

Visual Studio Code

William Harrell
LUG@NC State
March 20, 2019



Program goes here :)

The screenshot displays the Visual Studio Code interface during a Live Share session. The main editor shows a Python script named `train_ann.py` with the following code:

```
26
27 # split data into a training set and a test set
28 from sklearn.model_selection import train_test_split
29 # test_size: percentage of data to use for test set
30 # random_state: seed used for random selection
31 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=0)
32
33 # feature scaling: makes computation less intense and ensures that just because some features are big it won't lead to using them as a main predictor"
34 from sklearn.preprocessing import StandardScaler
35 sc = StandardScaler()
36 x_train = sc.fit_transform(x_train)
37 x_test = sc.transform(x_test)
38
39 import json
40 with open("transform.json", "w") as f:
41     json.dump(sc.get_params(), f)
```

To the right, a table displays data from a CSV file named `'Churn_Modelling.csv'`:

Row Number	Customer Id	Surname	Credit Score	Geography	Gender
1	15634602	Hargrave	619	France	Female
2	15647311	Hill	608	Spain	Female
3	15619304	Onio	502	France	Female
4	15701354	Boni	699	France	Female
5	15737888	Mitchell	850	Spain	Female
6	15574012	Chu	645	Spain	Male
7	15592531	Bartlett	822	France	Male
8	15656148	Obinna	376	Germany	Female
9	15792365	He	501	France	Male
10	15592389	H?	684	France	Male
11	15767821	Bearce	528	France	Male
12	15737173	Andrews	497	Spain	Male
13	15632264	Kay	476	France	Female
14	15691487	Chin	549	France	Female
15	15600882	Scott	635	Spain	Female
16	15643966	Goforth	616	Germany	Male
17	15737452	Romeo	653	Germany	Male
18	15788218	Henderson	549	Spain	Female
19	15661507	Muldrow	587	Spain	Male
20	15568982	Hao	726	France	Female
21	15577657	McDonald	732	France	Male
22	15597945	Dellucci	636	Spain	Female

The bottom panel shows the PROBLEMS window with several errors:

- single_prediction.py: 3: Unable to import 'keras' pylint(import-error) [1, 1]
- single_prediction.py: 8: unresolved import 'keras' Python(unresolved-import) [1, 8]
- single_prediction.py: 14: 'keras' used before definition Python(use-before-def) [4, 14]
- train_ann.py: 17: Unable to import 'pandas' pylint(import-error) [3, 1]
- train_ann.py: 14: Unable to import 'sklearn.preprocessing' pylint(import-error) [14, 1]
- train_ann.py: 28: Unable to import 'sklearn.model_selection' pylint(import-error) [28, 1]
- train_ann.py: 34: Unable to import 'sklearn.preprocessing' pylint(import-error) [34, 1]
- train_ann.py: 44: Unable to import 'keras' pylint(import-error) [44, 1]
- train_ann.py: 45: Unable to import 'keras.models' pylint(import-error) [45, 1]
- train_ann.py: 46: Unable to import 'keras.layers' pylint(import-error) [46, 1]
- train_ann.py: 85: Unable to import 'sklearn.metrics' pylint(import-error) [85, 1]
- train_ann.py: 3: unresolved import 'pandas' Python(unresolved-import) [3, 8]
- train_ann.py: 8: 'pd' used before definition Python(use-before-def) [8, 11]
- train_ann.py: 14: unresolved import 'sklearn.preprocessing' Python(unresolved-import) [14, 6]
- train_ann.py: 28: unresolved import 'sklearn.model_selection' Python(unresolved-import) [28, 6]
- train_ann.py: 34: unresolved import 'sklearn.preprocessing' Python(unresolved-import) [34, 6]
- train_ann.py: 44: unresolved import 'keras' Python(unresolved-import) [44, 8]

