



СТИ

НИЯУ МИФИ

Северский
технологический
институт

УЧЕБНАЯ ДИСЦИПЛИНА
ВВЕДЕНИЕ В СПЕЦИАЛЬНОСТЬ

ЛЕКЦИЯ 4
СРЕДА РЕДАКТИРОВАНИЯ КОДА VISUAL STUDIO CODE

лектор: к.т.н., доцент каф. ЭАФУ
Иванов Константин Александрович

2025 г.

- 1. РЕДАКТОРЫ КОДА**
- 2. VISUAL STUDIO CODE**
- 3. ЛИЦЕНЗИЯ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

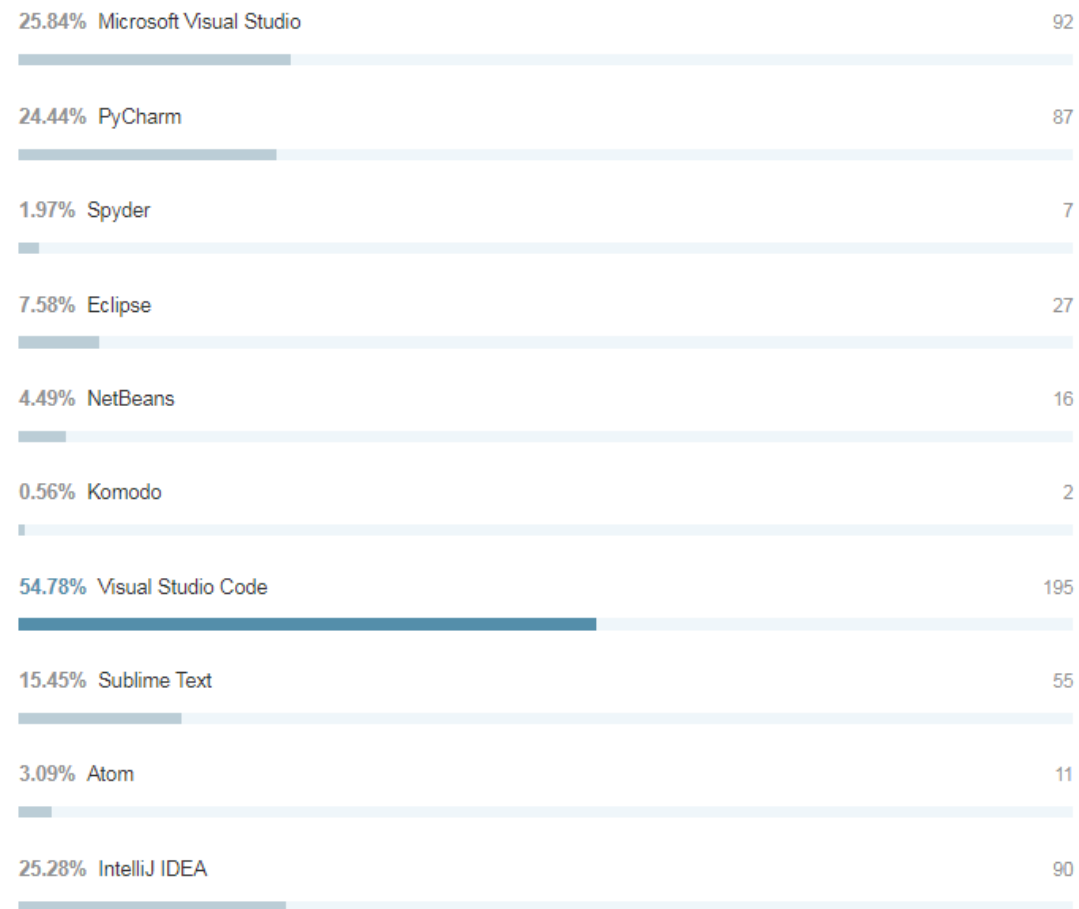
РЕДАКТОРЫ КОДА

<https://practicum.yandex.ru/blog/obzor-redaktorov-koda/>

<https://habr.com/ru/companies/serverspace/articles/693374/>

<https://github.com/collections/text-editors>

Какие инструменты используете вы?



