


Теперь доступна версия 1.84 (/updates)! Читайте о новых функциях и исправлениях с октября.



Topics

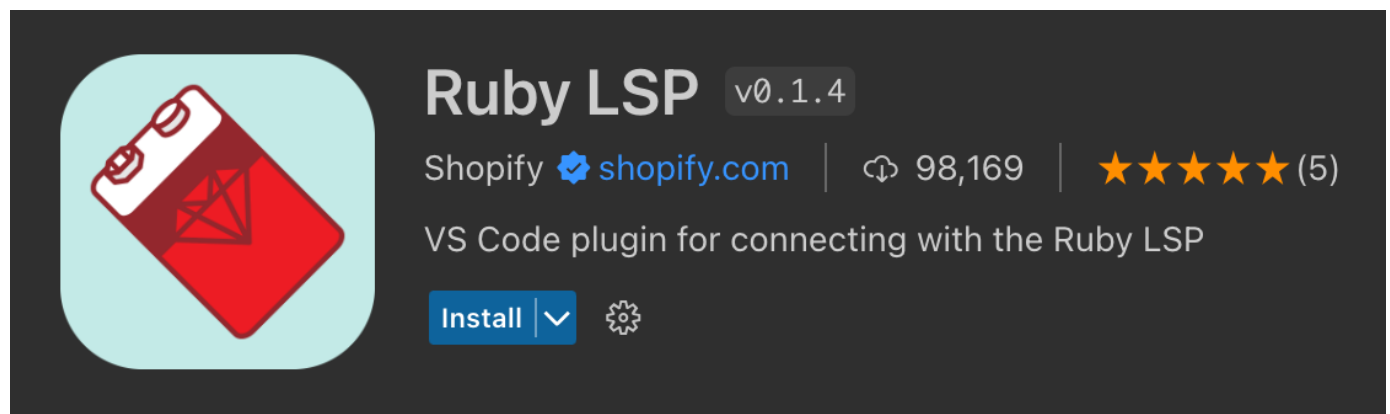
In this article

 (<https://vscode.dev/github/microsoft/vscode-docs/blob/main/docs/languages/ruby.md>)




## Ruby в коде Visual Studio

Ruby (<https://www.ruby-lang.org>) - динамичный язык программирования с открытым исходным кодом, известный своей простотой и производительностью. Благодаря выразительному и элегантному синтаксису, часть философии Ruby заключается в том, чтобы радовать разработчиков. Он часто используется для веб-разработки с использованием различных фреймворков и для написания сценариев, что позволяет выполнять быстрые итерации при создании прототипов.



В этом разделе подробно рассказывается о настройке и использовании Ruby в коде Visual Studio с расширением Ruby LSP (<https://marketplace.visualstudio.com/items?itemName=Shopify.ruby-lsp>).



**Ruby LSP** v0.1.4

Shopify  [shopify.com](https://shopify.com) |  98,169 |  (5)

VS Code plugin for connecting with the Ruby LSP

[Install](#)  

