

javaguru1 ~ \$ git

Что такое git?

Git - это система управления версиями, которая используется для отслеживания изменений в исходном коде или других текстовых документах.

Git позволяет сохранять различные версии файлов и возвращаться к ним в случае необходимости, а также объединять изменения, внесённые несколькими людьми в один проект.

Как начать работать с git?

1. Нужно установить сам **git**

(ссылка на официальный сайт: <https://git-scm.com/downloads>)

2. Всё, теперь всё готово, чтобы начать работать с git



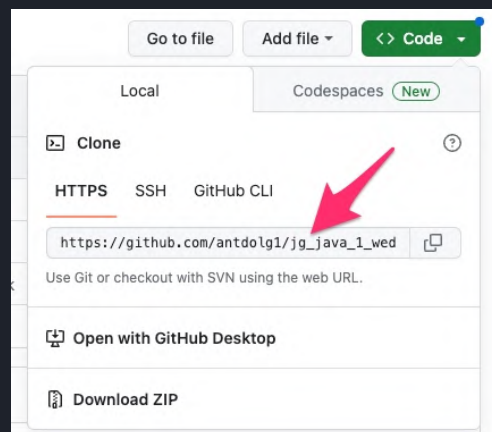
Как клонировать проект?

Через терминал (old school)

1. Перейдите в директорию, в которую вы хотите скопировать (клонировать) проект.

```
| cd ~/Desktop/Projects/JavaGuru/Java1
```

2. Скопируйте URL git репозитория, который вы хотите клонировать.



3. Введите следующую команду:

```
| git clone https://github.com/antdog1/jg_java_1_wednesday_march_2023.git
```

4. Запустите команду нажав 'Enter'

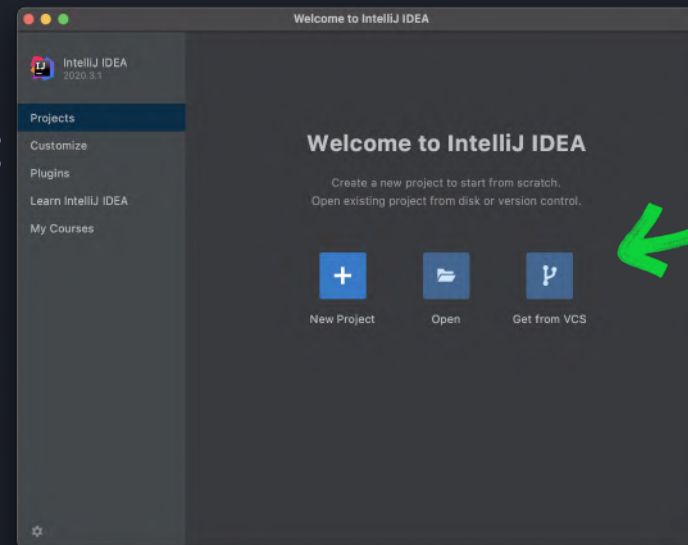
5. Как только процесс клонирования будет завершён, можете перейти в папку проекта и убедиться что всё прошло успешно.

```
| cd jg_java_1_wednesday_march_2023
```

Как клонировать проект?

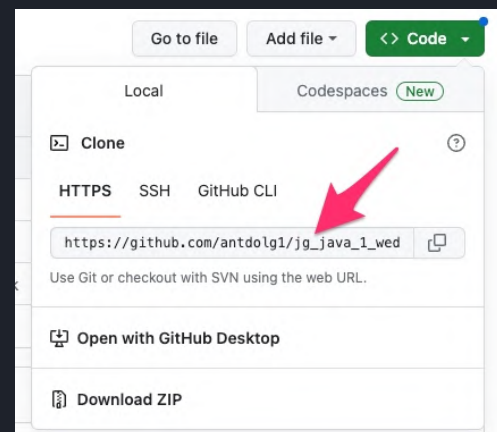
Через IntelliJ Idea (new school)

1. Запускаем IntelliJ Idea и видим такое окошко:



