.2 Definição do Atalho

```
[
    { "keys": ["ctrl+alt+n"], "command": "run_selection", "context":
        [
            { "key": "num_selections", "operator": "equal", "operand": 1 }
        ]
    }
]
```