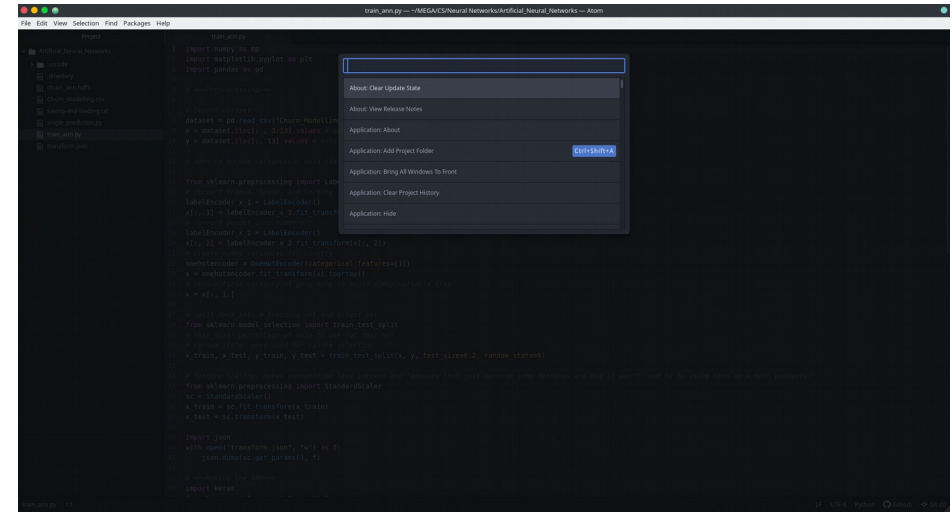
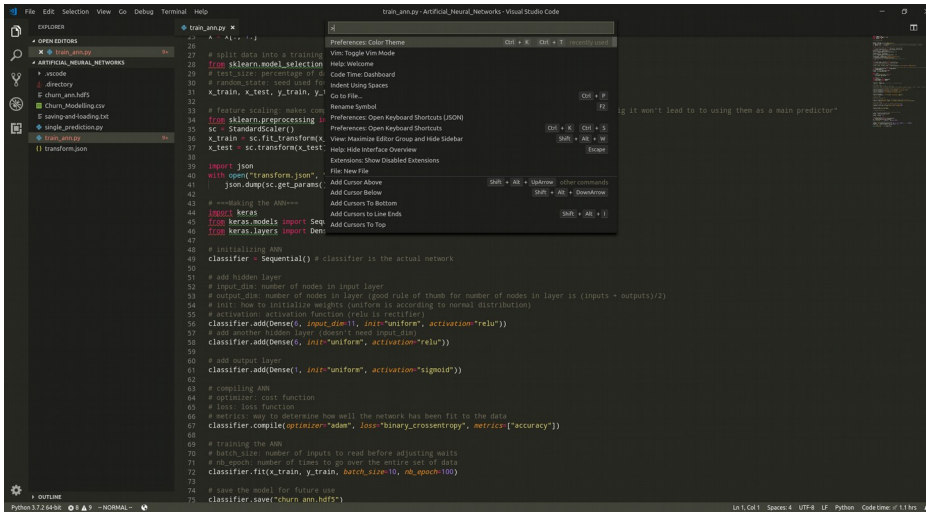
The Live Share Chat window on the right shows a message from `prophetofxenu` at 4:57 PM: "has started the Live Share session".

Overview

- **Written with TypeScript and Electron by Microsoft**
- **MIT Licensed**
- **Built-in support for TypeScript, JavaScript, and Node applications**
- **Configured through JSON**
- **Easily upgraded with extensions to build the IDE of your dreams for whatever language your heart desires**



VS Code vs Atom



VS Code

Developed by Microsoft

Lightweight out of the box, but faster

Native support for TS, JS and Node

Decent Git support

Atom

Developed by GitHub Microsoft

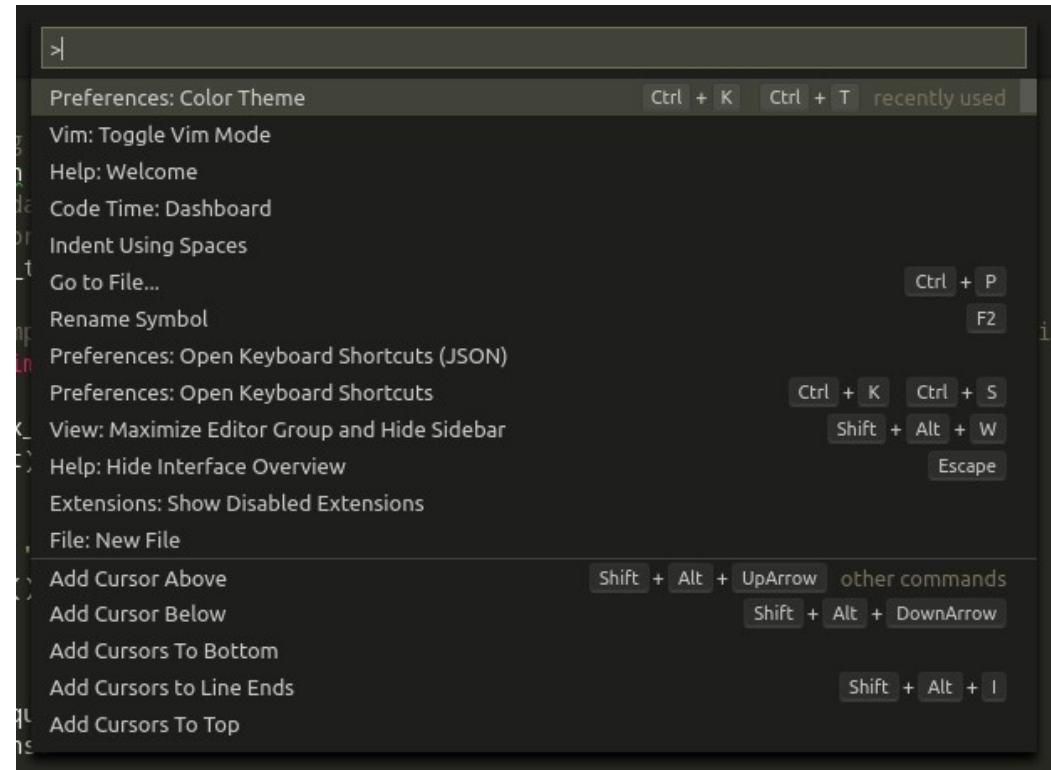
Large number of default packages, almost every component is a package

