

Тренинг по Git

Часть 1

git clone

Если вы желаете получить копию существующего репозитория Git, например, проекта, в котором вы хотите поучаствовать, то вам нужна команда `git clone [url]`

```
nick@NICK-PC /c/students
$ cd rtsoft_training/

nick@NICK-PC /c/students/rtsoft_training
$ git clone https://github.com/nbalandin/RTSoftGitTraining.git
Cloning into 'RTSoftGitTraining'...
remote: Counting objects: 40, done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 40 (delta 19), reused 33 (delta 12), pack-reused 0
Unpacking objects: 100% (40/40), done.

nick@NICK-PC /c/students/rtsoft_training
$
```

git config

В состав Git'a входит утилита `git config`, которая позволяет просматривать и устанавливать параметры, контролирующие все аспекты работы Git'a и его внешний вид.

Первое, что вам следует сделать после установки Git'a, — указать ваше имя и адрес электронной почты. Это важно, потому что каждый коммит в Git'e содержит эту информацию, и она включена в коммиты, передаваемые вами:

```
nick@NICK-PC ~/Documents/Visual Studio 2013/Projects/WindowsFormsApplication1 (master)
$ git config user.name
Nikolay Balandin

nick@NICK-PC ~/Documents/Visual Studio 2013/Projects/WindowsFormsApplication1 (master)
$ git config user.email
nbalandin@dev.rtsoft.ru

nick@NICK-PC ~/Documents/Visual Studio 2013/Projects/WindowsFormsApplication1 (master)
$ -
```

git log — просмотр истории

По умолчанию, без аргументов, git log выводит список коммитов созданных в данном репозитории в обратном хронологическом порядке.

```
MINGW32:/c/students/rtsoft_training/RTSoftGitTraining
commit 951667a6bd6cfd56629bcb9b6535e4a7b271430
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 17:04:38 2015 +0300

    change combobox dropdown style

commit d411cb714cc8704ea82e49ce39390ed8814c6989
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 17:01:13 2015 +0300

    add conversion to grayscale in button handler

commit 60a63619f8a34490de9a1490900db6e5b406b834
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 16:58:13 2015 +0300

    add grayscale filter

commit 6da66e936b3bf7a1278a7d2830f07b7a4f1ed42f
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 16:42:04 2015 +0300

    Add button handlers

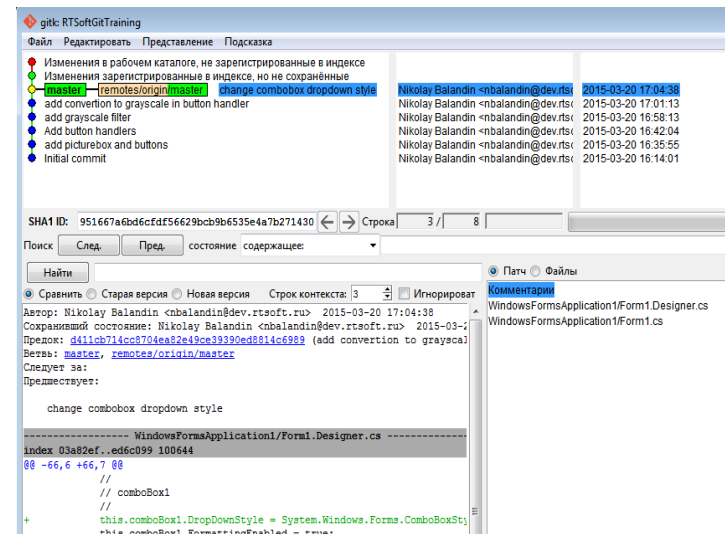
commit 14b0ef603b8f3be27d2f30c8b10063023889d819
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 16:35:55 2015 +0300

    add picturebox and buttons

commit 8bdb0093eda81d45260ee205c173e99e3fba8651
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 16:14:01 2015 +0300

    Initial commit
(END)
```

