

Visual Studio Code

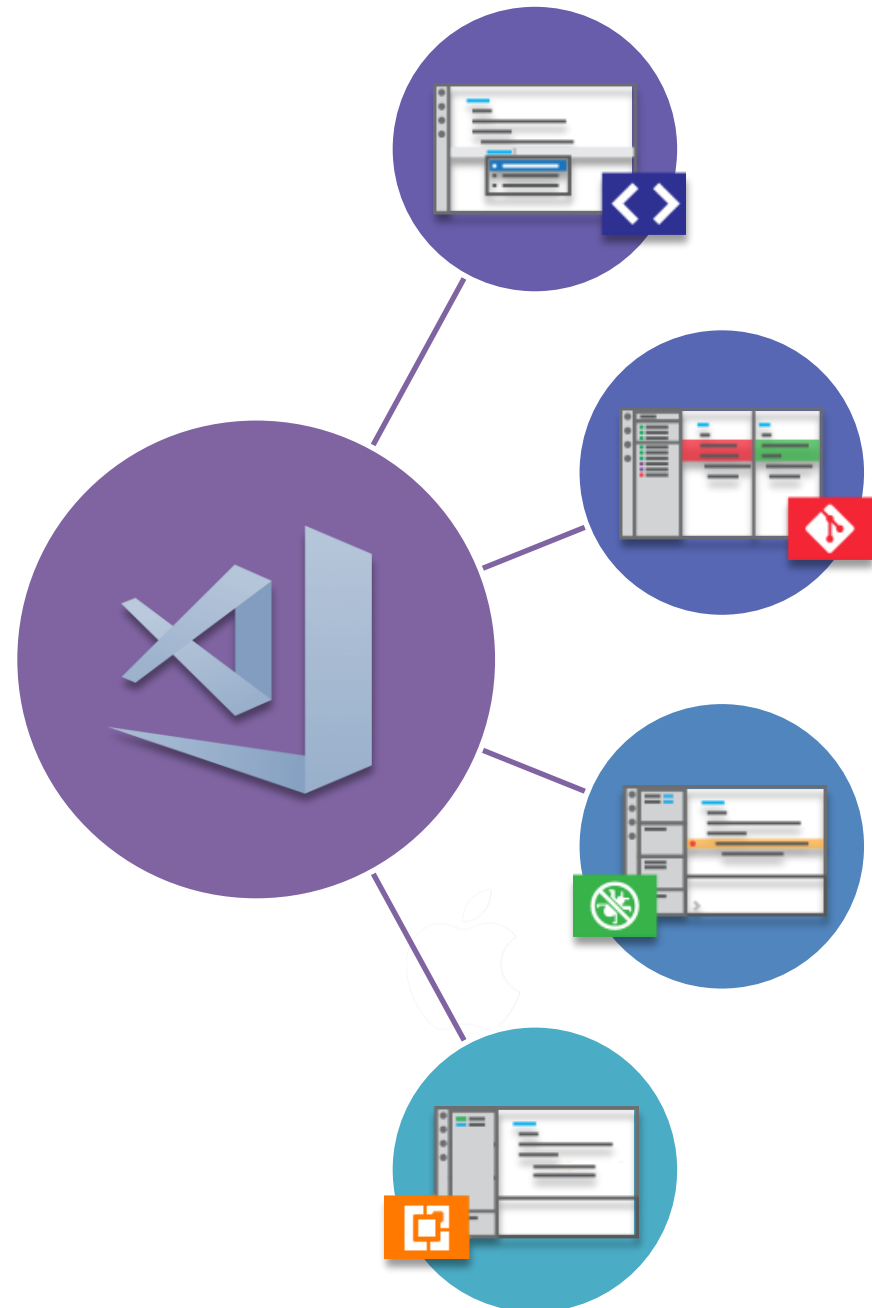
Your Next Coding
Companion for
Advanced Research
Computing

Armin Sobhani
asobhani@sharcnet.ca

SHARCNET

University of Ontario Institute
of Technology (UOIT)