## Установка

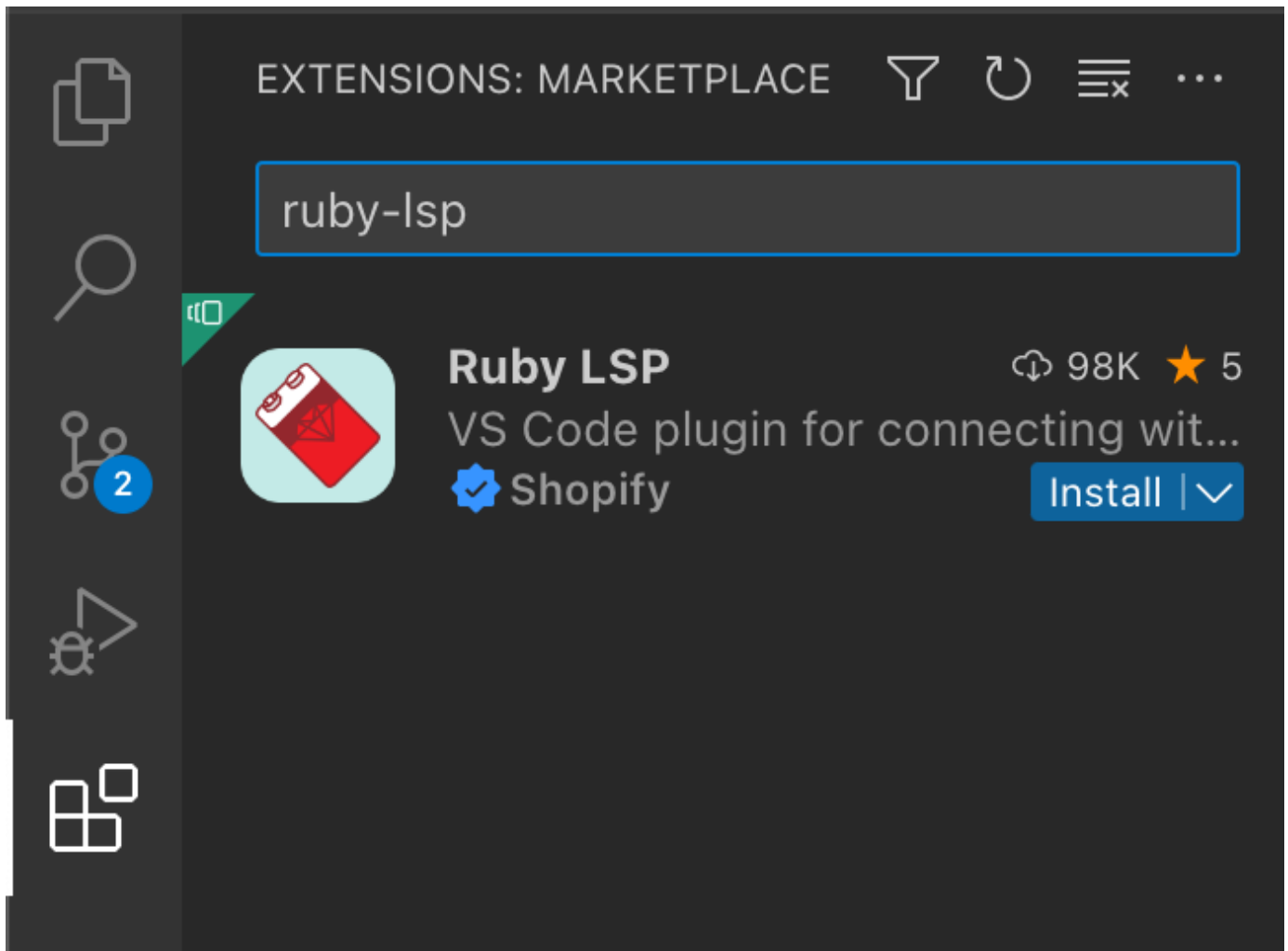
### Установите Ruby через диспетчер версий

Хотя Ruby установлен по умолчанию в некоторых операционных системах (таких как macOS и некоторые дистрибутивы Linux), мы рекомендуем использовать менеджер версий, такой как `rbenv` (<https://github.com/rbenv/rbenv>), чтобы иметь возможность доступа к более новым версиям Ruby в Windows, macOS и Linux. Следуйте руководству по установке (<https://github.com/rbenv/rbenv#installation>) для вашей платформы.

Примечание: Как и при установке любого нового набора инструментов на вашем компьютере, вам нужно обязательно перезапустить экземпляры терминала / командной строки и VS Code, чтобы использовать расположение обновленного набора инструментов в переменной PATH вашей платформы.

## Установите расширение Ruby LSP в VS Code

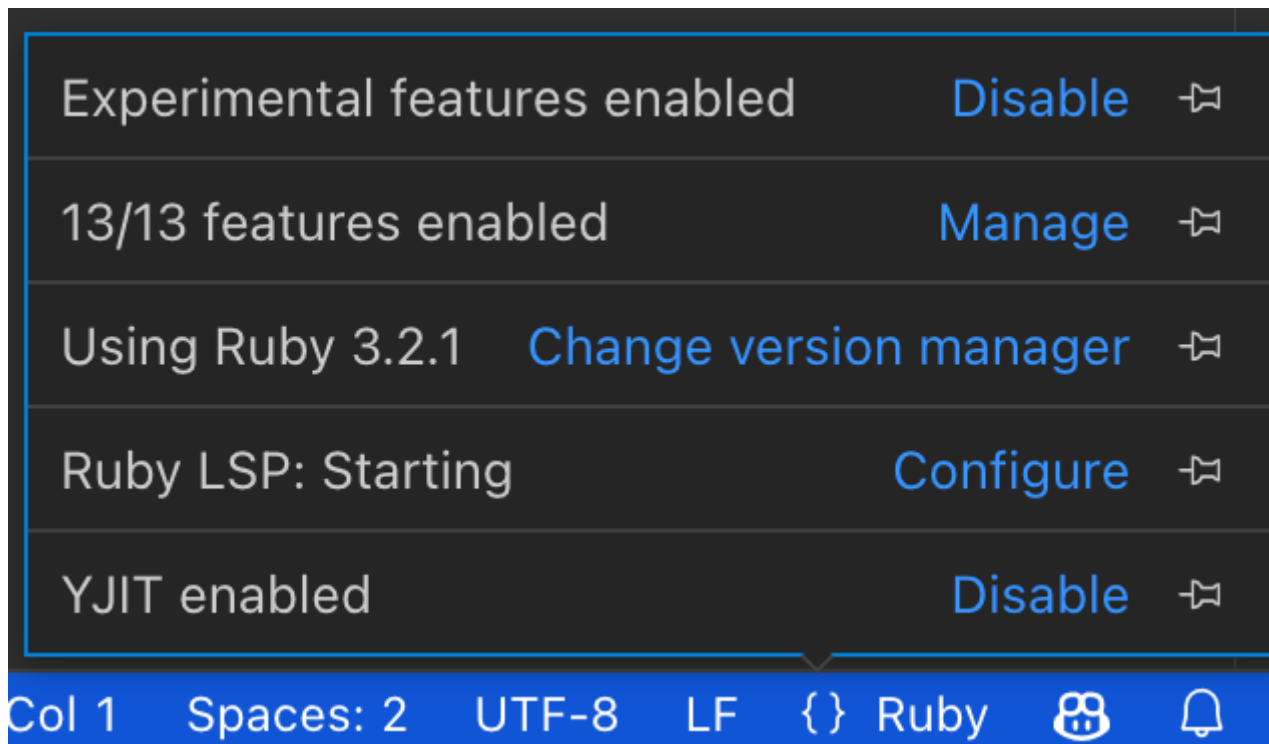
Вы можете найти и установить расширение Ruby LSP из VS Code через представление расширений (Ctrl + Shift + X) и выполнить поиск "Ruby LSP".



