# **Разрешение конфликта через Visual Studio Code**

В прошлом уроке мы показали, как разрешать конфликты слияния вручную и с помощью vimdiff. Эти способы помогают лучше понять, как взаимодействовать с конфликтами, но всё же в работе над реальными проектами чаще используются другие.

Например, большинство разработчиков предпочитают более удобные интерфейсы, чем у vimdiff. Такие интерфейсы предлагают почти все современные **среды разработки** (англ. ***I****ntegrated* ***D****evelopment* ***E****nvironment*, или *IDE*, «интегрированная среда разработки») — программы, в которых собраны инструменты для быстрой и эффективной работы над проектом.

В этом уроке покажем, как выглядят конфликты в среде разработки **Visual Studio Code** (или сокращённо **VS code**).

### **Разрешаем конфликт с помощью VS Code**

Разберём, какие инструменты предоставляет VS Code для разрешения конфликта. Но сперва снова сымитируйте его.

Удалите текущую папку git-conflict, которую создали в прошлом уроке. Затем пересоздайте её с помощью следующей команды.

mkdir git-conflict && cd git-conflict && git init && \

echo 'main version' > readme.md && git add . && git commit -m 'main' && \

git checkout -b br1 && echo 'version 1' > readme.md && git add . && git commit -m 'v1' && \

git checkout main && \

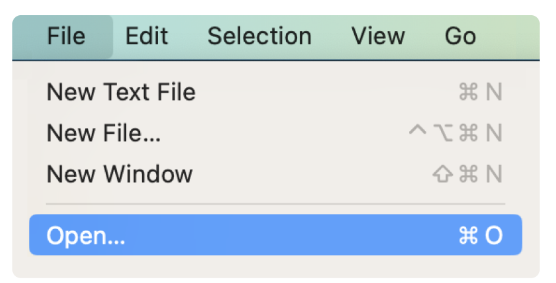
git checkout -b br2 && echo 'version 2' > readme.md && git add . && git commit -m 'v2' && \

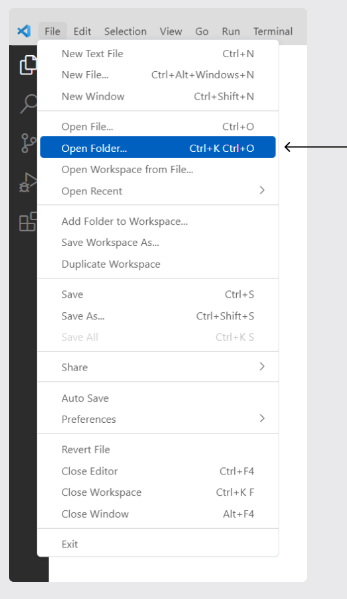
git checkout main && git merge br1 && \

git checkout main

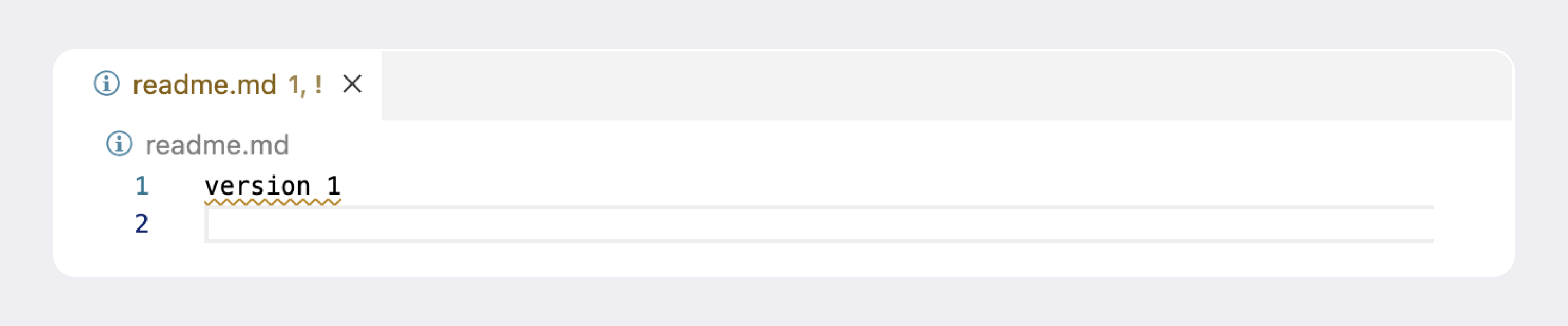
Важно: если вы используете ветку master по умолчанию, то замените команды git checkout main на git checkout master.