28 February, 2018



Links

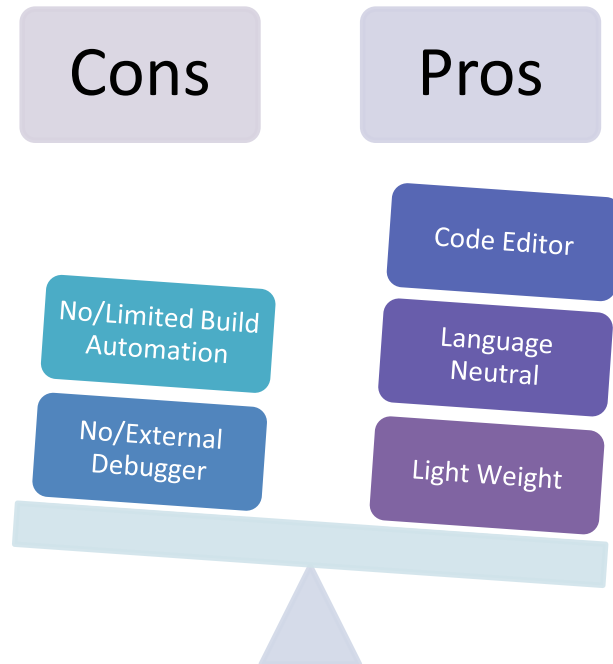
- Nix Package Manager <https://youtu.be/pQE9WTLAPHQ>
- Git Version Control <https://youtu.be/meFv-GDTkjE>
- CMake Part I <https://youtu.be/7wxzoAdZcKE>
- CMake Part II <https://youtu.be/Tsd6bAW8uLg>
- Tutorial Files <https://git.sharcnet.ca/asobhani/cmake-tutorial>

Visual Studio Code (VSCode)

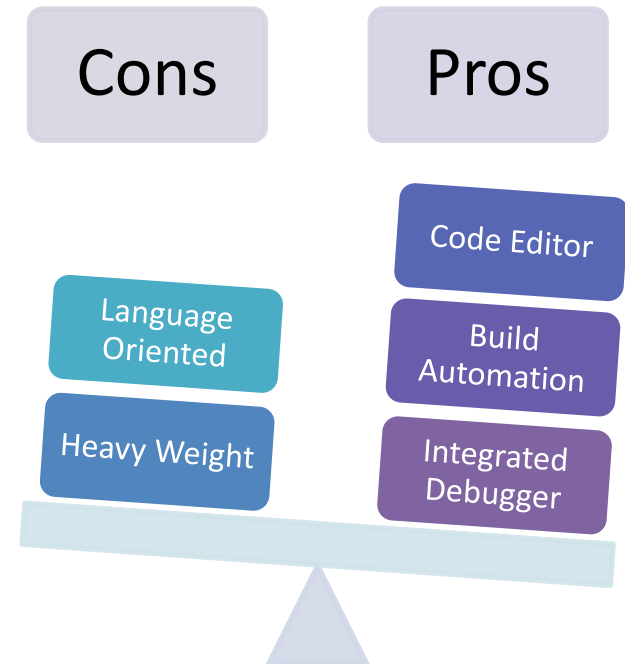
- GUI Source Code Editor (SCE) / Debugger
 - Free (source: [MIT](#), binaries: [Freeware](#))
 - Open-source (<https://github.com/Microsoft/vscode>)
 - Multiplatform (Windows | Linux | MacOS)
- Has been being developed by Microsoft since 2015
- Written in TypeScript / JavaScript / CSS and based on Electron