Native support for very large number of languages

Excellent GitHub (and Git) support

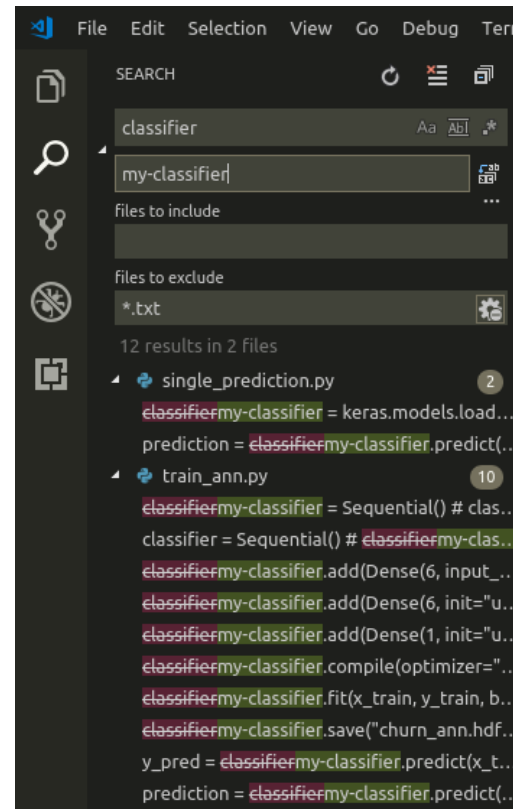
Command Palette (demo)

- Easily access every possible action with **Ctrl+Shift+P**
- Type what you want to do instead of having to memorize keyboard shortcuts
- Memorize keyboard shortcuts anyway because it shows you the shortcut associated with an action
- Undo stops created before executing commands that create edits
- Extensions can add to the command palette; not limited to built-in functionality

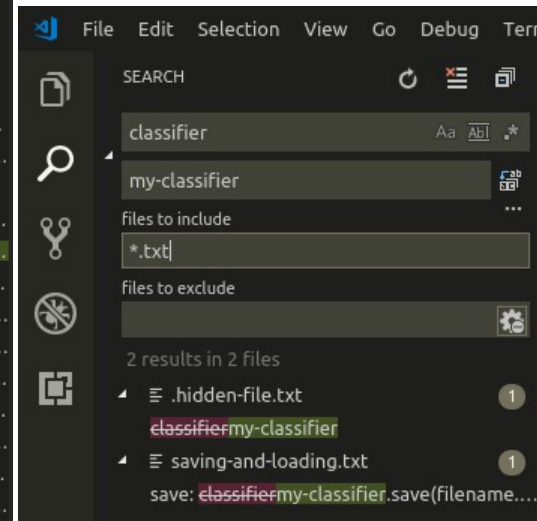


Find and Replace (demo)

- Search for a phrase across entire workspace
- Get a list of all occurrences; replace individually or all
- Supports regex
- Auto ignore .gitignored and pre-configured patterns in settings



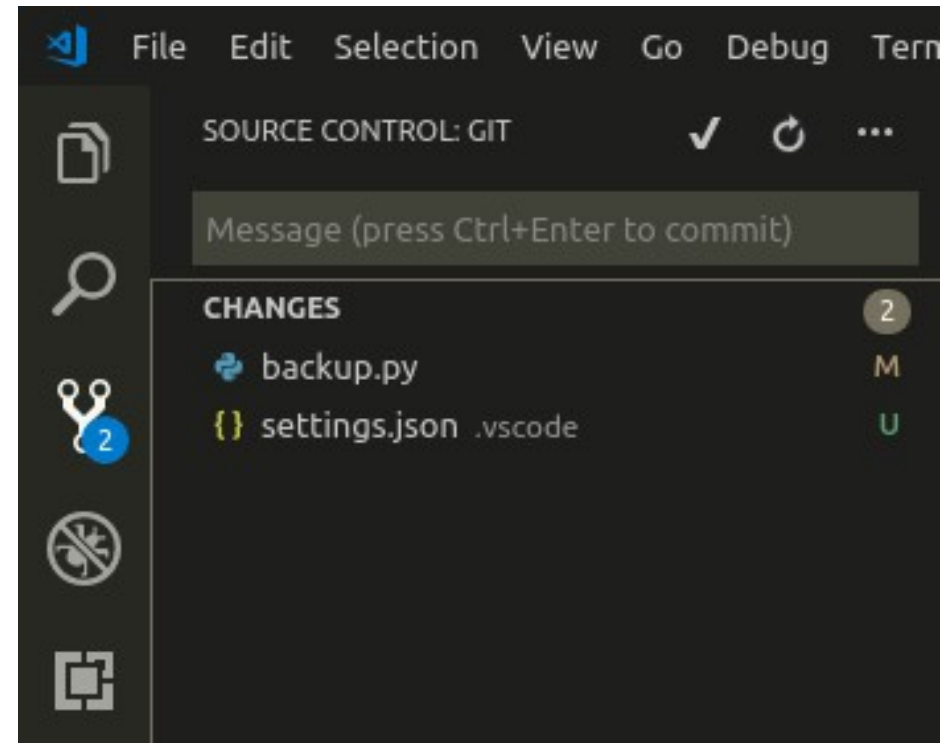
Searches across entire workspace by default