Вы в одном шаге от конфликта! Но перед тем как создать его, откройте папку git-conflict в VS Code. Сделать это можно через меню **File → Open…** (для macOS) или **Open Folder…** (для Windows).

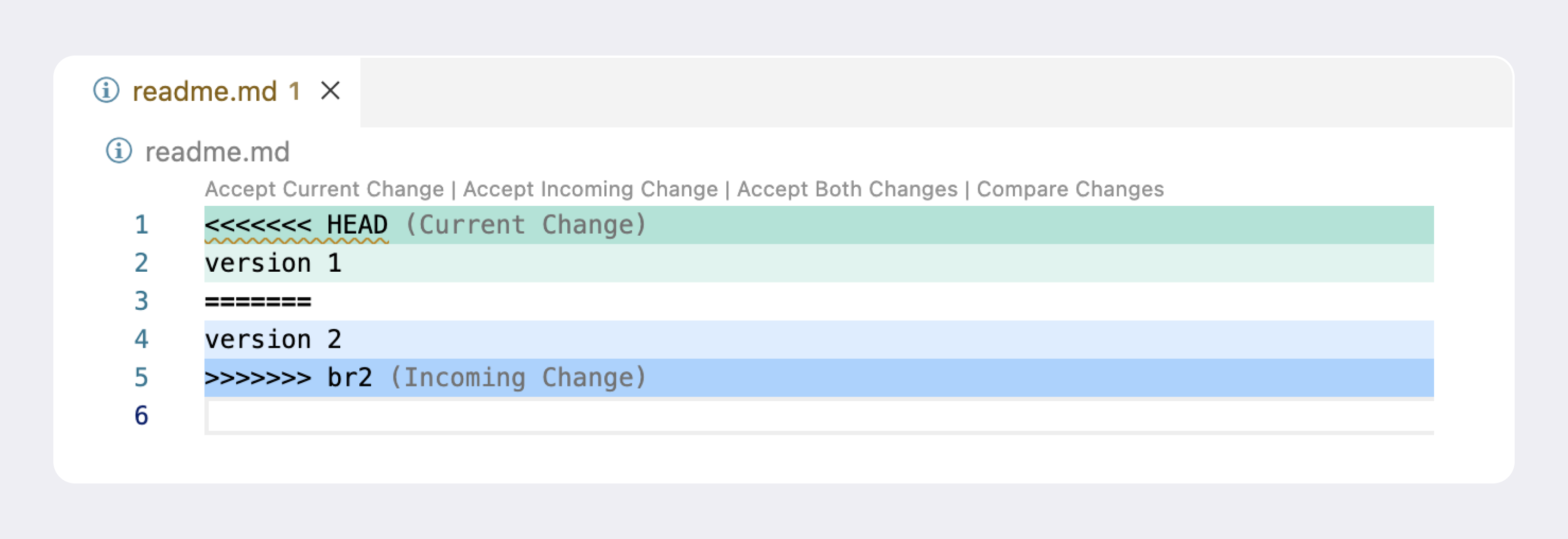




Откроется окно редактора.



Пришло время для конфликта! Выполните git merge br2 в консоли. Теперь в окне редактора появятся маркеры конфликта.