В этом разделе мы обсудим многие функции Ruby LSP, но вы также можете обратиться к документации по расширению по адресу <https://github.com/Shopify/vscode-ruby-lsp> (<https://github.com/Shopify/vscode-ruby-lsp>).

## Проверьте правильность установки

После установки проверьте пункт "Статус языка", чтобы увидеть статус сервера Ruby LSP. Если диспетчер версий настроен, он должен отображать подходящую версию Ruby для вашего проекта. Состояние сервера должно отображать запуск или running, но не error.



Расширение автоматически создает `.ruby-lsp` папку с пользовательским пакетом, который включает в себя языковой сервер `gem ruby-lsp`. Настройка не требуется.

По умолчанию расширение пытается автоматически определить используемый вами Ruby Version manager и соответствующим образом использовать правильные версии и пути. Если вы хотите настроить такое поведение, задайте следующую конфигурацию в своих пользовательских настройках (</docs/getstarted/settings>):

```
"rubyLsp.rubyVersionManager": "rbenv"
```

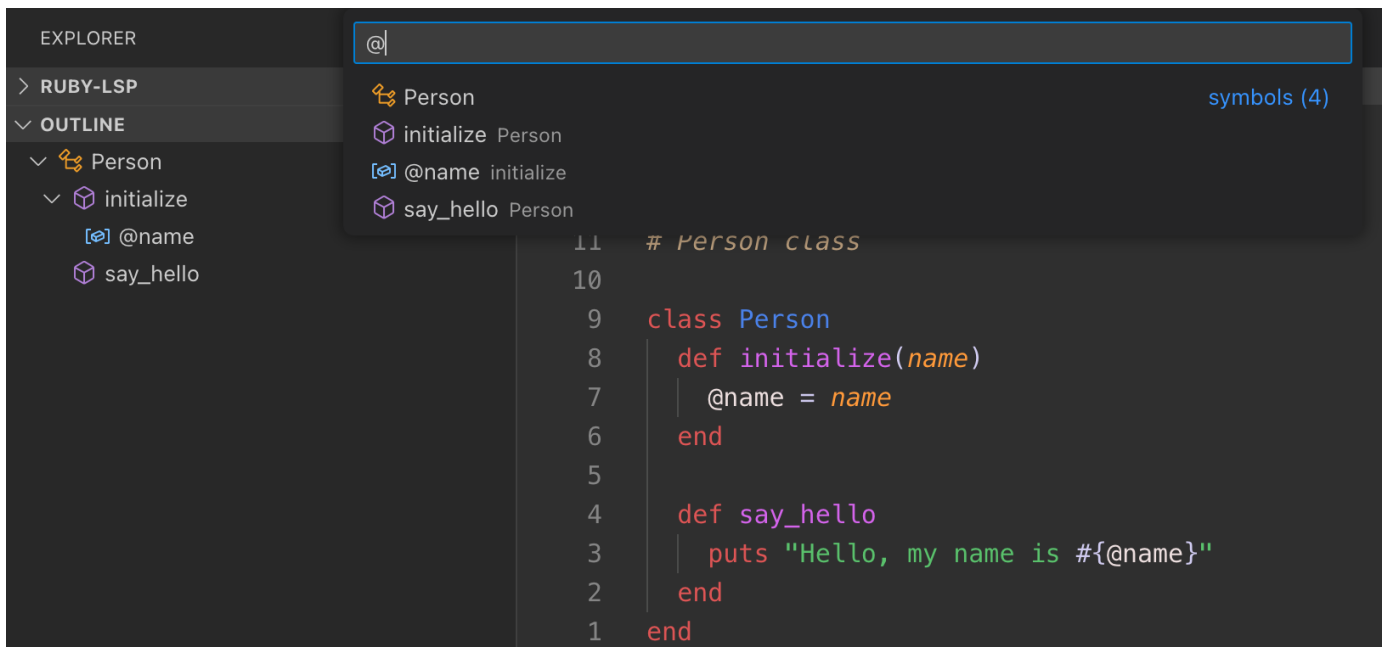
Расширение будет автоматически пытаться обновлять драгоценный камень `ruby-lsp` языкового сервера один раз в день; если вы хотите принудительно это сделать, используйте палитру команд (`Ctrl + Shift + P`), чтобы выполнить Ruby LSP: обновить драгоценный камень языкового сервера.