SCEs vs. IDEs

SCEs



IDEs



SCEs vs. IDEs

SCEs



IDEs

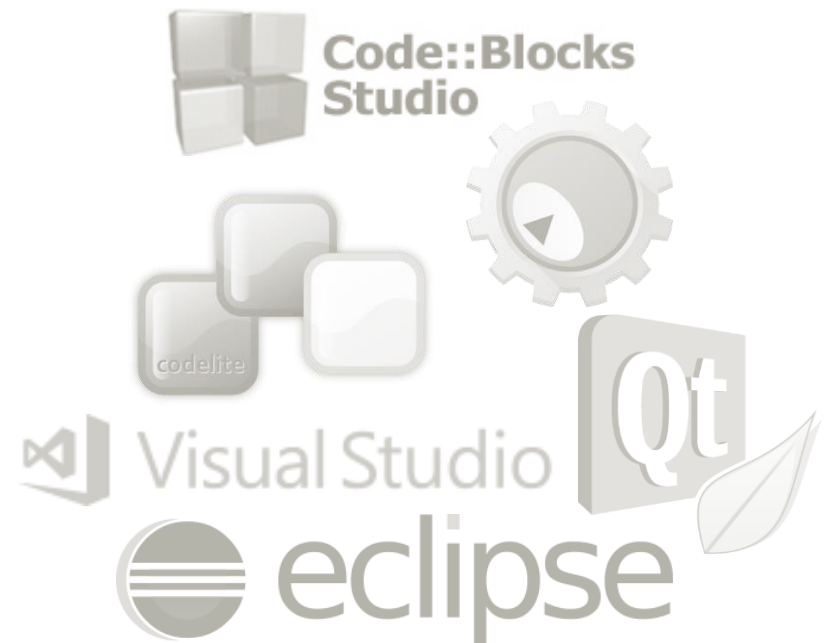


SCEs vs. IDEs