Explicit includes only search those includes

Source Control (demo)

- Simple but powerful Git interface
- List all uncommitted changes
- Work with staging area, remotes, tags, branches, stashes, conflicts either through sidebar or command palette
- Additional SCM support through extensions



Side by Side Change View

- View highlighted changes to a file
- Edit while viewing changes

```
124 def confirm(question: str, default_yes: bool=True, default_no: bool=False):
125     if default_yes:
126 -         response: str = input(question + " (Y/n)")
127     elif default_no:
128 -         response: str = input(question + " (y/N)")
129     else:
130 -         response: str = input(question + " (y/n)")
131     if response == "":
132         if default_yes:
133             return True
134         if default_no:
135             return False
136     response = response.lower()
137     if response == "y":
138         return True
139     if response == "n":
140         return False
141     print("Invalid response")
142     return confirm(question, default_yes, default_no)

124 def confirm(question: str, default_yes: bool=True, default_no: bool=False):
125     if default_yes:
126 +         response: str = input(question + " (Y/n) ")
127     elif default_no:
128 +         response: str = input(question + " (y/N) ")
129     else:
130 +         response: str = input(question + " (y/n) ")
131     if response == "":
132         if default_yes:
133             return True
134         if default_no:
135             return False
136     response = response.lower()
137     if response == "y":
138         return True
139     if response == "n":
140         return False
141     print("Invalid response")
142     return confirm(question, default_yes, default_no)
```


Debugging

- **Supports breakpoints, stepping actions, variable and expression watch, call stack analysis and debugging console**
- **Built-in support for Node. Other languages supported through extensions and system packages (e.g. C++ through extension and gdb)**
- **Debug session configuration (e.g. executable path, build commands) through automatically generated launch.json file**

Debugging

The screenshot shows the Visual Studio Code interface with a Python script named `backup.py` being debugged. The code is paused at line 289, which is `if not verify_conf(conf):`. The debugger interface includes several panels:

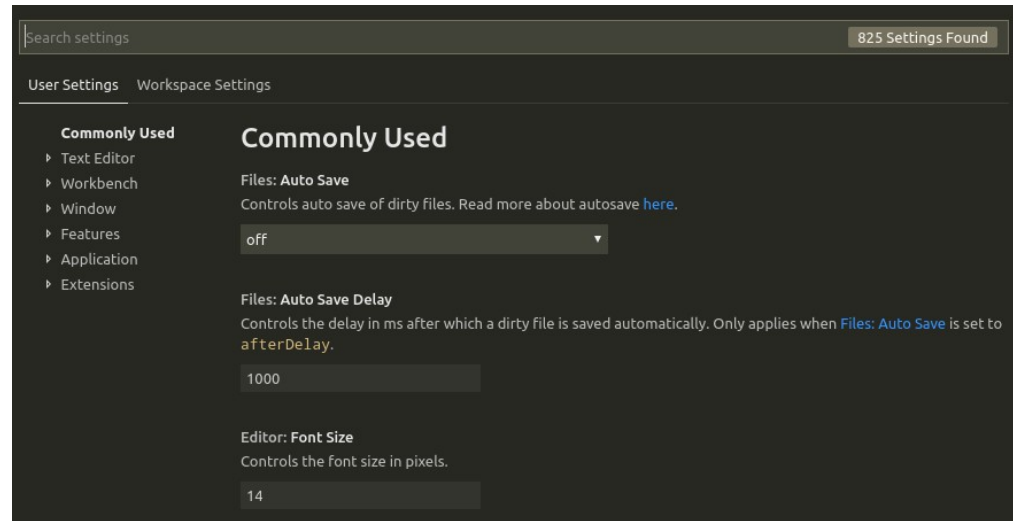
- VARIABLES:** Shows local variables such as `ZipFile`, `backup_dir`, `conf`, `confirm`, `copy`, `datetime`, `differential_backup`, `full_backup`, `gen_config_file`, `hashlib`, `init_logger`, `init_time`, `is_ignored`, and `item_from_path`.
- WATCH:** Shows the state of the `logger` object, including `disabled`, `filters`, `handlers`, `level`, `manager`, `name`, `parent`, `propagate`, and `root`.
- CALL STACK:** Shows the current call stack, with `backup.py` at line 289 being the active frame.
- BREAKPOINTS:** Shows two breakpoints set at lines 289 and 310.