Если у вас возникли какие-либо проблемы, следующие шаги см. в разделе "Устранение неполадок" (<https://github.com/Shopify/vscode-ruby-lsp#troubleshooting>).

## Основные возможности

### Обнаружение символов

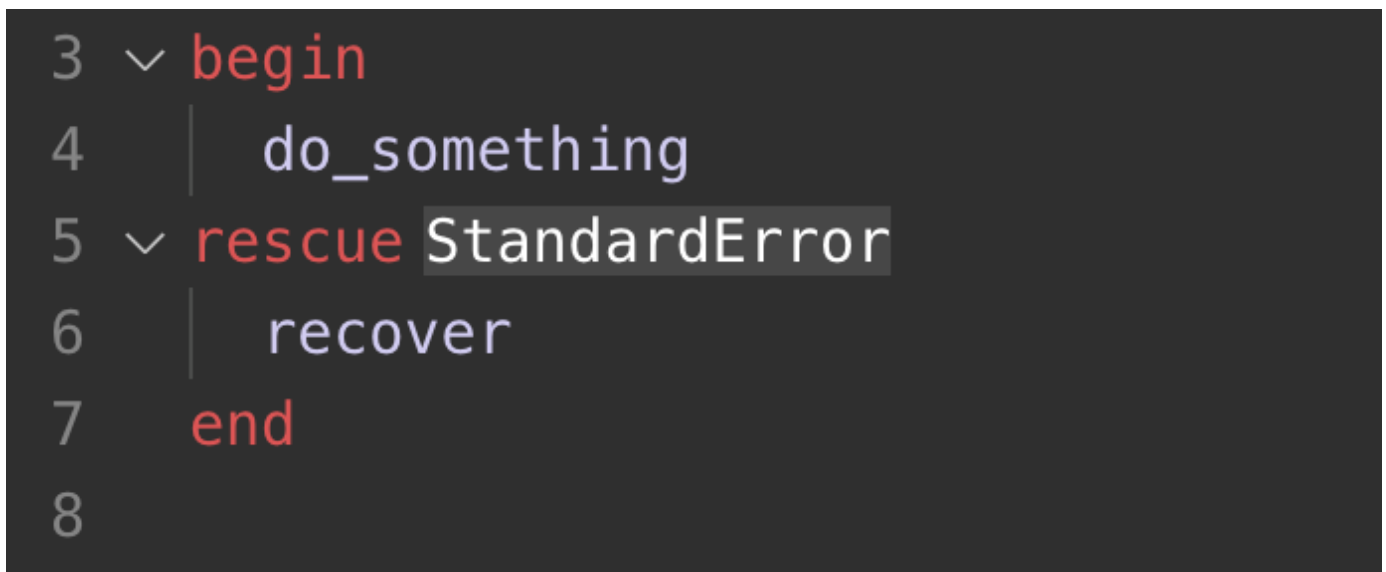
Ruby LSP анализирует исходный код и обеспечивает обнаружение символов и навигацию по панели структуры, макетам файлов и поиску символов (по умолчанию доступно с помощью `Ctrl + Shift + O`).



Чтобы узнать больше о быстром перемещении по исходному коду с помощью VS Code, ознакомьтесь с навигацией по коду (</docs/editor/editingevolved>).

### Подсказки по инкрустации

Ruby LSP способен отображать полезную информацию о предполагаемых или неявных значениях в вашем коде. В приведенном ниже примере вы можете видеть, что `StandardError` отображается как класс неявного исключения, который сохраняется при пустом `rescue` вызове.