SCEs



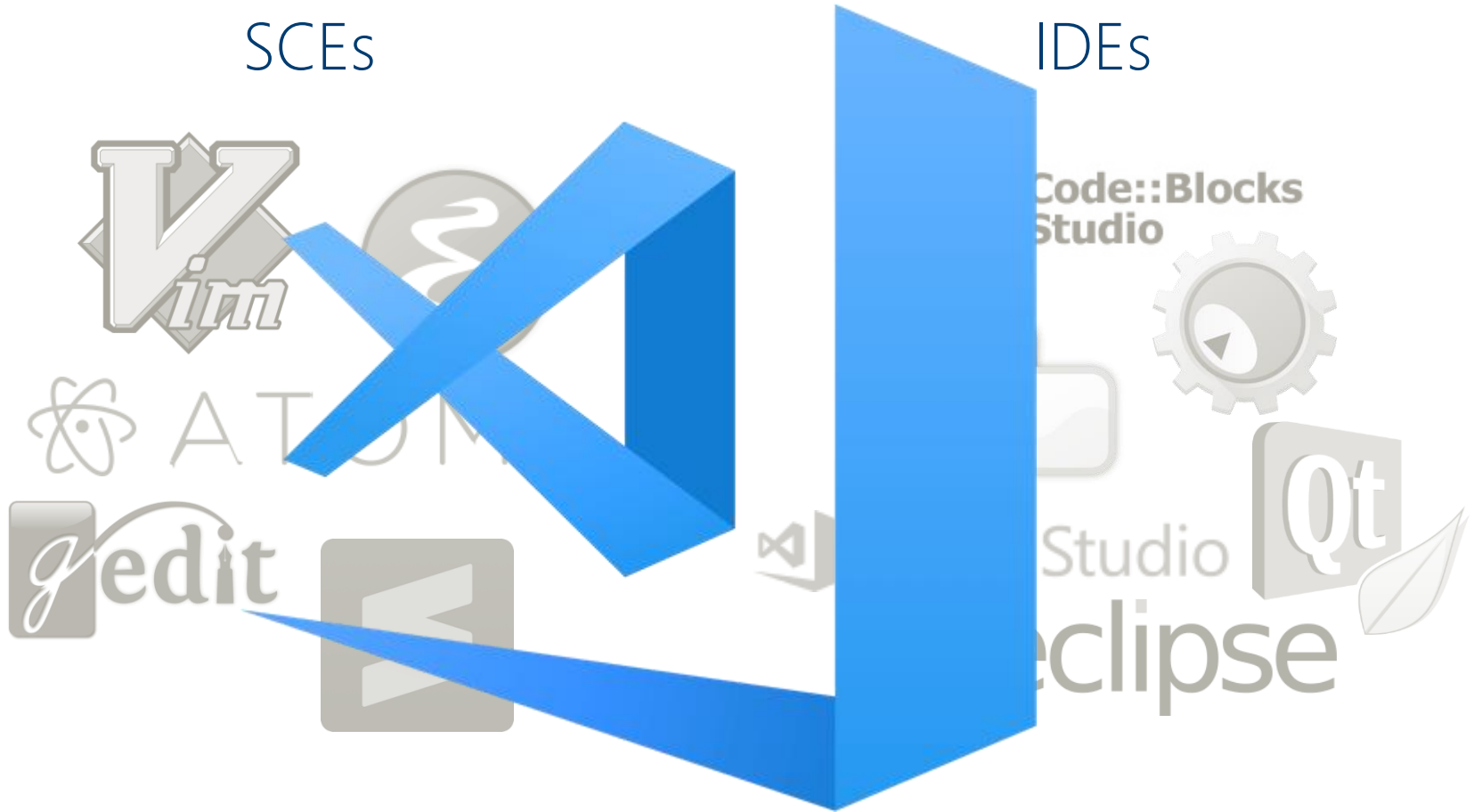
IDEs



SCEs vs. IDEs

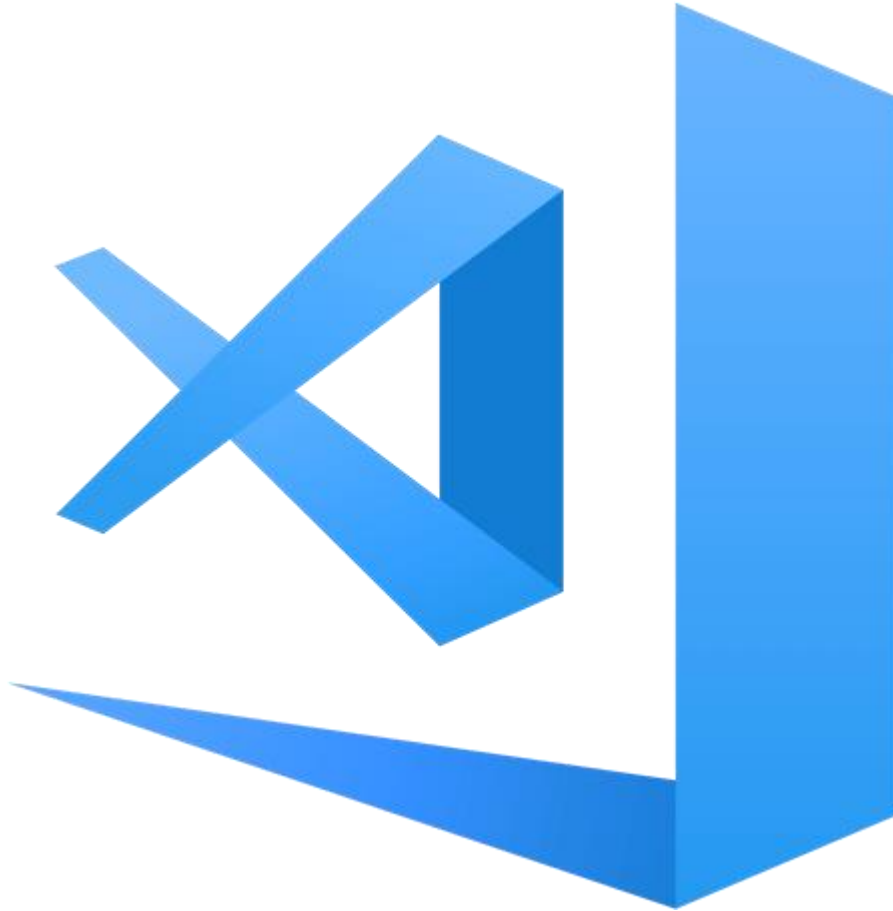