Проголосовали 356 пользователей. Воздержались 64 пользователя.



Интегри́рованная среда разрабо́тки, ИСР (англ. **Integrated development environment — IDE**), также единая среда разработки, ЕСР — комплекс программных средств, используемый программистами для разработки программного обеспечения (ПО).

Среда разработки включает в себя:

- **текстовый редактор,**
- **транслятор** (компилятор и/или интерпретатор),
- **средства автоматизации сборки,**
- **отладчик.**

VISUAL STUDIO CODE

File Edit Selection View Transition Debug Terminal Help PIO Home - test - Visual Studio Code [Администратор]

EXTENSIONS

Поиск расширений в Marketplace

ВКЛЮЧЕНО 7

- C/C++** 0.26.3
C/C++ IntelliSense, debugging, and code bro...
Microsoft
- C#** 1.21.12
C# for Visual Studio Code (powered by Omni...
Microsoft
- C++ Intellisense** 0.2.2
C/C++ Intellisense with the help of GNU Glob...
austin
- PlatformIO IDE** 1.10.0
Development environment for Embedded, IoT...
PlatformIO
- Prettier - Code formatter** 3.20.0
Code formatter using prettier
Esen Petersen
- Python** 2020.2.64397
Linting, Debugging (multi-threaded, remote), ...
Microsoft
- Russian Language Pack for Visual St...** 1.42.2
Language pack extension for Russian
Microsoft

РЕКОМЕНДУЕМОЕ 0

Расширений не найдено.

Home Projects Inspect Libraries Boards Platforms Devices

Welcome to PlatformIO

Home 3.1.0 · Core 4.2.1

Quick Access

- + New Project
- Import Arduino Project
- Open Project
- Project Examples

Recent News

Join us on LinkedIn

Jan 2 PlatformIO
Do you follow @PlatformIO_Org on LinkedIn? Click the link and connect with us today
👉 [linkedin.com/company/platformio/](https://www.linkedin.com/company/platformio/)
💡 #Embedded #OpenSource #Freedom #Neutrality

Feb 18 PlatformIO
Massive update for Atmel SAM dev-platform 4.1.0! 🚀
New boards, updated #Arduino cores, improvements and bugfixes
👉 community.platformio.org/t/atmel-sam-de...