The code in the editor is as follows:

```
277 if __name__ == "__main__":
278     init_logger()
279     logger.debug("Program started")
280     if(not os.path.exists(conf_path)):
281         logger.debug("Config file at %s does not exist" %conf_path)
282         gen_config_file(conf_path)
283         logger.debug("Generated config file at " + conf_path)
284         print("config file created at %s. Please edit it before running this program again." %os.path.abspath(conf_path))
285         os.mkdir("logs")
286         write_log()
287     else:
288         conf: dict = toml.load(conf_path)
289         if not verify_conf(conf):
290             exit(1)
291         if conf["last-full-timestamp"] == 0.0 or conf["current-differential-backups"] >= conf["differential-backups"]:
292             backup_type: int = 0
293         else:
294             backup_type: int = 1
295         if backup_type == 0:
296             response: bool = confirm("The next backup is set to be a full backup. Proceed?")
297             if not response:
298                 if confirm("Perform a differential backup instead?", default_yes=False, default_no=True):
299                     backup_type = 1
300             else:
301                 exit(0)
302         else:
303             response: bool = confirm("The next backup is set to be a differential backup. Proceed?")
304             if not response:
305                 if confirm("Perform a full backup instead?", default_yes=False, default_no=True):
306                     backup_type = 0
307             else:
308                 exit(0)
309         if backup_type == 0:
310             logger.info("Started full backup")
311             now: datetime = datetime.datetime.now()
312             log_destination = full_backup(conf)
313             conf["current-differential-backups"] = 0
314             logger.debug("Reset current-differential-backups")
```

Configuration (demo)

- Configuration is stored in JSON files, but a GUI can also be used to easily edit settings
- Configurations follow a hierarchy:
 - Default settings are overridden by user settings
 - User settings are overridden by workspace settings (which are saved in workspace files)



```
home ▸ william ▸ .config ▸ Code ▸ User ▸ {} settings.json ▸ ...
1  {
2    "gitlens.advanced.messages": {
3      "suppressShowKeyBindingsNotice": true
4    },
5    "gitlens.currentLine.enabled": false,
6    "gitlens.hovers.currentLine.over": "line",
7    "editor.wordWrap": "on",
8    "window.zoomLevel": 0,
9    "window.titleBarStyle": "custom",
10   "workbench.colorTheme": "Monokai",
11   "workbench.startupEditor": "welcomePage",
12   "python.jediEnabled": false,
13   "editor.suggestSelection": "first",
14   "vsintellicode.modify.editor.suggestSelection":
   "automaticallyOverrodeDefaultValue",
15   "workbench.statusBar.feedback.visible": false,
16   "breadcrumbs.enabled": true,
17   "fileheader.authorName": "William Harrell"
18 }
```

Extensions (demo)

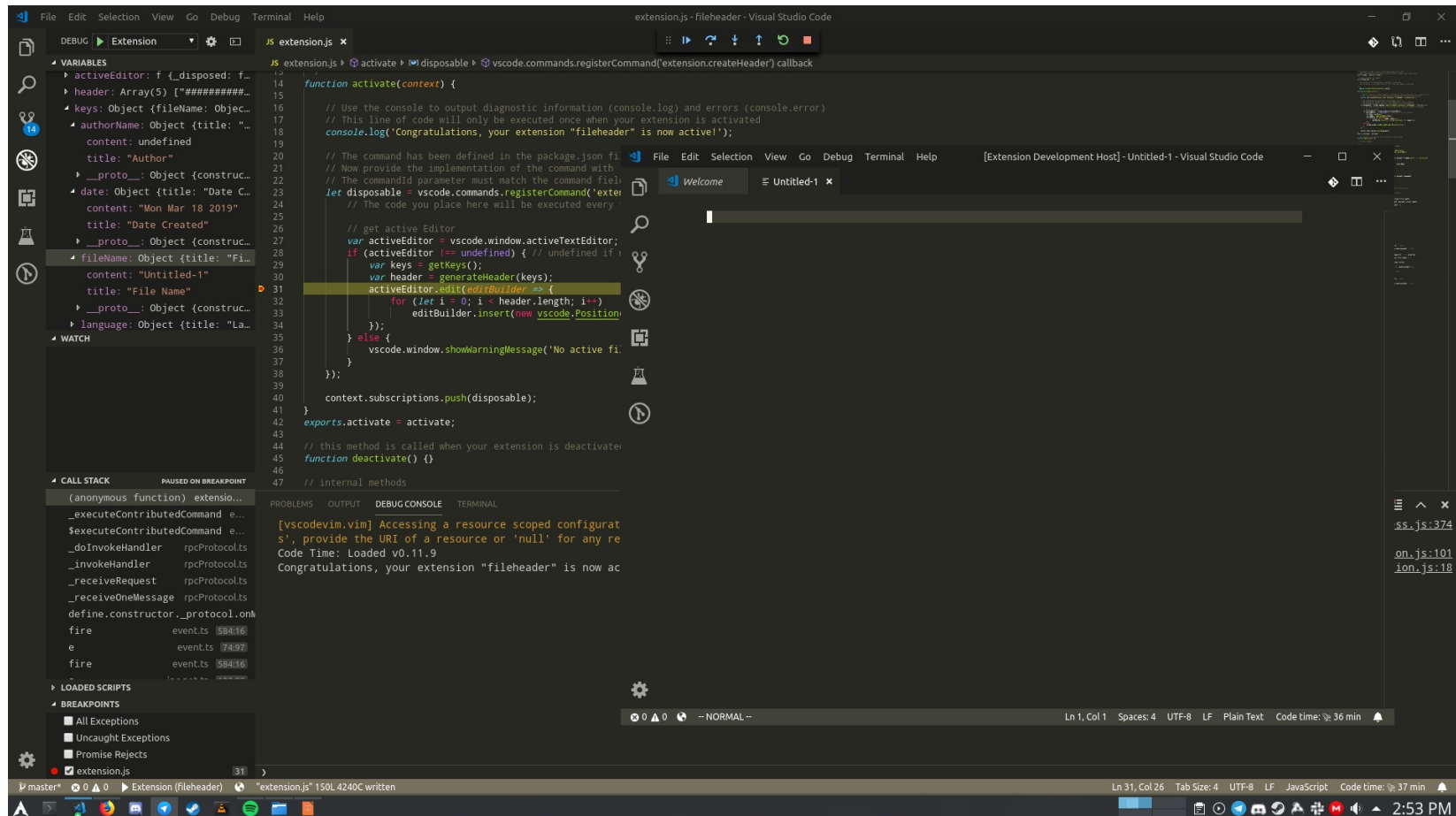
- **Extensions are created using three main components**