SCEs

IDEs

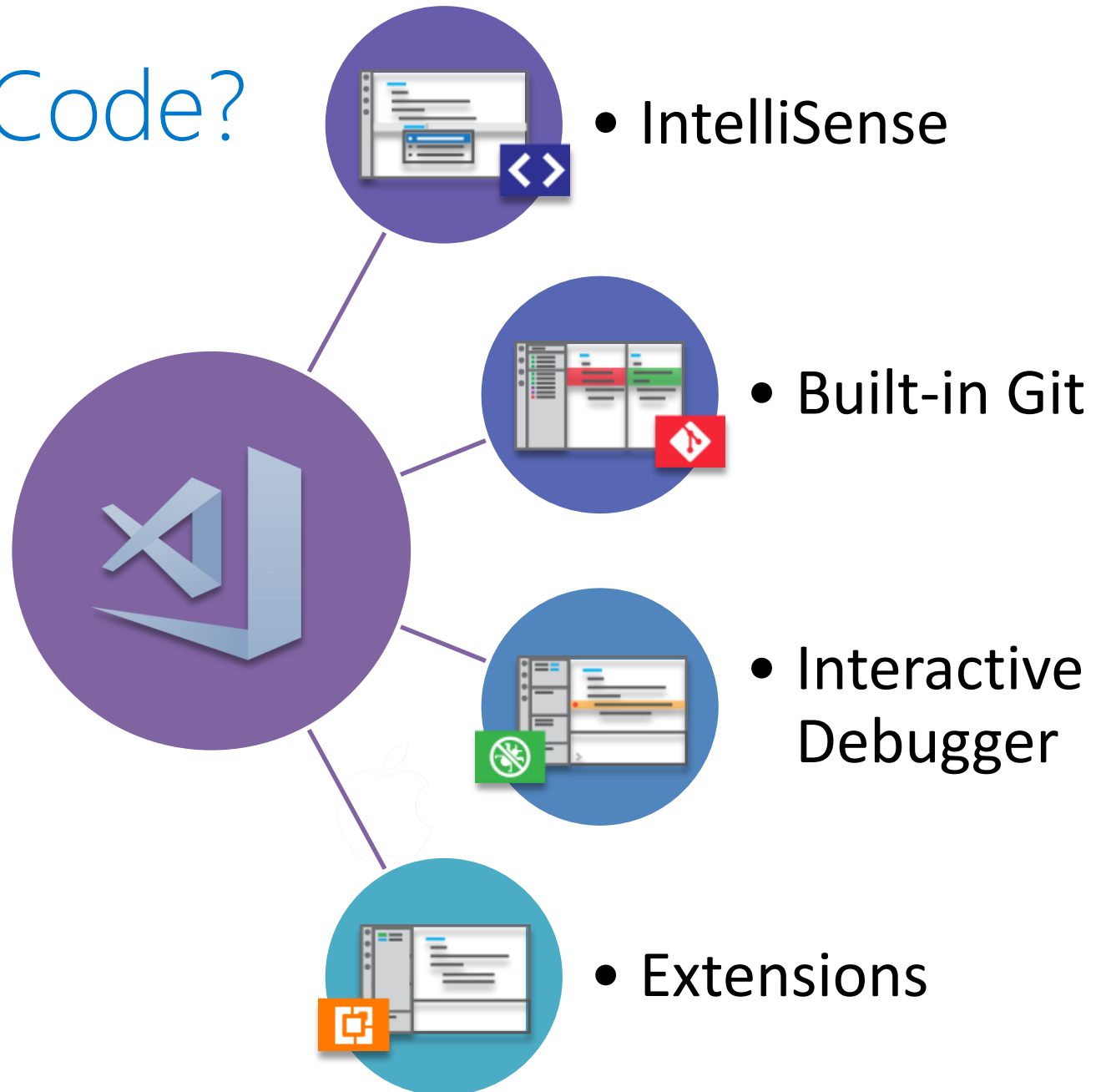


Why VSCode?

Good Balance between SCE and IDE



Why VSCode?