- ключ —stat покажет статистику по файлам в коммитах
- есть также графичекая тулза gitk



git status

Основной инструмент, используемый для определения, какие файлы в каком состоянии находятся — это команда `git status`

```
nick@NICK-PC /c/students/rtsoft_training/RTSoftGitTraining (master)
$ git status
# On branch master
nothing to commit, working directory clean
```

Меняем файл — добавляем новую кнопку

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1.v12.suo
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
```

gitignore

После компиляции клонированного проекта получаем новые файлы (.exe, .pdb, .dll, .cache, и т.д.)

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1.v12.suo
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       WindowsFormsApplication1/bin/
#       WindowsFormsApplication1/obj/
no changes added to commit (use "git add" and/or "git commit -a")
```

Чтобы отфильтровать результаты компиляции и ненужные файлы, в корень репозитория нужно добавить файл .gitignore, в котором перечислить расширения ненужных файлов, таким образом мы не будем их видеть

gitignore (продолжение)

Можно найти готовый шаблон такого файла для конкретного языка и IDE, например <https://github.com/github/gitignore/blob/master/VisualStudio.gitignore>

```
45 *.meta
46 *.obj
47 *.pch
48 *.pdb
49 *.pgc
50 *.pgd
51 *.---
```

Загружаем его и сохраняем в .gitignore

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1.v12.suo
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .gitignore
no changes added to commit (use "git add" and/or "git commit -a")
```

git add

Для добавления нового файла или измененного файла к коммиту используется git add

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git add WindowsFormsApplication1/Form1.Designer.cs

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git add .gitignore
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       new file:   .gitignore
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1.v12.suo
#
```

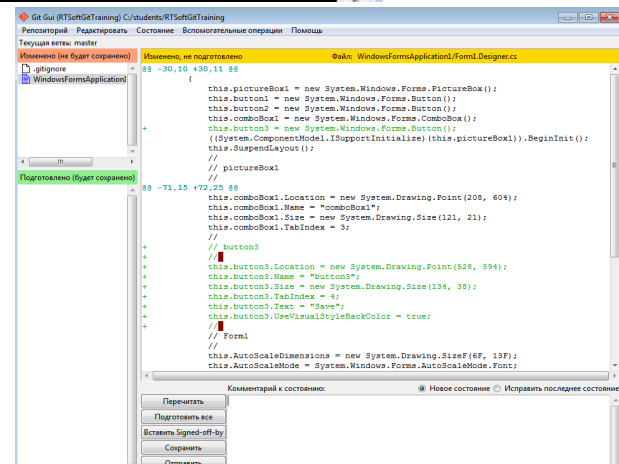

git commit

Для сохранения изменений `git commit -m «комментарий»`
После этого изменения попадут в лог, но только в локальной копии репозитория

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git commit -m "add button, add .gitignore"
[master 242e8de] add button, add .gitignore
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory.
2 files changed, 208 insertions(+)
create mode 100644 .gitignore

nick@NICK-PC /c/students/RTSoftGitTraining (master)
```

Есть графическая тулза — git citool



Отмена изменений

На любой стадии может возникнуть необходимость что-либо отменить.

Будьте осторожны, так как не всегда можно отменить сами отмены. Это одно из немногих мест в Git'е, где вы можете потерять свою работу если сделаете что-то неправильно.

Помните, что всё, что является частью коммита в Git'е, почти всегда может быть восстановлено.

Всё, что никогда не попадало в коммит, вы скорее всего уже не увидите снова

git checkout

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git checkout d411cb WindowsFormsApplication1/Form1.Designer.cs

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
# Changes not staged for commit:
```

Отмена изменений файла:

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1.v12.suo
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
```

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git checkout HEAD WindowsFormsApplication1/Form1.Designer.cs

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1.v12.suo
#
```

git reset

Отмена индексации файла:

