**Настройка git**

**Примечание**

1. Данная инструкция составлена для тех, кто использует переносную версию git [PortableGit.zip](https://drive.google.com/file/d/10SuqnA5K_zJ0jUHfaf0Tt8lJME7JXw68/view?usp=sharing).
2. Шаги, отмеченные (однажды), выполняются один раз при начальной настройке git.
3. Шаги, отмеченные (каждый запуск), выполняются при каждом запуске терминала git-bash.
4. Шаги, отмеченные (каждая отправка файлов), выполняются при каждой отправке файлов на проверку.

**Ход работы**

1. Если не сделано ранее, зарегистрироваться на github.com и сообшить ИП свой логин (через старосту).
2. Создать 2 репозитория:
   1. репозиторий для практики

Repository name - “train\_repo”.

Проверка репозитория будет выполняться только в случае правильного именования.

* 1. репозиторий для проекта

При работе в группе создаётся только одним участником группы. Репозиторий будет общим, остальные участники присоединятся к нему.

Имя репозитория должно быть связано с названием проекта.

1. Добавить Ilya-Petrovich в список участников в оба репозитория.

Settings -> Collaborators -> Add people

1. Запустить IT/PortableGit/git-bash.exe.
2. Создать пару ключей. (однажды)

Примечание: /e/IT/.ssh - соответствует папке E:\IT\.ssh

| ssh-keygen -f /e/IT/.ssh/key |
| --- |

1. Ввести пароль для ключа. При вводе пароля символы не будут отображаться на экране - это нормально. Вслепую ввести пароль, после нажатия Enter повторить.

| 123 |
| --- |

1. Запустить ssh агент, который будет следить за ключами. (каждый запуск)

| ssh-agent bash |
| --- |

1. Добавить ключ для отслеживания агентом. (каждый запуск)

| ssh-add /e/IT/.ssh/key |
| --- |

| Identity added ... |
| --- |

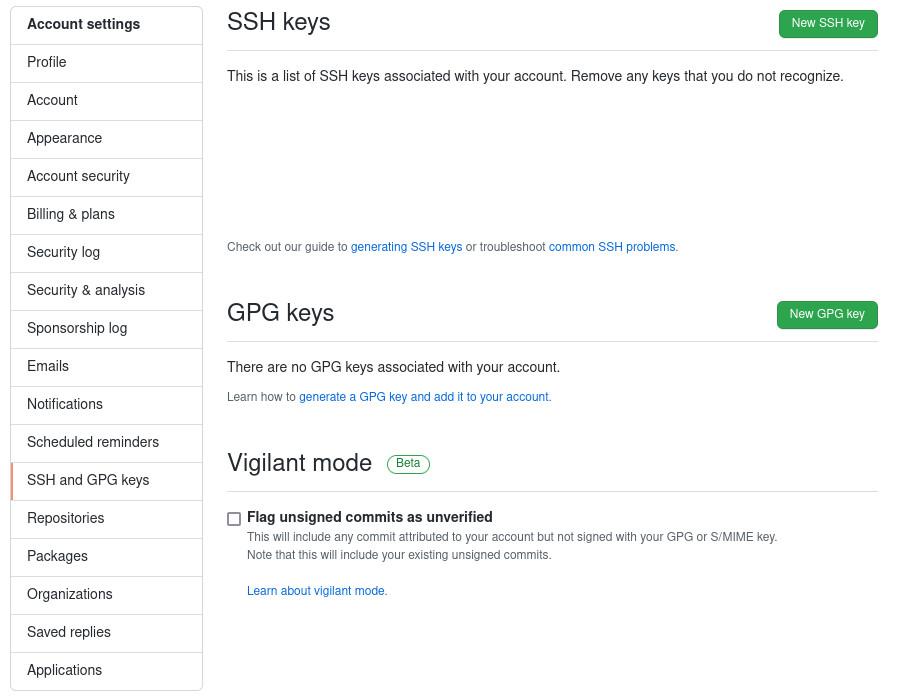
1. Проверить добавился ли ключ можно командой (если имеются добавленные ключи, они будут выведены в консоль).

| ssh-add -L |
| --- |

1. Скопировать ключ из файла /e/IT/.ssh/key.pub.
2. Добавить ssh ключ в свой аккаунт на github.com.

Settings -> SSH and GPG keys -> New SSH key

Задать произвольное название ключа.



1. Проверить соединение с github.

| ssh -T git@github.com |
| --- |

| Hi User! ... |
| --- |

Если github поздоровался, переходи на шаг +2.

Если выходит вопрос:

| Are you sure ... ? |
| --- |

Нужно ввести:

| yes |
| --- |

Если вместо приветствия отображается:

| Permission denied: ... |
| --- |

перейти к следуюшему шагу.

1. Проверить ssh настройки. (каждый запуск)

Открыть файл IT/PortableGit/etc/gitconfig.

Заменить значение переменной sshCommand.

| sshCommand = ssh -i /e/IT/.ssh/key |
| --- |

Сохранить файл.

Открыть файл IT/PortableGit/etc/ssh/ssh\_config

В конце должно быть:

| IdentityFile /e/IT/.ssh/key |
| --- |

Перейти на шаг -1.