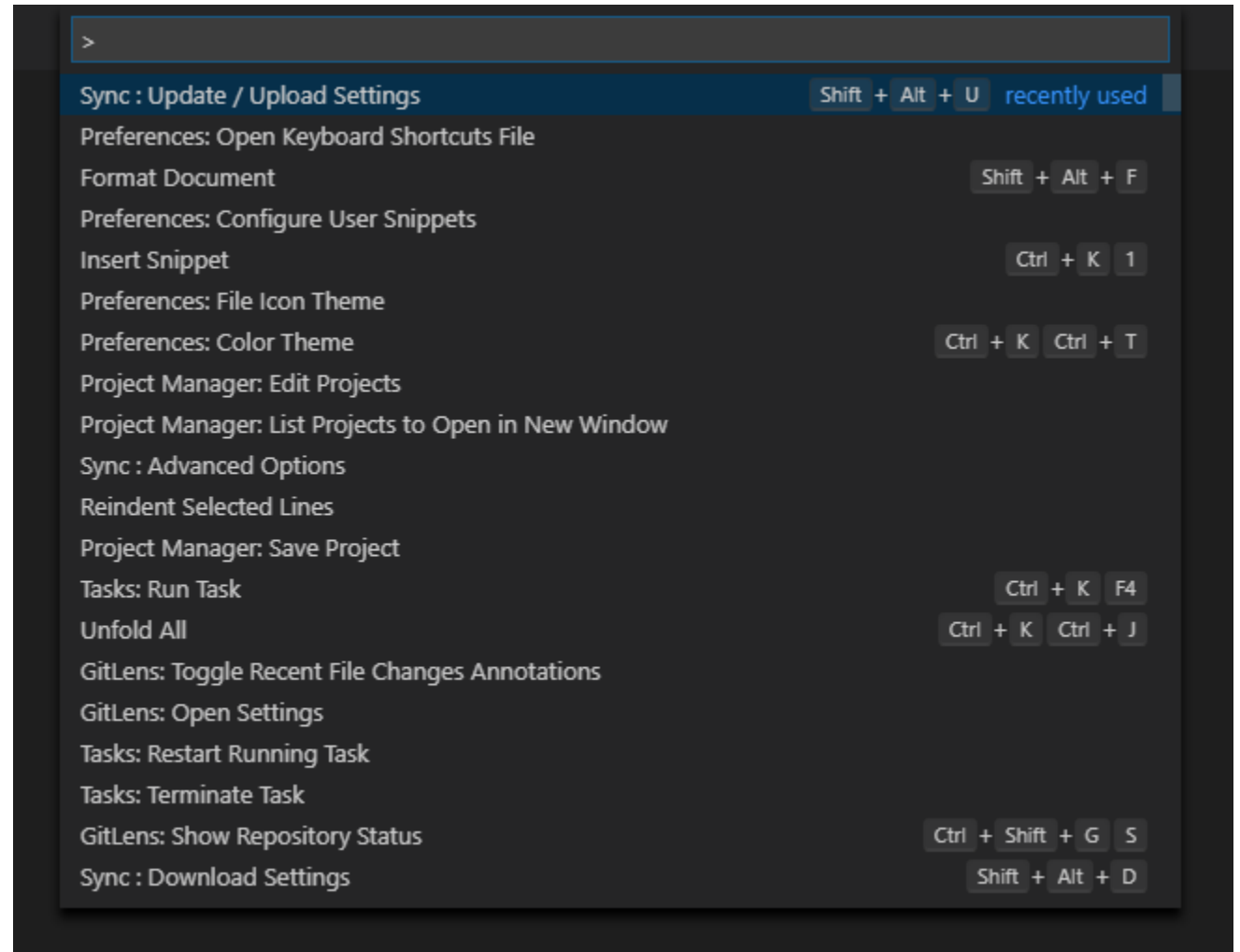
Feb 12 PlatformIO
PlatformIO Core 4.2 and PIO Home 3.1 are out! 🚀
✅ Project Manager and UI for "platformio.ini"
✅ PIO Check for PVS-Studio
✅ Compilation database "compile_commands.json"

ПРОБЛЕМЫ ВЫВОД КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ

1. **Палитра команд.** Это основной интерфейс для общения с редактором.
2. **Управление настройками.** Общий подход к заданию настроек. Прописываем ряд полезных глобальных настроек и решаем вопрос синхронизации между различными рабочими устройствами.
3. **Написание кода.** Инструменты и подходы, позволяющие ускорить набор кода. Использование *Emmet* в *VSCode* и редактирование с *множественным курсором*.
4. **Навигация по проекту.** Преимущества символьного поиска над текстовым в ряде типовых ситуаций.
5. **Работа с проектами.** Подход к управлению проектами с возможностью быстрого переключения между ними.
6. **Встроенный терминал.** Применение интегрированного терминала, как более удобная альтернатива сторонним приложениям.
7. **Система контроля версий.** Встроенные возможности по работе с Git.
8. **Отладка.** Настраиваем отладку клиентского кода на уровне редактора.
9. **Сниппеты.** Общая информация по сниппетам и описание процесса работы с ними в *VSCode*.
10. **Intellisense.** Встроенные возможности автосажджеста и способы увеличения отдачи редактора по написанному вами коду.
11. **Внешние зависимости.** Использование возможностей *VSCode* для упрощения управления версиями подключаемых библиотек.
12. **Ассоциация типов файлов.** Правим распознавание редактором файлов экзотических расширений.
13. **Расширения.** Несколько полезных плагинов, не попавших под остальные категории.

Палитра команд представляет собой командную строку с выводом списка наиболее актуальных команд

Ctrl + Shift + P — вызов палитры команд



Настройки прописываются через json-файлы, но также доступен и пользовательский интерфейс (UI).