Хотя подсказки по инкрустации могут быть полезны для понимания вашего кода, вы также можете отключить эту функцию с помощью редактора> "Подсказки по инкрустации: включенный параметр" (`editor.inlayHints.enabled`) или использовать следующее, чтобы отключить эту функцию только для Ruby LSP:

```

"rubyLsp.enabledFeatures": {
  "inlayHint": false,
}

```

## Фрагменты кода

При вводе файла Ruby LSP Ruby может предложить расширить фрагменты, используемые для обычных операций Ruby, таких как создание нового метода, класса, блока или шаблона тестирования. Чтобы ознакомиться с полным списком, ознакомьтесь с файлом `snippets` в репозитории GitHub (<https://github.com/Shopify/vscode-ruby-lsp/blob/main/snippets.json>) Ruby LSP.

## Семантическая подсветка синтаксиса

Ruby LSP способен использовать подсветку семантического синтаксиса (<https://github.com/microsoft/vscode/wiki/Semantic-Highlighting-Overview>) и стилизацию благодаря глубокому пониманию исходного кода проекта.

Например, он может выделять:

- Вызовы методов выполняются последовательно, не путая их с локальными переменными.
- Локальные аргументы (такие как аргументы метода, блока или лямбда) последовательно находятся внутри области, в которой они существуют.

```
3  def process(item)
4    description = "Processing #{item.name}"
5    item.process(description)
6    lock
7    item
8  end
```

Примечание: На этом скриншоте используется тема Spinel, включенная в пакет расширений Ruby (<https://marketplace.visualstudio.com/items?itemName=Shopify.ruby-extensions-pack>). Темы должны использовать информацию, предоставляемую Ruby LSP, чтобы обеспечить насыщенное выделение файлов Ruby.

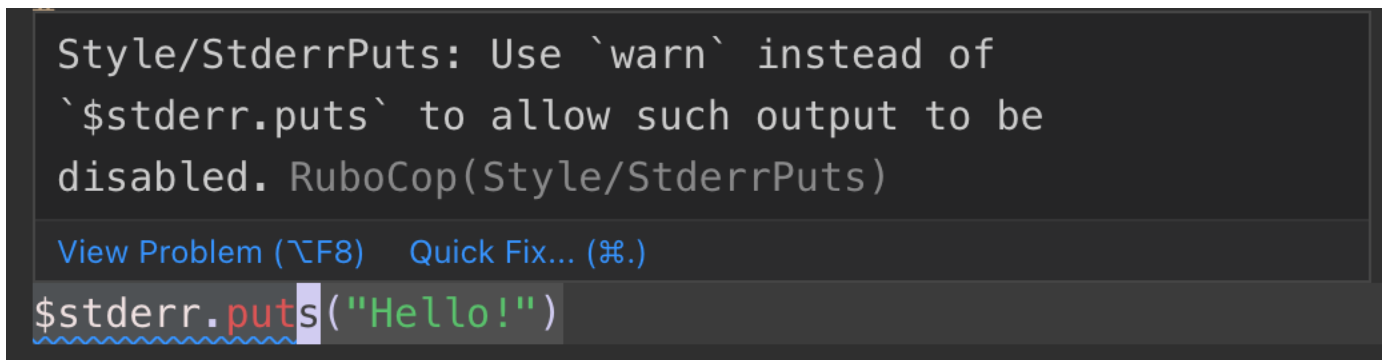
Чтобы использовать эту функцию, в редакторе должна быть включена семантическая подсветка.

```
"editor.semanticHighlighting.enabled": true,
```

## Компоновка и форматирование

По умолчанию Ruby LSP обеспечивает компоновку и форматирование благодаря интеграции с RuboCop (<https://github.com/rubocop/rubocop>). Вы можете отформатировать свой файл Ruby с помощью `Ctrl + Shift + I` или выполнив команду Форматировать документ из палитры команд (`Ctrl + Shift + P`) или контекстного меню редактора.

Если ваш проект не использует RuboCop, Ruby LSP будет форматировать файлы с использованием SyntaxTree ([https://ruby-syntax-tree.github.io/syntax\\_tree](https://ruby-syntax-tree.github.io/syntax_tree)).



Вы также можете запускать программу форматирования при каждом сохранении (редактор: Форматирование при сохранении), чтобы ваш Ruby-код автоматически сохранял правильное форматирование во время работы. Для этого вы должны включить форматирование при сохранении.