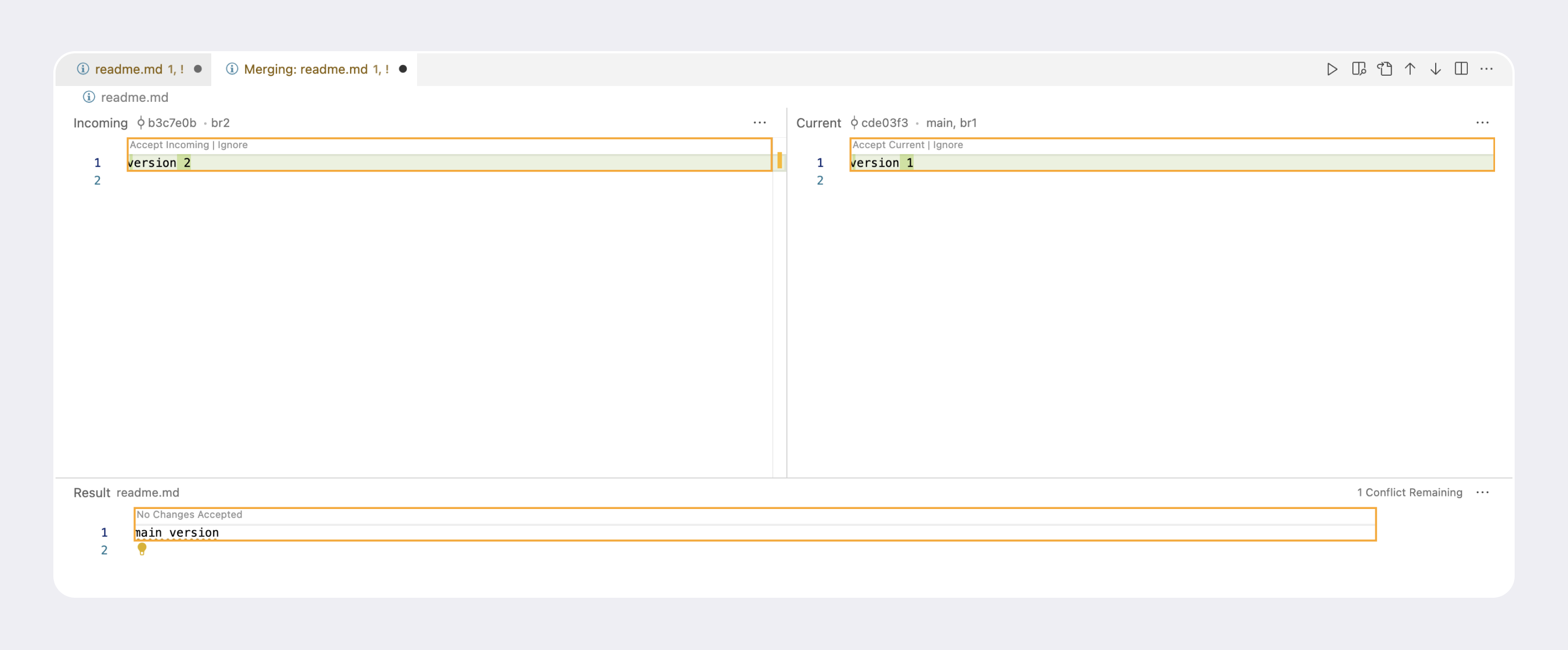
Зелёным цветом подсвечивается текущая версия, а синим — новые изменения. Вы можете разрешить конфликт прямо в файле вручную. А можете использовать вспомогательные кнопки:

* **Accept Current Change** (англ. «принять текущие изменения») — разрешить конфликт через изменения, которые были раньше. В нашем примере эта кнопка оставит строку version 1 и удалит всё остальное.
* **Accept Incoming Change** (англ. «принять входящие изменения») — разрешить конфликт через изменения, которые внесли вы.
* **Accept Both Changes** (англ. «принять все изменения») — добавить оба изменения одно за другим.