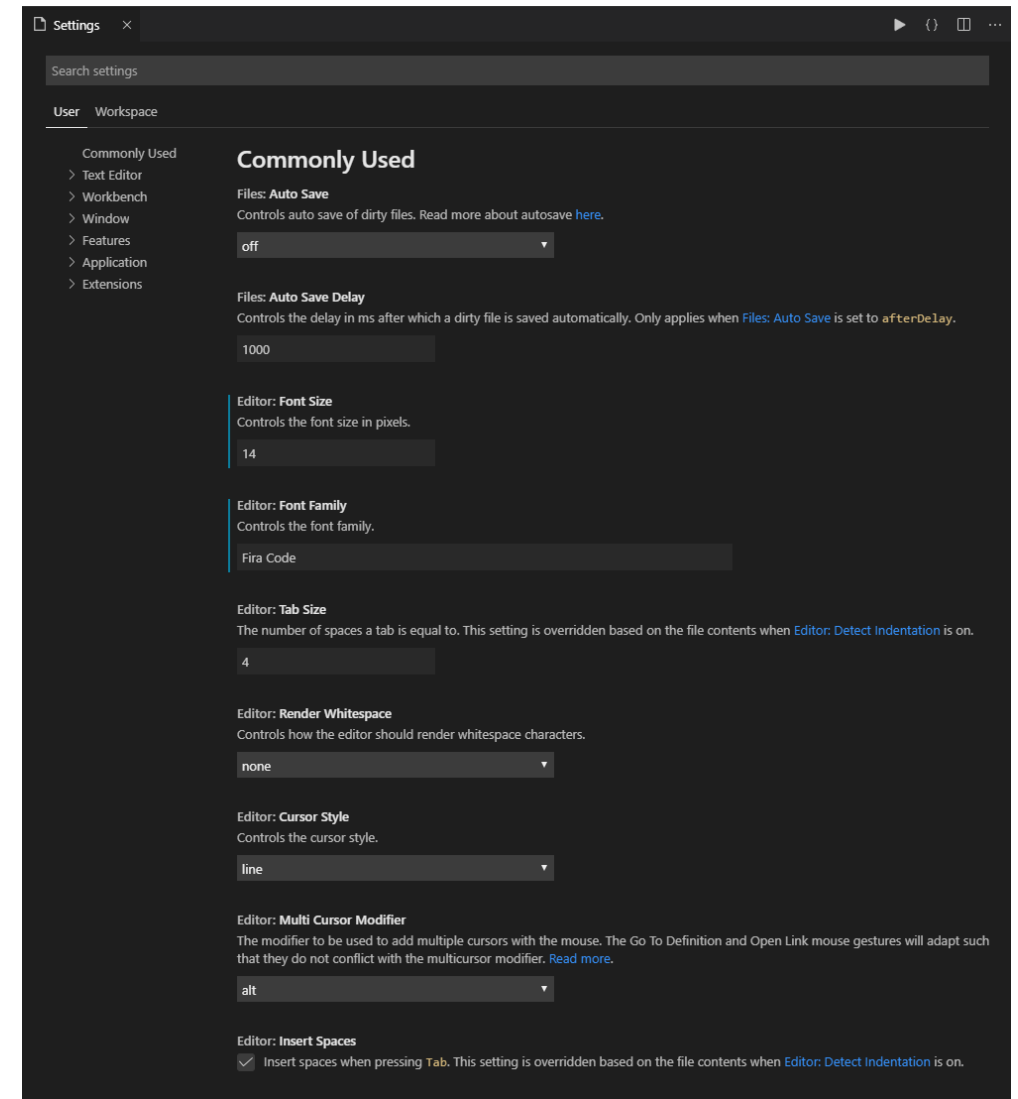
Ctrl + , — переход в окно настроек.

Настройки делятся на 3 типа:

Default User Settings — стандартные настройки. Здесь определяются дефолтные значения для всех возможных настроек редактора. Они доступны только в режиме на чтение, т.е. не переопределяются.

User Settings — пользовательские настройки. Здесь прописываются все глобально переопределяемые стандартные значения.

Workspace Settings — настройки рабочей директории. Те же пользовательские настройки, только не глобальные, а на уровне отдельного проекта.