1. Перейти в папку IT. Для этого в терминале ввести команду: (каждый запуск)

| cd /e/IT |
| --- |

1. Скопировать репозиторий. (однажды)

| git clone git@github.com:Ilya-Petrovich/train\_repo.git |
| --- |

1. Перейти в скопированный репозиторий. (каждый запуск)

| cd train\_repo |
| --- |

1. Установить имя и почту пользователя. (однажды)

Результаты проверки будут отправляться на указанную почту.

local - для переносной версии (на флэшке или чужом компьютере), global - для своего компьютера.

| git config --local user.name "Ilya Alekseev" |
| --- |

| git config --local user.email "ipteacher@mail.ru" |
| --- |

1. Если после выполнения шага -1 выходит ошибка:

| fatal: ... |
| --- |

Выполнить команду:

| git status |
| --- |

После выполнить предложенную команду:

| git config --global ... |
| --- |

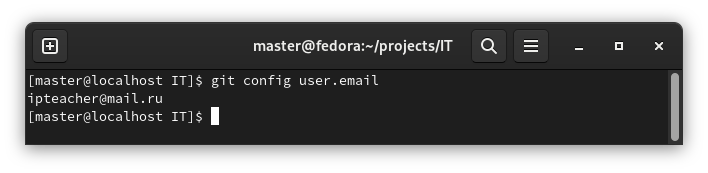
Вернуться к шагу -1.

1. Проверить имя и почту.

| git config user.name |
| --- |



| git config user.email |
| --- |



1. Создать локальную ветку. (однажды)

| git branch first\_branch |
| --- |

1. Перейти на свою ветку. (каждый запуск)

| git checkout first\_branch |
| --- |

1. Проверить текушую ветку. (каждый запуск)

| git branch |
| --- |

Если вышло предупреждение:

| WARNING: terminal is not fully functional Press RETURN to continue |
| --- |

Не обрашать внимание, нажать Enter.

1. Создать файл test.py. Написать код для вывода текстовой строки “ПРивет, мир!”. Сохранить файл.
2. Проверить статус git. (можно выполнять в любой момент и сколько угодно раз для проверки текушего состояния локального репозитория)

| git status |
| --- |

1. Добавить файл в новый коммит. (каждая отправка файлов)

| git add test.py |
| --- |

1. Ешё раз проверить статус.

| git status |
| --- |

1. Сделать коммит. (каждая отправка файлов)

| git commit -m "Added test.py" |
| --- |

1. Установить связь локального репозитория с удалённым. (при самом первом push)

| git push --set-upstream origin first\_branch |
| --- |

1. Отправить файлы, добавленные в коммит, в удалённый репозиторий. (каждая отправка файлов)

Чтобы убедиться, что push был выполнен успешно, проверить содержимое своей ветки на github.com.

| git push |
| --- |

1. Исправить ошибку, сделать ешё один коммит, убедиться в правильном результате на сайте github.com.

**Исправление ошибок**

1. Если при выполнении команды git push выходит одна из ошибок:

| hint: Updates were rejected because the tip of your current branch is behind hint: its remote counterpart. Integrate the remote changes (e. g. hint: 'git pull ...') before pushing again. hint: See the 'Note about fast-forwards' in 'git push -help' for details. |
| --- |

| error: failed to push some refs to 'github.com:Ilya-Petrovich/IT\_LR.git' hint: Updates were rejected because the remote contains work that you do hint: not have locally. This is usually caused by another repository pushing hint: to the same ref. You may want to first integrate the remote changes hint: (e.g., 'git pull ...') before pushing again. hint: See the 'Note about fast-forwards' in 'git push --help' for details. |
| --- |

следует выполнить следуюшие действия:

1. Перенести всё содержимое папки **IT\_LR** (кроме папки **.git**, если она отображается) в другое место (например в папку IT).
2. Последовательно выполнить команды:

| git add . |
| --- |

| git commit -m "fixing branch step 1" |
| --- |

| git pull |
| --- |

1. Если выходит следуюшее сообшение:

| If you wish to set tracking information for this branch you can do so with:   git branch --set-upstream-to=origin/TRP-1-21-1 |
| --- |

следует выполнить предложенную команду, после чего выполнить команду:

| git pull |
| --- |

1. Если выходит следуюшее сообшение (или похожее):

| fatal: The current branch TRP-1-21-1 has no upstream branch. To push the current branch and set the remote as upstream, use  git push --set-upstream origin TRP-1-21-1  To have this happen automatically for branches without a tracking upstream, see 'push.autoSetupRemote' in 'git help config'. |
| --- |

следует выполнить предложенную команду.

1. Если выходит следуюшее сообшение (или похожее):

| hint: You have divergent branches and need to specify how to reconcile them. hint: You can do so by running one of the following commands sometime before hint: your next pull: hint: hint: git config pull.rebase false *# merge* hint: git config pull.rebase true *# rebase* hint: git config pull.ff only *# fast-forward only* hint: hint: You can replace "git config" with "git config --global" to set a default hint: preference for all repositories. You can also pass --rebase, --no-rebase, hint: or --ff-only on the command line to override the configured default per hint: invocation. |
| --- |

следует выполнить следуюшие команды:

| git config pull.rebase true |
| --- |

| git pull |
| --- |

1. Если выходит следуюшее сообшение (или похожее):

| hint: Resolve all conflicts manually, mark them as resolved with hint: "git add/rm <conflicted\_files>", then run "git rebase --continue". hint: You can instead skip this commit: run "git rebase --skip". hint: To abort and get back to the state before "git rebase", run "git rebase --abort". |
| --- |

следует выполнить следуюшую команду (возможно несколько раз):

| git rebase --skip |
| --- |

1. Далее выполнить следуюшую команду:

| git push |
| --- |

1. Удалить всё содержимое папки **IT\_LR** (при наличии). КРОМЕ ПАПКИ “.git”, если она отображается!
2. Скопировать (НЕ ПЕРЕНЕСТИ!) папки и файлы, перенесённые в шаге 1 в папку **IT\_LR**.
3. Выполнить команды:

| git add . |
| --- |

| git commit -m "fixing branch step 2" |
| --- |

| git push |
| --- |

1. Если команда push выполнилась без ошибок, можно удалить папки и файлы (перенесённые на шаге 1) из папки **IT**.