- Activation Events: Definitions of when the extension should be activated
- Contribution Points: Descriptions of what exactly the extension adds to VS Code
- VS Code API: Extensive JS API for interacting with the editor

- **Being heavily integrated with Node, extensions can use npm packages as dependencies which will be installed along with the package**

Debugging Extensions (demo)

Extensions can be debugged while they run in another instance of VS Code



Discord Presence

- **Show off to all of your gamer friends**
- **Shows language, filename, workspace, and time elapsed**
 - 15s delay in file changes due to Discord's policies



Excel Viewer

- Preview Excel or CSV data
- Sort by columns

The screenshot displays a software interface for previewing CSV data. The top window shows the raw CSV text with columns: RowNumber, CustomerId, Surname, CreditScore, Geography, Gender, Age, Tenure, Balance, NumOfProducts, HasCrCard, IsActiveMember, EstimatedSalary, Exited. The bottom window shows a table view of the same data with columns: Row Number, Customer Id, Surname, Credit Score, Geography, Gender, Age, Tenure, Balance, Num Of Prod, Has C. The table is sorted by Row Number.

Row Number	Customer Id	Surname	Credit Score	Geography	Gender	Age	Tenure	Balance	Num Of Prod	Has C
1	15634602	Hargrave	619	France	Female	42	2	0	1	
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	
3	15619304	Onio	502	France	Female	42	8	159660.8	3	
4	15701354	Boni	699	France	Female	39	1	0	2	
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	
6	15574012	Chu	645	Spain	Male	44	8	113755.78	2	
7	15592531	Bartlett	822	France	Male	50	7	0	2	
8	15656148	Obinna	376	Germany	Female	29	4	115046.74	4	
9	15792365	He	501	France	Male	44	4	142051.07	2	
10	15592389	H7	684	France	Male	27	2	134603.88	1	
11	15767821	Bearce	528	France	Male	31	6	102016.72	2	
12	15737173	Andrews	497	Spain	Male	24	3	0	2	
13	15632264	Kay	476	France	Female	34	10	0	2	
14	15691483	Chin	549	France	Female	25	5	0	2	
15	15600882	Scott	635	Spain	Female	35	7	0	2	
16	15643966	Goforth	616	Germany	Male	45	3	143129.41	2	
17	15737452	Romeo	653	Germany	Male	58	1	132602.88	1	
18	15788218	Henderson	549	Spain	Female	24	9	0	2	
19	15661507	Muldrow	587	Spain	Male	45	6	0	1	
20	1568982	Hao	726	France	Female	24	6	0	2	

Git History

• Explore history of a Git project

The screenshot displays the Visual Studio Code interface with the Git History extension active. The main window shows a list of commits for the 'smileforthecamera' repository. The commits are listed in chronological order from top to bottom, with the most recent at the top. Each commit entry includes a commit icon, the commit message, the author's name, the date and time, and a commit hash. The current commit is highlighted in green.

Commit Message	Author	Date	Hash
added blog placeholder	William Harrell	Mon, Mar 18, 2019, 11:19 AM	ac23456
added projects placeholder	William Harrell	Mon, Mar 18, 2019, 9:35 AM	5e03380
added memes placeholder	William Harrell	Mon, Mar 18, 2019, 9:14 AM	3ecb6a7
updated contact	William Harrell	Mon, Mar 18, 2019, 9:06 AM	8401749
added contact placeholder	William Harrell	Sun, Mar 17, 2019, 11:25 PM	e49be9a
added photography placeholder	William Harrell	Sun, Mar 17, 2019, 10:41 PM	6c39997
updated about section	William Harrell	Sun, Mar 17, 2019, 10:12 PM	871ddf a
added about me	William Harrell	Sun, Mar 17, 2019, 12:09 AM	019e6d2
index page added	William Harrell	Sat, Mar 16, 2019, 1:14 PM	27e4a20