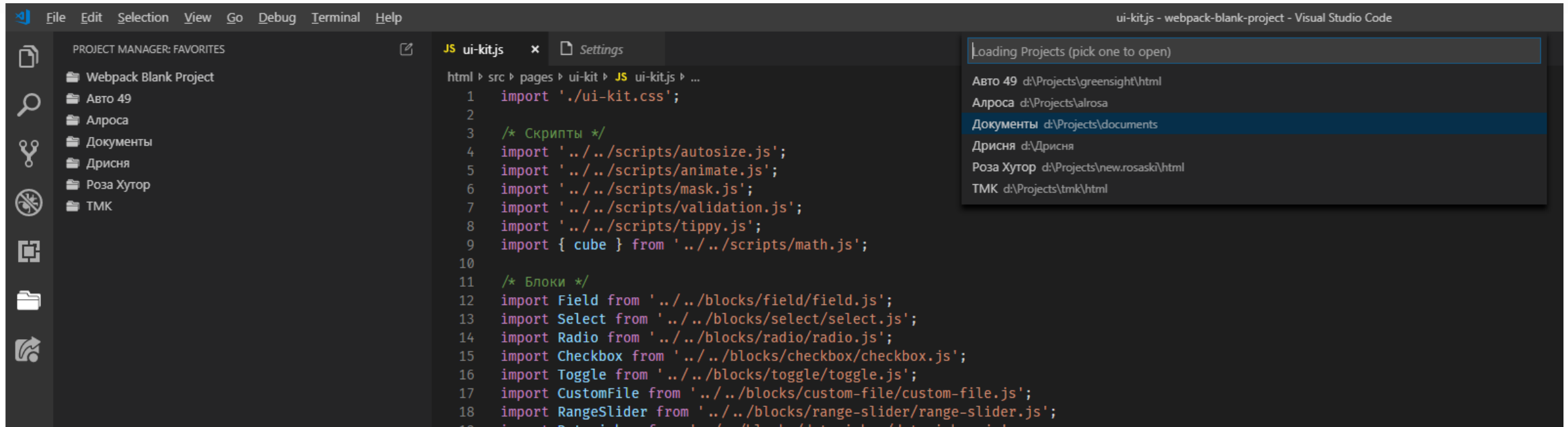
```
1155
1156 <ul class="list">
1157   <li class="list__item list__item--mod">Item 1</li>
1158   <li class="list__item list__item--mod">Item 2</li>
1159   <li class="list__item list__item--mod">Item 3</li>
1160   <li class="list__item list__item--mod">Item 4</li>
1161   <li class="list__item list__item--mod">Item 5</li>
1162 </ul>
1163
```

Emmet встроен в VSCode, его можно увидеть в списке плагинов в категории **Built-in Extensions**.

ul.list>.list__item*5{Item \$}

Нажать **Tab** и получить:

```
<ul class="list">
  <li class="list__item">Item 1</li>
  <li class="list__item">Item 2</li>
  <li class="list__item">Item 3</li>
  <li class="list__item">Item 4</li>
  <li class="list__item">Item 5</li>
</ul>
```



Разные проекты строятся на различных стеках технологий и требуют своего подхода, а значит и своих специфичных настроек. Такие настройки прописываются внутри папки `.vscode`, в корневой директории вашего проекта.

Эта папка создаётся автоматически, когда вы меняете что-либо на уровне проекта.

Это могут быть уже упомянутые ранее **Workspace Settings** (`settings.json`), настройки отдельных языков (`jsconfig.json`), настройки отладки (`launch.json`) и т.д.

ВСТРОЕННЫЙ ТЕРМИНАЛ

Засчёт того, что терминал встроен в редактор, уменьшается переключение контекста, что увеличивает вашу производительность.