IntelliSense



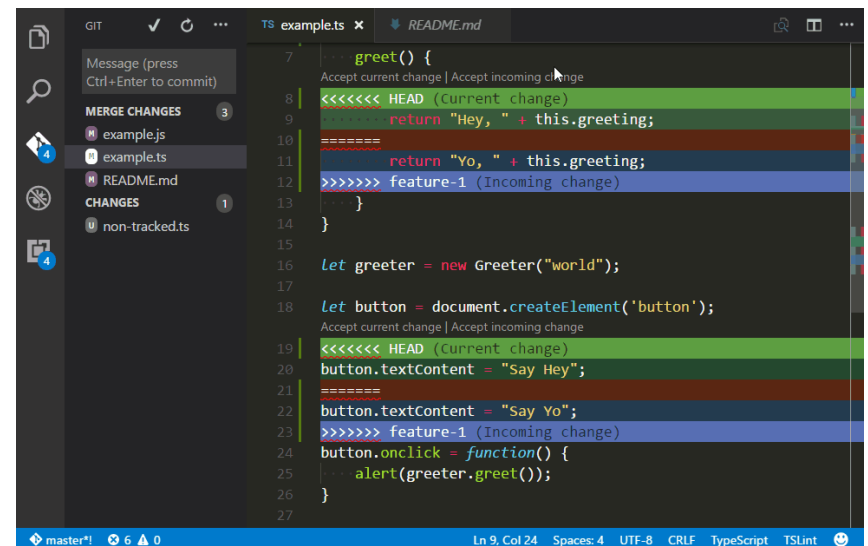
- Smart completions based on variable types, function definitions, and imported modules

```
api_wrapper.py
1 from __future__ import absolute_import
2 import time
3 import logging
4 import random, base64, struct
5 import hashlib
6 import os
7 from pgoapi.exceptions import (ServerSideRequestThrottlingException,
8                                NotImplemented, ServerBusyOrOfflineException,
9                                NoPlayerPositionSetException, EmptySubrequestChainException,
10                                UnexpectedResponseException)
11 from pgoapi.pgoapi import PGoApi
12 from pgoapi.pgoapi import PGoApiResponse
13 from pgoapi.pgoapi import RpcApi
14 from pgoapi.protos.POGOProtos.Networking.Requests.RequestType_pb2 import RequestType
15 from pgoapi.utilities import get_time
16 from .human_behavior import sleep, gps_noise_rng
17 from pokemonbot.base_dir import _base_dir
18
19
20 class PermaBannedException(Exception):
21     pass
22
23 class ApiWrapper(PGoApi, object):
24     DEVICE_ID = None
25
26     def __init__(self, config=None):
27         self.config = config
28         self.gen_device_id()
29         device_info = {
30             "device_id": ApiWrapper.DEVICE_ID,
31             "device_brand": "Apple",
32             "device_model": "iPhone",
33             "device_model_boot": "iPhone8,2",
34             "hardware_manufacturer": "Apple",
35             "hardware_model": "N66AP",
36             "firmware_brand": "iPhone OS",
37             "firmware_type": "9.3.3"
38         }
```

Built-In Git Support



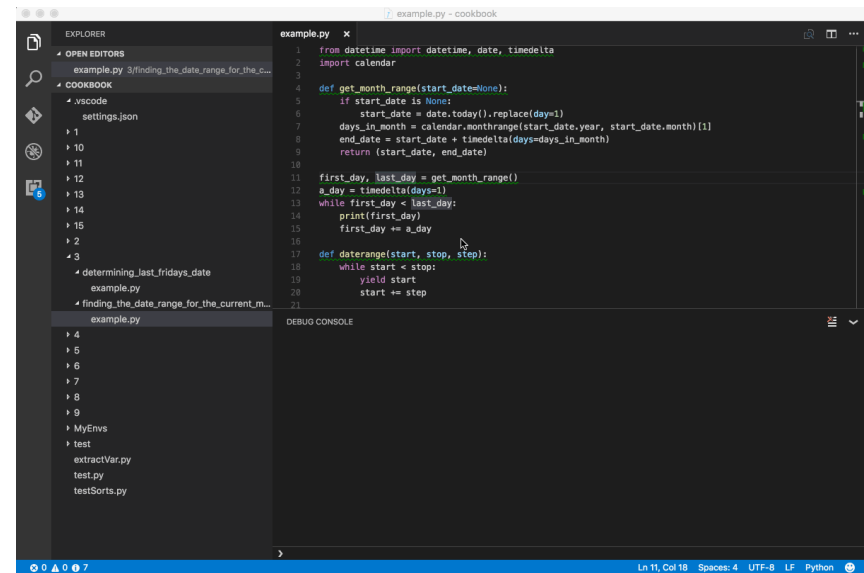
- Review diffs, stage files, and make commits right from the editor
- Push and pull from any hosted SCM service