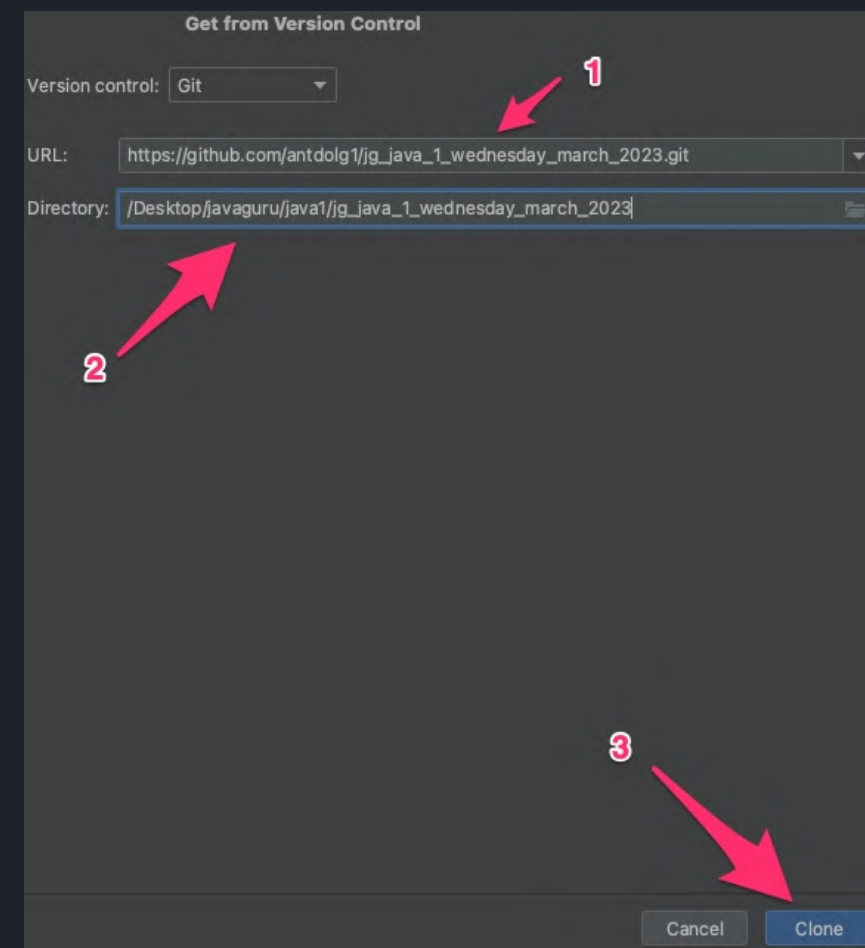
Нажимаем 'Get from VCS'

2. Скопируйте URL git репозитория, который вы хотите клонировать.



3. Переходим в следующее окошко и в нём:

1. Вставляем URL репозитория
2. Выбираем место, куда мы будем копировать наш проект
3. Нажимаем 'Clone'



Шпаргалка по некоторым командам git

`git add`

Команда `git add` добавляет содержимое рабочей директории в индекс (staging area) для последующего коммита. По умолчанию `git commit` использует лишь этот индекс, так что вы можете использовать `git add` для сборки слепка вашего следующего коммита.

`git commit`

Команда `git commit` берёт все данные, добавленные в индекс с помощью `git add`, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

Шпаргалка по некоторым командам git

`git status`

Команда `git status` показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.

`git push`

Команда `git push` используется для установления связи с удалённым репозиторием, вычисления локальных изменений отсутствующих в нём, и собственно их передачи в вышеупомянутый репозиторий. Этой команде нужно право на запись в репозиторий, поэтому она использует аутентификацию.

Шпаргалка по некоторым командам git

`git fetch`

Команда `git fetch` связывается с удалённым репозиторием и забирает из него все изменения, которых у вас пока нет и сохраняет их локально.

`git pull`

Команда `git pull` работает как комбинация команд `git fetch` и `git merge`, т.е. Git вначале забирает изменения из указанного удалённого репозитория, а затем пытается слить их с текущей веткой.

Шпаргалка по некоторым командам git

`git branch`

Команда `git branch` — это своего рода “менеджер веток”. Она умеет перечислять ваши ветки, создавать новые, удалять и переименовывать их.

`git checkout`

Команда `git checkout` используется для переключения веток и выгрузки их содержимого в рабочую директорию.

Полезные ссылки

- [Что такое Git? \(от Atlassian\)](#)
- [Шпаргалка по командам](#)
- [Ё***ый Git!!! \(что делать когда всё сломалось\)](#)
- [Официальная документация](#)
- [Ссылка для скачивания Git](#)
- [Ссылка на наш репозиторий](#)