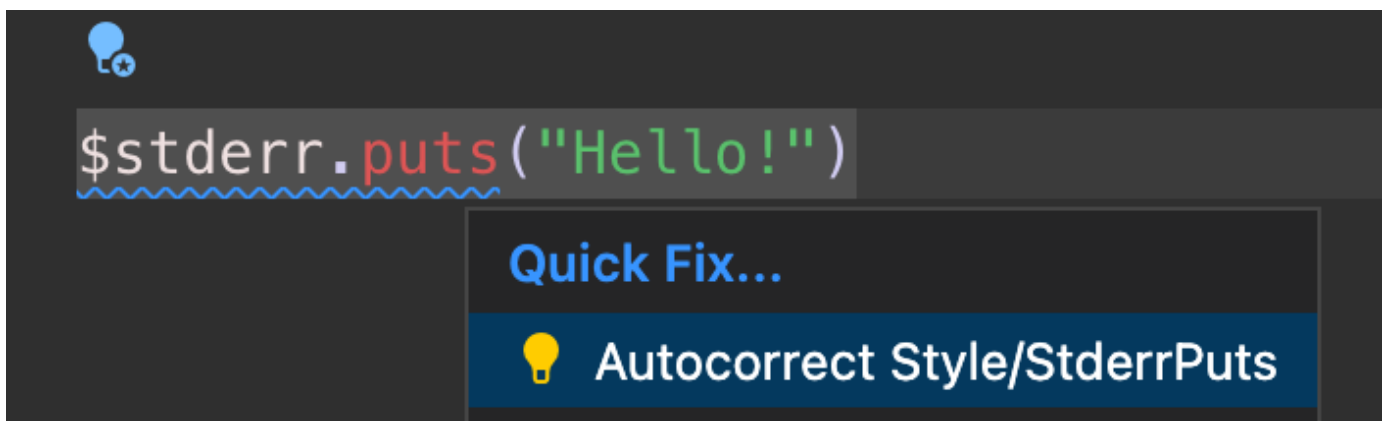
```
"editor.formatOnSave": true
```

Расширение Ruby LSP также предоставляет некоторые удобные дополнения с использованием зависимости формата от типа. Например, оно автоматически продолжит строки комментариев и автоматически закроет `end` токены, каналы или фигурные скобки для интерполяции строк. Чтобы использовать форматирование для типа, убедитесь, что он включен в редакторе с помощью:

```
"editor.formatOnType": true
```

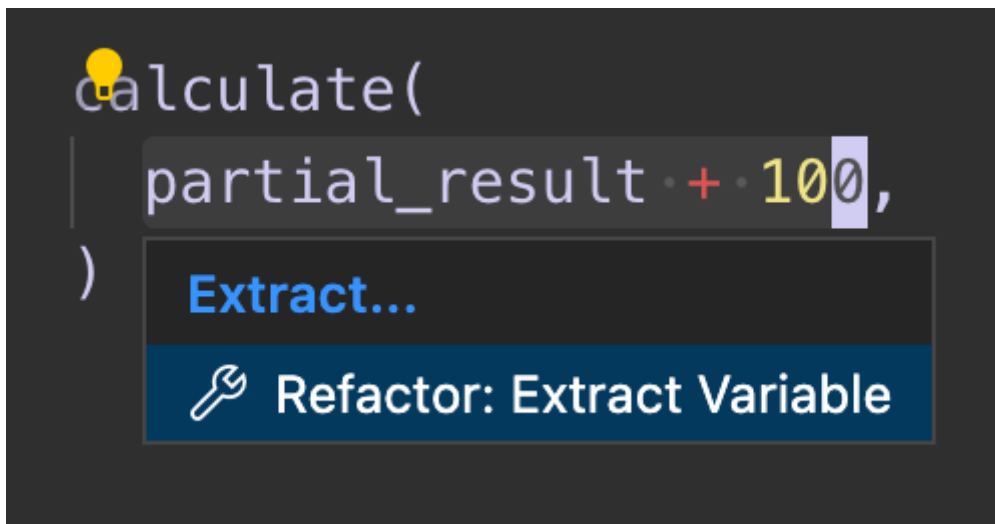
## Быстрые исправления

Когда линтер обнаруживает ошибки и предупреждения в вашем исходном коде, Ruby LSP часто может предложить быстрые исправления (также называемые действиями кода), которые доступны при наведении курсора мыши на лампочку в редакторе. Вы можете быстро открыть доступные быстрые исправления с помощью `Ctrl + .`.



## Рефакторинг

В дополнение к быстрым исправлениям, Ruby LSP также предоставляет параметры рефакторинга с помощью действий с кодом. Например, он может извлечь выражение Ruby в локальную переменную одним щелчком мыши.



## Debugging support with rdbg

For debug support inside of VS Code, the Ruby LSP requires the VS Code RDBG extension to connect to debug (Ruby's official debugger).