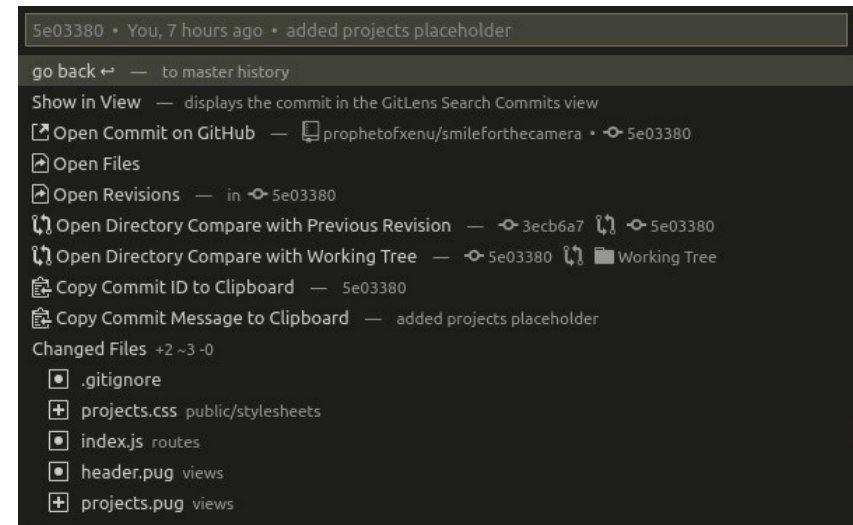
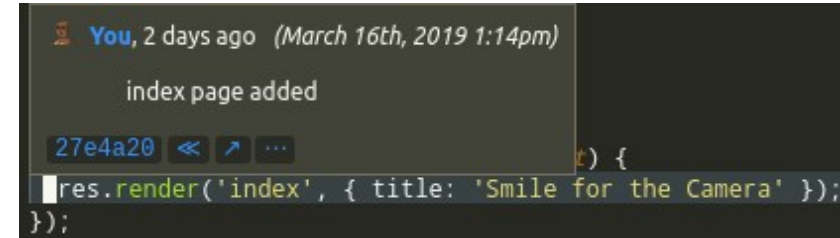
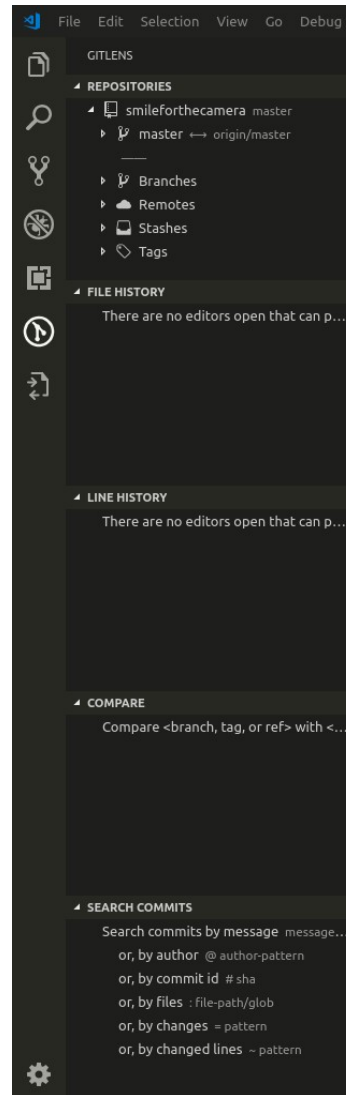
The bottom panel shows a diff view for the selected commit 'added blog placeholder'. It lists the files that were added or modified:

- 3 lines added to public/stylesheets/blog.css
- 4 lines added to routes/index.js
- 11 lines added to views/blog.pug
- 2 lines added to views/header.pug

The left sidebar shows the GitLens view with the 'FILE HISTORY' section selected. The bottom status bar shows the current branch as 'master' and the file encoding as 'UTF-8'.

GitLens (demo)

- “Supercharge” VS Code’s Git functionalities
- In-editor commit details
- Numerous blame annotations
- Highly configurable



Live Share (demo)

- **Collaborate with other users**
- **Only one user needs to have project files**
- **Share editors, terminals, and server applications**
- **Chat through text or voice**

sftp (demo)

- Easily sync between local folder and a remote server using sftp
- Sync whole folders or individual files
- Auto-sync on local change

```
>sftp
SFTP: Open SSH in Terminal
SFTP: Upload Active File
SFTP: Sync Local -> Remote
SFTP: Config
SFTP: List
SFTP: Diff Active File with Remote
SFTP: Cancel All Transfers
SFTP: Download Active File
SFTP: Download Active Folder
SFTP: Download Project
SFTP: List Active Folder
SFTP: List All
SFTP: Set Profile
SFTP: Sync Both Directions
SFTP: Sync Remote -> Local
SFTP: Upload Active Folder
SFTP: Upload Project
```

```
smileforthecamera ▸ .vscode ▸ {} sftp.json ▸ ...
1  {
2      "name": "neon: xenu/smileforthecamera",
3      "host": "smileforthe.camera",
4      "protocol": "sftp",
5      "port": 22,
6      "username": "xenu",
7      "remotePath": "/home/xenu/smileforthecamera",
8      "uploadOnSave": true
9  }
```

Vim (demo)