Interactive Debugger



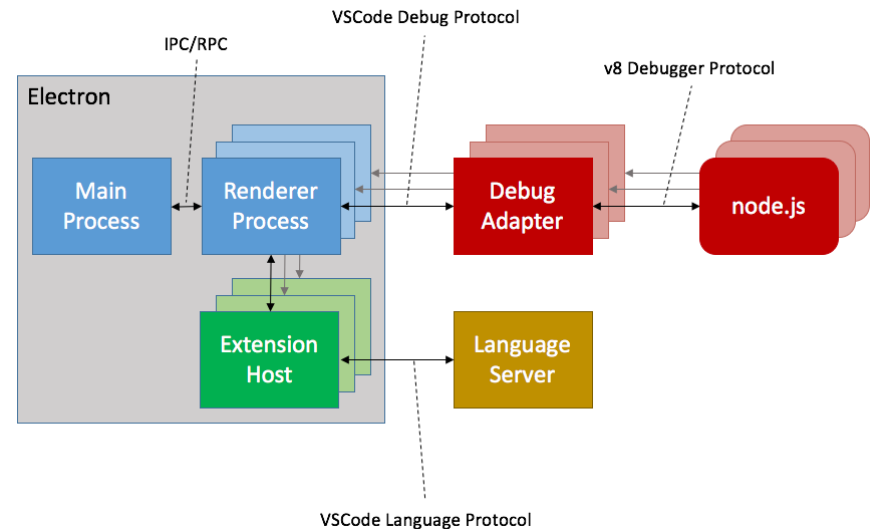
- Debug code right from the editor
- Launch or attach to your running apps
- Debug with break points, call stacks, and an interactive console



Interactive Debugger



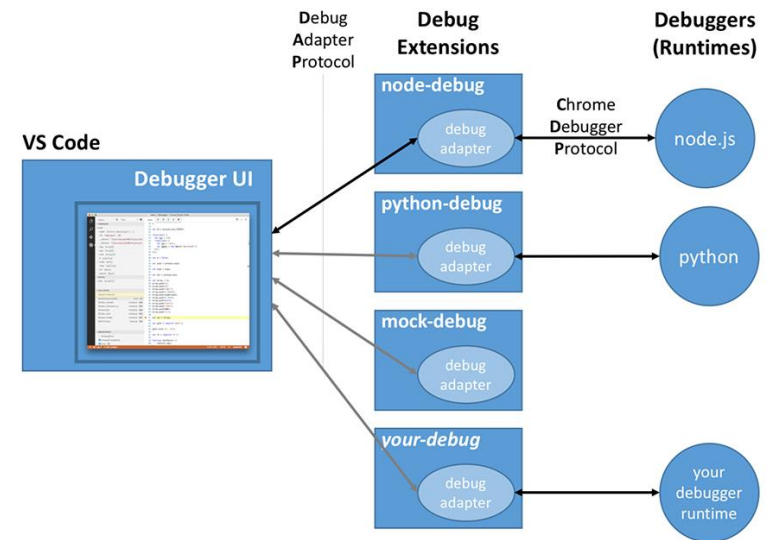
- A generic (language agnostic) debug UI
- Cannot talk to real debuggers directly
- Relies on debug extensions (Adaptor) for implementing the debugger