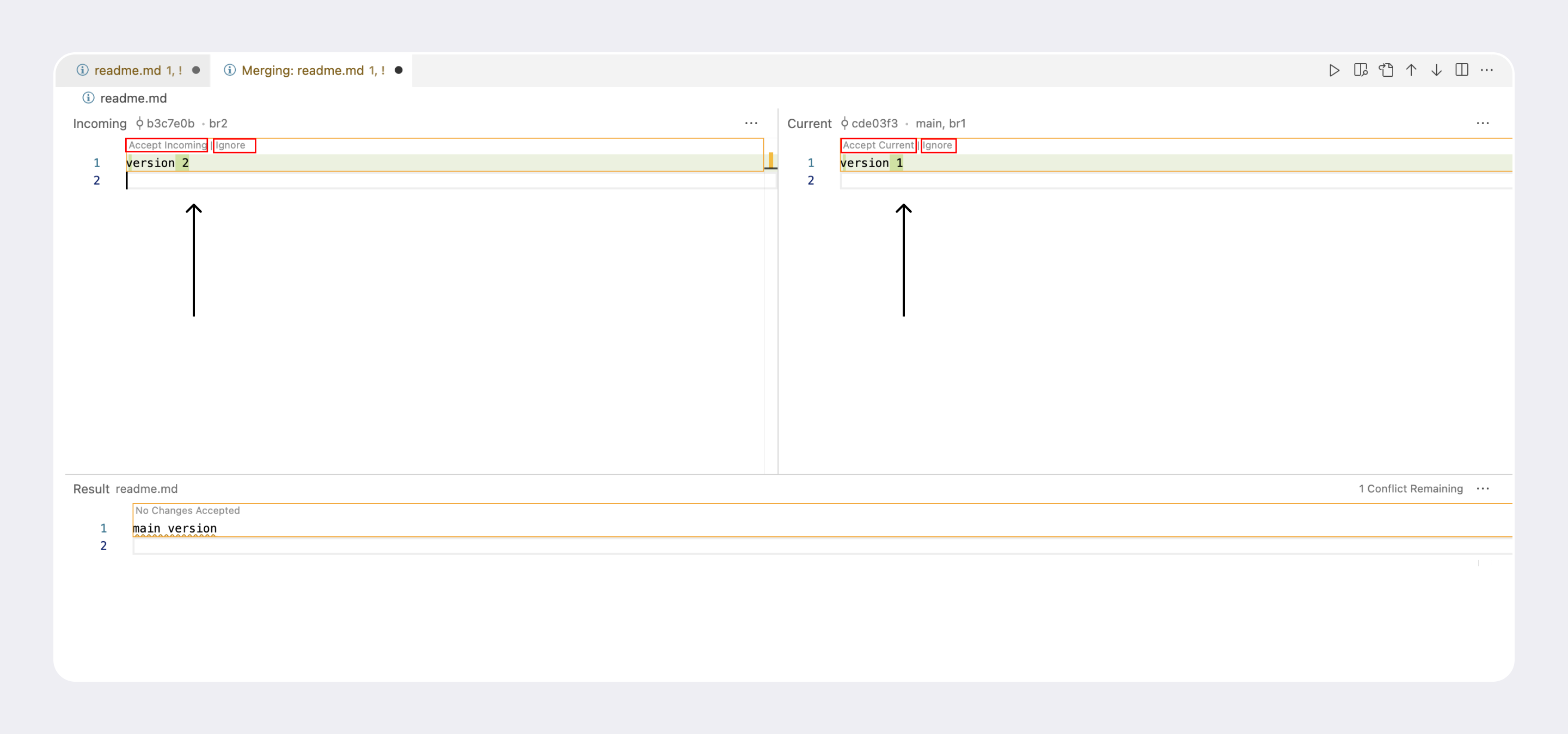
Это удобно, если изменений немного. На случай, если таких изменений больше, у VS Code есть более мощный графический интерфейс для конфликтов. Чтобы его открыть, нажмите на кнопку **Resolve in Merge Editor** (англ. «разрешить в редакторе слияний») в правом нижнем углу экрана.