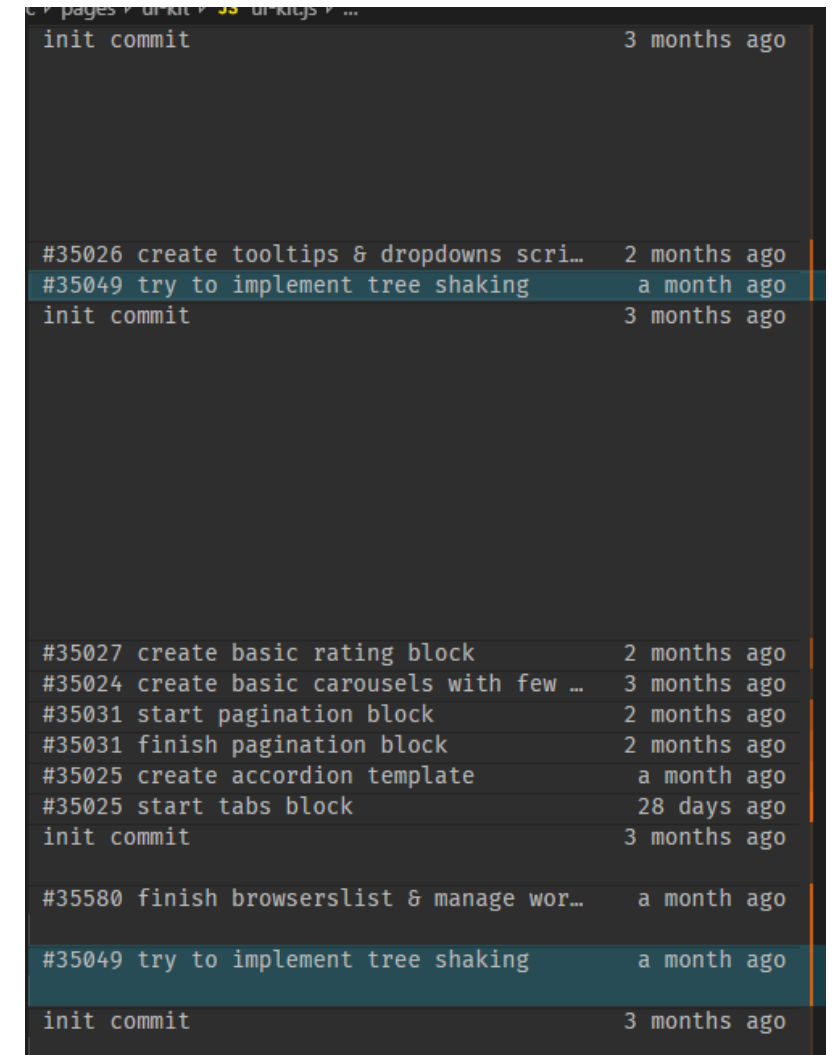
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
User@WIN-4TCJ4MOUHM7 MINGW64 /d/Projects/webpack-blank-project/html (master)
$ yarn dev
yarn run v1.9.4
$ cross-env BROWSERSLIST_ENV=development webpack-dev-server --env.mode dev
i [wds]: Project is running at http://localhost:3000/
i [wds]: webpack output is served from /
i [wdm]: wait until bundle finished: /ui-kit.html
i [wdm]: Hash: a7567f3526d340c6565f
Version: webpack 4.20.2
Time: 11831ms
```

Asset	Size	Chunks	Chunk Names
fonts/pt_sans-web-bold.woff	43.1 KiB	[emitted]	
images/ampm.jpg	134 KiB	[emitted]	
images/macdonalds.jpg	263 KiB	[emitted]	
images/merlin.jpg	148 KiB	[emitted]	
images/mvideo.jpg	731 KiB	[emitted]	
images/sportmaster.jpg	287 KiB	[emitted]	
images/yakitoriya.jpg	230 KiB	[emitted]	
fonts/pt_sans-web-bold-italic.woff	45.3 KiB	[emitted]	
images/baon.jpg	151 KiB	[emitted]	