Interactive Debugger



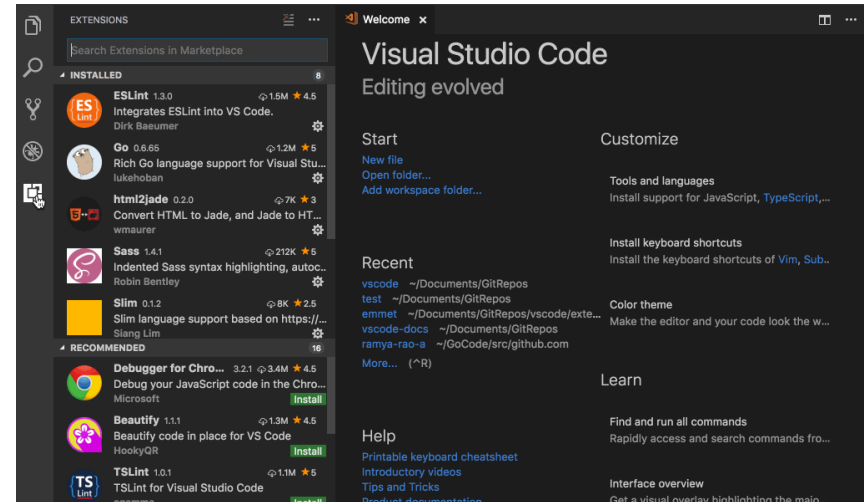
- Ships with a debug extension for Node.js
- More debugger extensions are available from the VSCode Marketplace



Extensions

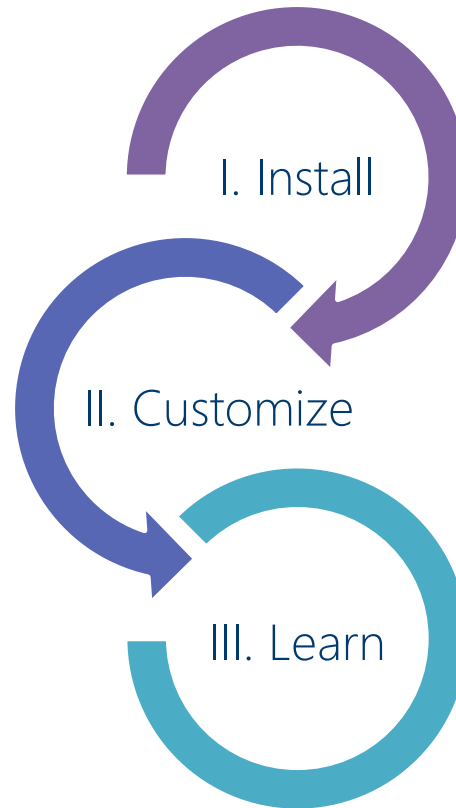


- Languages, themes, debuggers, tools, etc.
- Extensions run in separate processes and won't slow the editor



VSCode Setup

Setup Overview



I. Installation

- Follow the platform specific guides:
<https://code.visualstudio.com/docs/setup>

- Sharcnet Systems:
 - Already installed on some legacy VDI nodes
 - Using Nix Package Manager:

```
$ module load nix
$ export NIXPKGS_ALLOW_UNFREE=1
$ nix-env -qa --attr-path --description | grep vscode
$ nix-env -iA nixpkgs.vscode-with-extensions
```

I. Installation

- VDI Nodes on Graham:
gra-vdi.user.sharcnet.ca
- Caveats
 - The module system is not operational yet
 - The project filesystem is not mounted yet
- Using Nix Package Manager:

```
$ cat vdi-setup.sh
source /nix/etc/profile.d/nix-profile.sh
export export SSL_CERT_FILE=/etc/pki/tls/cert.pem
$ source vdi-setup.sh
$ nix-env -iA nixpkgs.git nixpkgs.cmake
$ code
```

II. Customizing (using Extensions)

Themes

- Color, File Icons

Keymaps

- Vim, Sublime, Emacs, Atom, Visual Studio, Eclipse, etc.

Languages

- C++, C#, Go, Java, Python, etc.

Tools

- ESLint, JSHint, CMake Tools, etc.

Debuggers

- Chrome, PHP Xdebug, GDB, LLDB, etc.

Fine Tuning

- Tinkering with [Settings](#) (Ctrl+,)

III. Learning

Find and Run All Commands

- Command Palette ([Ctrl+Shift+P](#), [F1](#))

Interface Overview

- Get a visual overlay highlighting the major components of the UI

Interactive Playground

- Try essential editor features out in a short walkthrough