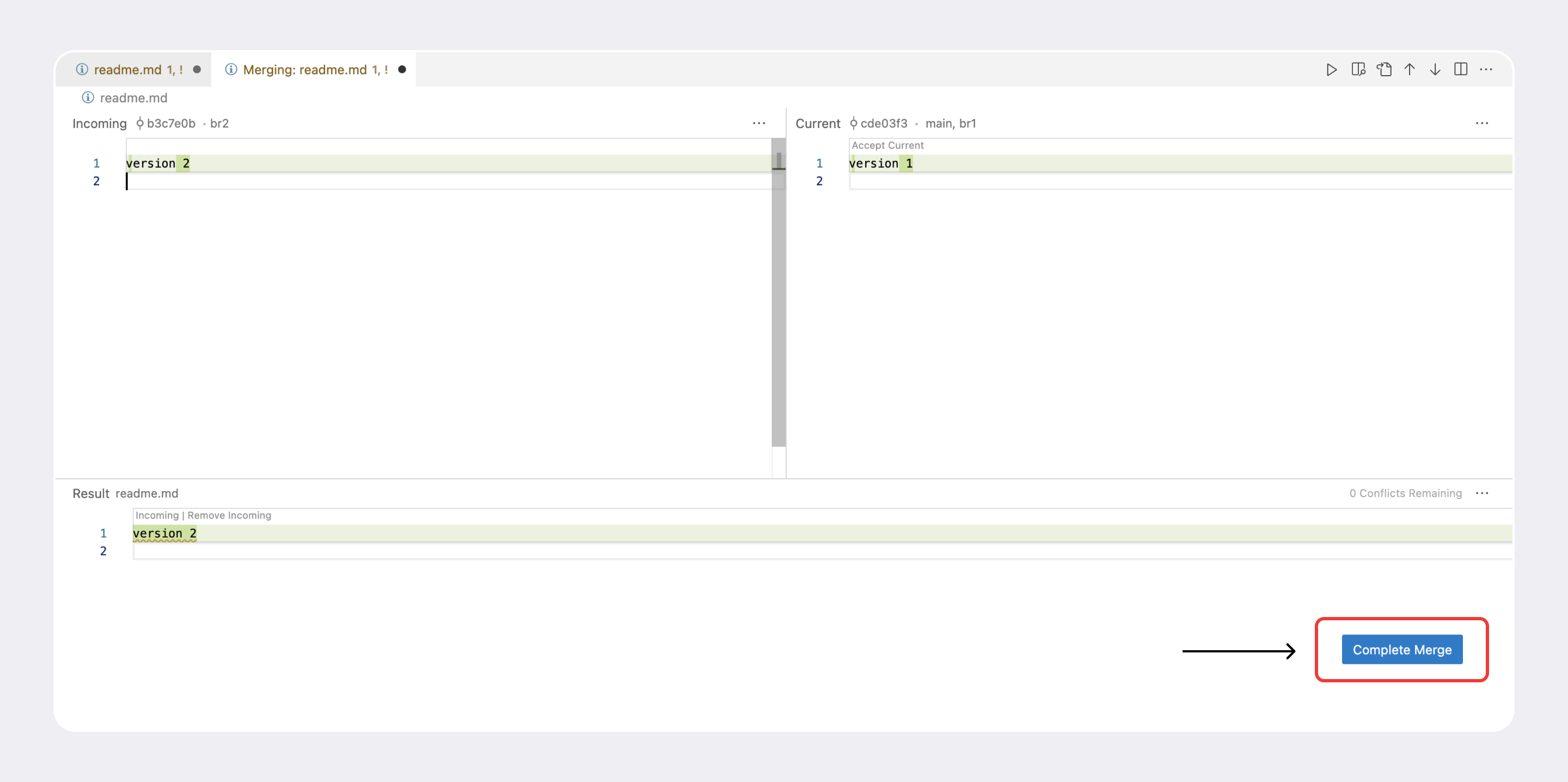
Вы увидите окно разрешения конфликтов. Оно состоит из трёх частей: в левой части содержатся новые изменения, в правой — текущие, а в нижней — результат.



Используйте вспомогательные кнопки **Accept Incoming** (англ. «принять входящие»), **Ignore** (англ. «игнорировать») и **Accept Current** (англ. «принять текущие»), чтобы быстро добавить изменения в результат.



Щёлкните на **Accept Incoming** в левой верхней части экрана. Все конфликты разрешены, поэтому теперь можно нажать на **Complete Merge** (англ. «завершить слияние»).



Готово! Остался последний шаг — создать коммит слияния. VS Code попытается сделать это автоматически в левой части экрана. Измените сообщение коммита, если нужно, и нажмите на кнопку **Commit**. Этот шаг также можно выполнить через консоль.