fonts/pt_sans-web-italic.woff	55.9 KiB	[emitted]	
fonts/pt_sans-web-regular.woff	45.3 KiB	[emitted]	
scripts/test-1.js	1.62 MiB	test-1 [emitted]	test-1

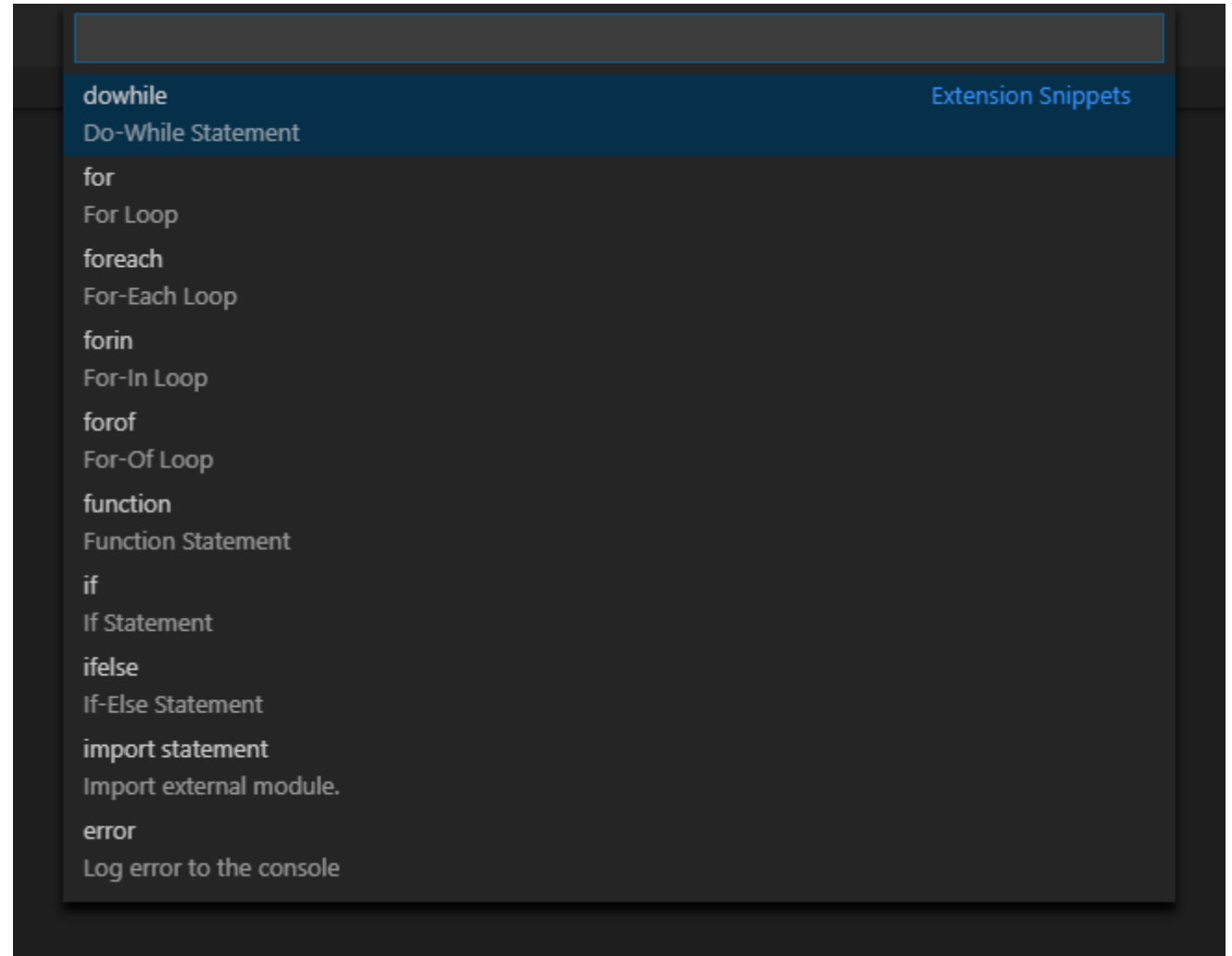
git status. Для отслеживания статуса достаточно посмотреть на представление файлов на вкладке Source Control.

git add. Кнопка “**Stage Changes**” рядом с именем файла (иконка “+”) добавляет указанный файл.

git pull / git push. В статус баре выводится количество несинхронизированных коммитов. Нажатие по этой кнопке выполняет синхронизацию, т.е. как раз pull и push.



Сниппеты — это фрагменты кода, спрятанные за кодовым словом. При вводе этого слова и нажатии Tab или Enter ассоциированный с ним код раскрывается, и кроме того предоставляется набор позиций для курсора, по которым мы можем пройти по Tab, пока не окажемся на конечной позиции — так называемые **табстопы** (tab stops).