\$ git reset HEAD benchmarks.rb

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   WindowsFormsApplication1.v12.suo
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .gitignore
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git reset HEAD WindowsFormsApplication1/Form1.Designer.cs
Unstaged changes after reset:
M   WindowsFormsApplication1/Form1.Designer.cs
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   WindowsFormsApplication1.v12.suo
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
```

git reset --hard

Самые страшные команды в гите: git reset --hard и git push --force.
Команда на все случаи, когда нужно что-то отменить:

\$ git reset --hard <commit>

<commit> в данном случае — та точка, к которой нужно откатиться.
Все текущие изменения будут потеряны и история сбросится до <commit>

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#       modified:   WindowsFormsApplication1.v12.suo
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1/Form1.Designer.cs
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .gitignore
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git reset --hard HEAD
HEAD is now at 951667a change combobox dropdown style
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .gitignore
nothing added to commit but untracked files present (use "git add" to track)
```

git revert

\$ git revert <commit>

ОТМЕНИТ КОММИТ С ЗАДАНЫМ ХЕШЕМ. Откат будет сохранен в виде нового коммита. Можете запустить git log, чтобы убедиться в этом.

```
commit ad34341ae3dec38f5815ca76ce5f513e36d37a26
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Wed Mar 25 16:52:25 2015 +0300
```

wrong commit

```
WindowsFormsApplication1/Form1.Designer.cs | 2 ++
1 file changed, 2 insertions(+)
```

```
commit 951667a6bd6cfd56629bcb9b6535e4a7b271430
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 17:04:38 2015 +0300
```

change combobox dropdown style

```
WindowsFormsApplication1/Form1.Designer.cs | 1 +
WindowsFormsApplication1/Form1.cs          | 1 +
2 files changed, 2 insertions(+)
```

```
commit d411cb714cc8704ea82e49ce39390ed8814c6989
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 17:01:13 2015 +0300
```

add conversion to grayscale in button handler

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git revert ad3434
[master e7b20fa] Revert "wrong commit"
1 file changed, 2 deletions(-)
```

```
commit e7b20fa8b411daf2d917868a5f5a3b7c7ff722790
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Wed Mar 25 16:53:27 2015 +0300
```

Revert "wrong commit"

This reverts commit ad34341ae3dec38f5815ca76ce5f513e36d37a26.

```
WindowsFormsApplication1/Form1.Designer.cs | 2 --
1 file changed, 2 deletions(-)
```

```
commit ad34341ae3dec38f5815ca76ce5f513e36d37a26
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Wed Mar 25 16:52:25 2015 +0300
```

wrong commit

```
WindowsFormsApplication1/Form1.Designer.cs | 2 ++
1 file changed, 2 insertions(+)
```

```
commit 951667a6bd6cfd56629bcb9b6535e4a7b271430
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 17:04:38 2015 +0300
```

git commit —amend

Одна из типичных отмен происходит тогда, когда вы делаете коммит слишком рано, забыв добавить какие-то файлы, или напутали с комментарием к коммиту. Если вам хотелось бы сделать этот коммит ещё раз, вы можете выполнить `commit` с опцией `--amend`:

Для примера, если после совершения коммита вы осознали, что забыли проиндексировать изменения в файле, которые хотели добавить в этот коммит, вы можете сделать что-то подобное:

```
$ git commit -m 'initial commit'  
$ git add forgotten_file  
$ git commit --amend
```

git branch

git branch показывает список локальных ветвей

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git branch
* master

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git branch -a
* master
remotes/origin/HEAD -> origin/master
remotes/origin/master

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$
```

Создание новой ветви и переключение на неё:

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git branch experimental

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git checkout experimental
Switched to branch 'experimental'

nick@NICK-PC /c/students/RTSoftGitTraining (experimental)
$ git branch
* experimental
master
```

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git checkout -b experimental
Switched to a new branch 'experimental'
```


git merge

git merge <branch> - выполняет слияние текущей ветки с веткой branch

Создаем отдельную ветвь для добавления нового фильтра и переключаемся на нее

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git checkout -b black_white_filter
Switched to a new branch 'black_white_filter'
```

Добавляем новый тип фильтра и коммитим его