### Install debugging support

Install the VS Code RDBG (<https://marketplace.visualstudio.com/items?itemName=KoichiSasada.vscode-rdbg>) extension.

### Set up debugging configuration

To use the debugger, you will need to create debugging configurations ([/docs/editor/debugging#\\_launch-configurations](/docs/editor/debugging#_launch-configurations)) in a `launch.json` file. The configuration lets you pass arguments to your program, run pre-launch tasks, set environment variables, and much more.

To create a `launch.json` for a Ruby program:

1. In the Debug view (Ctrl+Shift+D), select the create a `launch.json` file link.
2. This displays a dropdown with several default launch configuration types. You can pick the first option, but we'll add more configurations.
3. We can now edit the created `.vscode/launch.json` file to add more ways to launch your Ruby program for debugging.

Example:

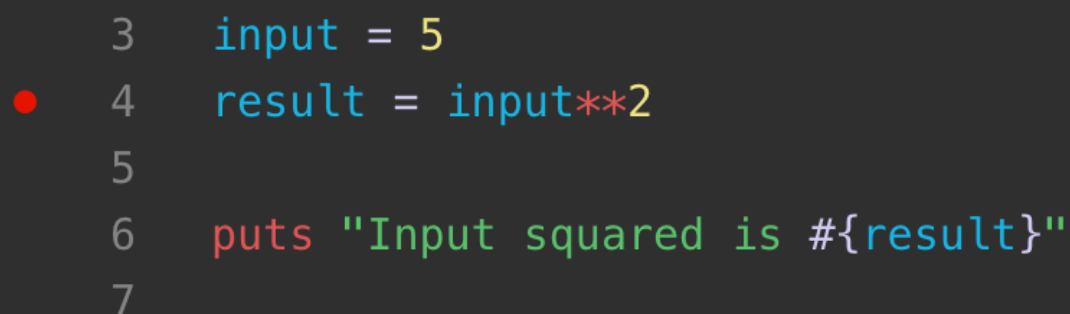
```

{
  "version": "0.2.0",
  "configurations": [
    // Run all tests in a file using Minitest
    {
      "type": "rdbg",
      "name": "Minitest - current file",
      "request": "launch",
      "script": "-Itest ${file}",
      "askParameters": false
    },
    // If your test runner supports line numbers, such as in Rails,
    // you can add a task like this one to run only the test under the cursor
    {
      "name": "Minitest - current line",
      "type": "rdbg",
      "request": "launch",
      "command": "${workspaceRoot}/bin/rails",
      "script": "test",
      "args": ["${file}:${lineNumber}"],
      "askParameters": false
    },
    // Attach the debugger to an active process (for example, Rails server)
    {
      "type": "rdbg",
      "name": "Attach with rdbg",
      "request": "attach"
    }
  ]
}

```

After adding the launch configurations, we can debug Ruby programs by adding breakpoints and executing our launch tasks.

1. Open a Ruby file and click the left gutter in the editor to set a break point. It should display as a red dot.



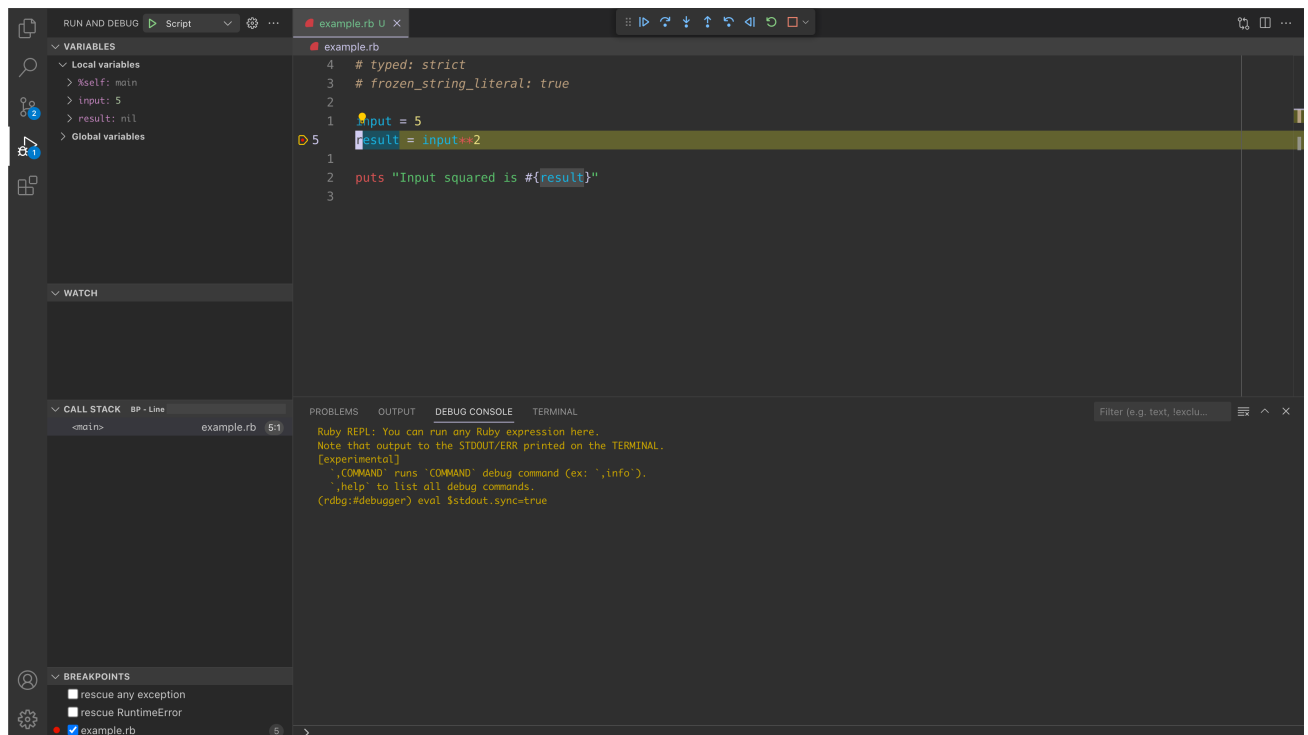
```

3   input = 5
●  4   result = input**2
5
6   puts "Input squared is #{result}"
7