```

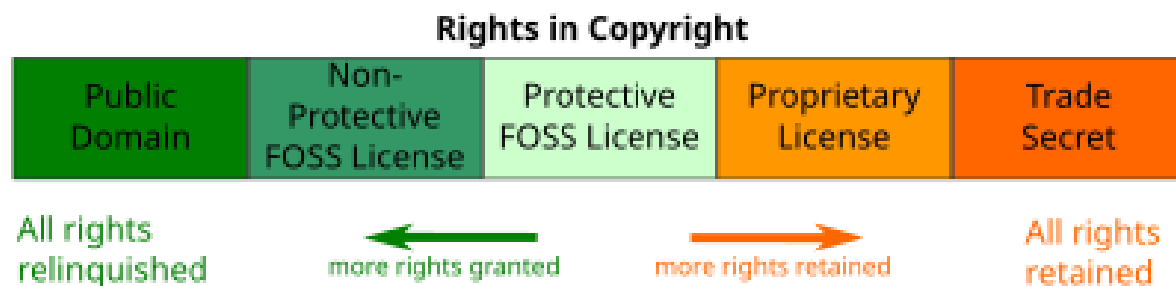
235  /**
236   * Обработчик клика по заголовку таба.
237   *
238   * @callback handleClick
239   * @param {jQuery} $item Заголовок таба.
240   */
241  function handleClick($item) {
242      const $block = $item.closest('.tabs');
243      const tabs = Tabs.getInstance($block);
244
245      const index = tabs.elements.$items.toArray().findIndex((item) => $(item).is($item));
246
247      tabs.
248  }
249
250  /**
251   * обрабо
252   *
253   * @callb
254   * @param
255   * @param {number} keyCode Код клавиши.
256   */
257  function handleItemKeydown($item, keyCode) {
258      if (!Tabs.keyCodes.includes(keyCode)) return;
259
260      const $block = $item.closest('.tabs');
261      const tabs = Tabs.getInstance($block);
262
263      const index = tabs.state.index;

```

\$block	(method) Tabs.openPanel(index: numb * er): void
addAdditionalMarkup	
elements	
init	Открывает панель.
initStates	
openPanel	@param index — Номер открываемой панели.
state	

Intellisense — это общий термин для обозначения автокомплита кода.

Лицензия на программное обеспечение — это правовой инструмент, определяющий использование и распространение программного обеспечения, защищённого авторским правом. Обычно лицензия на программное обеспечение разрешает получателю использовать одну или несколько копий программы, причём без лицензии такое использование рассматривалось бы в рамках закона как нарушение авторских прав издателя. По сути, лицензия выступает гарантией того, что издатель ПО, которому принадлежат исключительные права на программу, не подаст в суд на того, кто ею пользуется.



Лицензии на программное обеспечение с точки зрения степени ограничений со стороны авторского права согласно классификации профессора Марка Веббинка

Лицензии на программное обеспечение в целом делятся на две большие группы: **несвободные** (собственнические, они же проприетарные; и полусвободные) и лицензии **свободного и открытого ПО**. Их различия сильно влияют на права конечного пользователя в отношении использования программы.

Пользовательское соглашение (англ. **end-user license agreement** — «лицензионное соглашение с конечным пользователем»; сокращается до EULA) — договор между владельцем компьютерной программы и пользователем её копии. Обычно используется вместе с проприетарным программным обеспечением (например, продуктами корпораций Microsoft, Apple), а также дистрибутивами свободного программного обеспечения с несвободными элементами (например, Mozilla Firefox до версии 3.0.4 включительно).

ТИПЫ ЛИЦЕНЗИЙ





Проприетарное программное обеспечение

https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5

Коммерческое программное обеспечение

https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BC%D0%B5%D1%80%D1%87%D0%B5%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5

Свободное программное обеспечение

https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5

Определение Open Source

https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_Open_Source