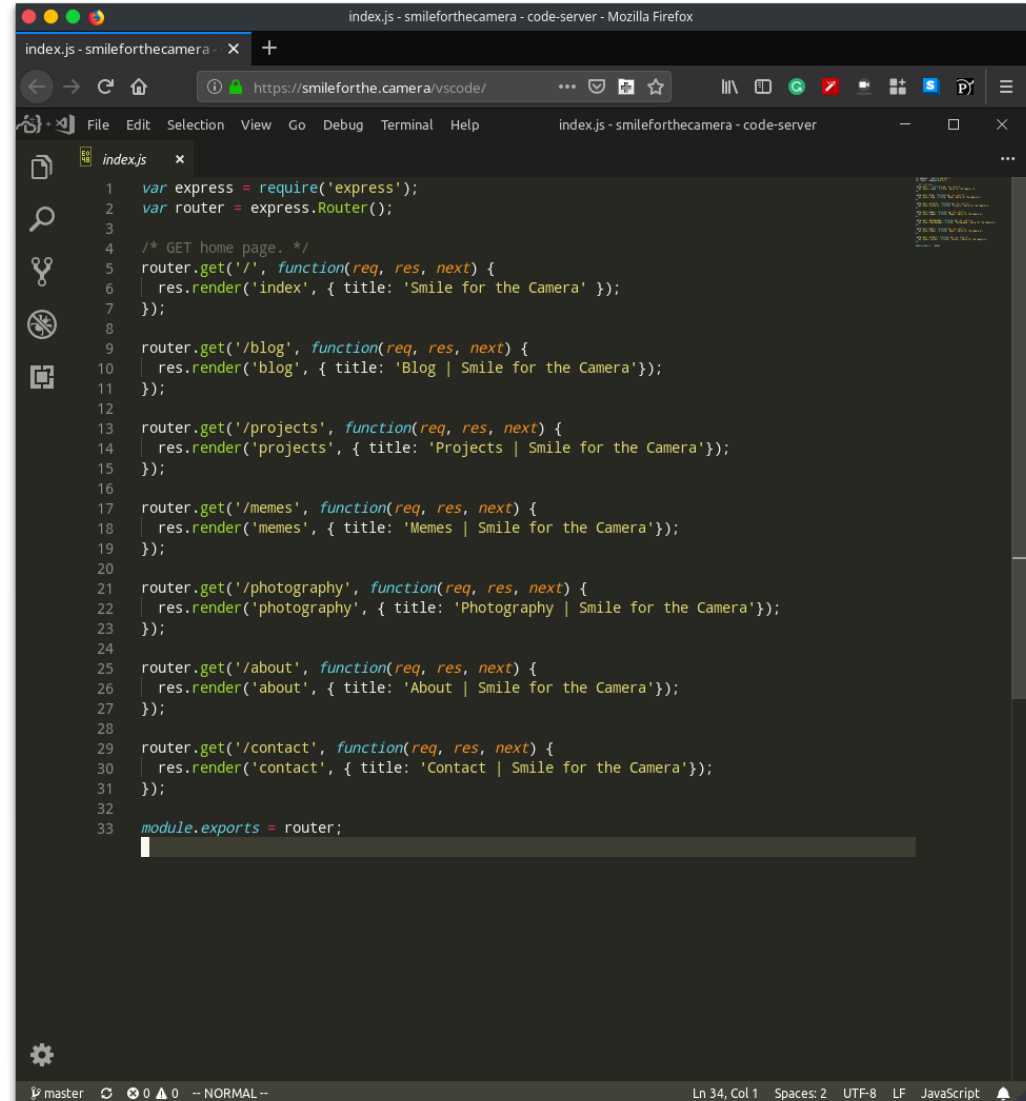
- **Use Vim motions in VS Code**

- Movement
- Insert Mode
- Visual Mode
- Search and Replace
- Range Selection

- **Does not support vimscript although many popular Vim extensions emulate**

code-server

- Fork of VS Code that can be run in a web browser
- Run on a server to access your IDE from any web browser
- Supports extensions



```
index.js - smileforthecamera - code-server - Mozilla Firefox
https://smileforthe.camera/vscode/
index.js - smileforthecamera - code-server
index.js
1  var express = require('express');
2  var router = express.Router();
3
4  /* GET home page. */
5  router.get('/', function(req, res, next) {
6    res.render('index', { title: 'Smile for the Camera' });
7  });
8
9  router.get('/blog', function(req, res, next) {
10   res.render('blog', { title: 'Blog | Smile for the Camera' });
11 });
12
13 router.get('/projects', function(req, res, next) {
14   res.render('projects', { title: 'Projects | Smile for the Camera' });
15 });
16
17 router.get('/memes', function(req, res, next) {
18   res.render('memes', { title: 'Memes | Smile for the Camera' });
19 });
20
21 router.get('/photography', function(req, res, next) {
22   res.render('photography', { title: 'Photography | Smile for the Camera' });
23 });
24
25 router.get('/about', function(req, res, next) {
26   res.render('about', { title: 'About | Smile for the Camera' });
27 });
28
29 router.get('/contact', function(req, res, next) {
30   res.render('contact', { title: 'Contact | Smile for the Camera' });
31 });
32
33 module.exports = router;
```