```

2. Start debugging by selecting the desired task under Run and Debug and clicking the start debugging button (default keyboard shortcut F5).





## Next steps

This has been a brief overview showing the Ruby LSP extension features within VS Code. For more information, see the details provided in the Ruby LSP README (<https://github.com/Shopify/vscode-ruby-lsp>), including how to tune specific VS Code editor (<https://github.com/Shopify/vscode-ruby-lsp#configuration>) configurations.

To stay up to date on the latest features/bug fixes for the Ruby LSP extension, see the Releases page for the language server gem (<https://github.com/Shopify/ruby-lsp/releases>) and for the Ruby LSP extension (<https://github.com/Shopify/vscode-ruby-lsp/releases>).

If you have any issues or feature requests, feel free to log them in the Ruby LSP extension GitHub repo (<https://github.com/Shopify/vscode-ruby-lsp/issues>).







If you'd like to learn more about VS Code, try these topics:

- Basic Editing (/docs/editor/codebasics) - A quick introduction to the basics of the VS Code editor.
- Install an Extension (/docs/editor/extension-marketplace) - Learn about other extensions are available in the Marketplace (<https://marketplace.visualstudio.com/vscode>).
- Code Navigation (/docs/editor/editingevolved) - Move quickly through your source code.

Was this documentation helpful?

Yes

No

-  [Subscribe\(/feed.xml\)](/feed.xml)
-  [Ask questions\(https://stackoverflow.com/questions/tagged/vscode\)](https://stackoverflow.com/questions/tagged/vscode)
-  [Follow @code\(https://go.microsoft.com/fwlink/?LinkID=533687\)](https://go.microsoft.com/fwlink/?LinkID=533687)
-  [Request features\(https://go.microsoft.com/fwlink/?LinkID=533482\)](https://go.microsoft.com/fwlink/?LinkID=533482)
-  [Report issues\(https://www.github.com/Microsoft/vscode/issues\)](https://www.github.com/Microsoft/vscode/issues)
-  [Watch videos\(https://www.youtube.com/channel/UCs5Y5\\_7XK8HLDX0SLNwkd3w\)](https://www.youtube.com/channel/UCs5Y5_7XK8HLDX0SLNwkd3w)


Hello from Seattle.    [Follow @code \(https://go.microsoft.com/fwlink/?LinkID=533687\)](https://go.microsoft.com/fwlink/?LinkID=533687)

Star 153,039

[Support \(https://support.serviceshub.microsoft.com/supportforbusiness/create?sapId=d66407ed-3967-b000-4cfb-2c318cad363d\)](https://support.serviceshub.microsoft.com/supportforbusiness/create?sapId=d66407ed-3967-b000-4cfb-2c318cad363d)

[Privacy \(https://go.microsoft.com/fwlink/?LinkId=521839\)](https://go.microsoft.com/fwlink/?LinkId=521839)

[Terms of Use \(https://www.microsoft.com/legal/terms-of-use\)](https://www.microsoft.com/legal/terms-of-use)    [License \(/License\)](/License)

 [Microsoft \(https://www.microsoft.com\)](https://www.microsoft.com)

© 2023 Microsoft