```
nick@NICK-PC /c/students/RTSoftGitTraining (black_white_filter)
$ git status
# On branch black_white_filter
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1/Form1.cs
#       modified:   WindowsFormsApplication1/ImageFilter.cs
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .gitignore
no changes added to commit (use "git add" and/or "git commit -a")

nick@NICK-PC /c/students/RTSoftGitTraining (black_white_filter)
$ git add *.cs

nick@NICK-PC /c/students/RTSoftGitTraining (black_white_filter)
$ git commit -m "Add black-white filter"
[black_white_filter 7ef9b37] Add black-white filter
2 files changed, 24 insertions(+)
```

git merge (продолжение)

Переключаемся в master и коммитим в него (добавляем копирайт в начало *.cs-файлов):

\$ git checkout master

```
nick@NICK-PC /c/students/RTSoftGitTraining (black_white_filter)
$ git checkout master
Switched to branch 'master'

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   WindowsFormsApplication1/Form1.cs
#       modified:   WindowsFormsApplication1/ImageFilter.cs
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       .gitignore
no changes added to commit (use "git add" and/or "git commit -a")

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git add *.cs

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git commit -m "Add copyrights"
[master 4871d8c] Add copyrights
2 files changed, 26 insertions(+), 2 deletions(-)

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$
```

git merge (продолжение)

Теперь сливаем изменения из black_white_filter в master

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git merge black_white_filter
Auto-merging WindowsFormsApplication1/Form1.cs
Merge made by the 'recursive' strategy.
 WindowsFormsApplication1/Form1.cs | 7 ++++++
 1 file changed, 7 insertions(+)
```

```
commit 103bca6889f3684da9bc39aaf8704533d3e932a0
Merge: 4871d8c 7ef9b37
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date: Wed Mar 25 19:41:42 2015 +0300

    Merge branch 'black_white_filter'

commit 4871d8cb705cb1f345f1a5c17e104db442f67353
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date: Wed Mar 25 19:24:59 2015 +0300

    Add copyrights

 WindowsFormsApplication1/Form1.cs | 6 +++++-
 WindowsFormsApplication1/ImageFilter.cs | 22 ++++++
 2 files changed, 26 insertions(+), 2 deletions(-)

commit 7ef9b375d4db538ef7c54f6a60dd04f92c50390a
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date: Wed Mar 25 19:19:43 2015 +0300

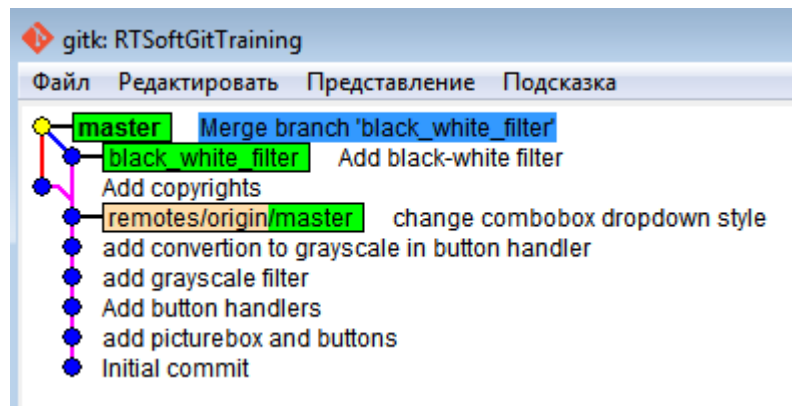
    Add black-white filter

 WindowsFormsApplication1/Form1.cs | 7 ++++++
 WindowsFormsApplication1/ImageFilter.cs | 17 ++++++
 2 files changed, 24 insertions(+)

commit 951667a6bd6cfd56629bcb9b6535e4a7b271430
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date: Fri Mar 20 17:04:38 2015 +0300

    change combobox dropdown style

 WindowsFormsApplication1/Form1.Designer.cs | 1 +
 WindowsFormsApplication1/Form1.cs | 1 +
 2 files changed, 2 insertions(+)
```



git remote

Чтобы просмотреть, какие удалённые серверы у вас уже настроены, следует выполнить команду `git remote`. Она перечисляет список имён-сокращений для всех уже указанных удалённых дескрипторов. Если вы клонировали ваш репозиторий, у вас должен отобразиться, по крайней мере, `origin` — это имя по умолчанию, которое Git присваивает серверу, с которого вы клонировали:

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git remote -v
origin https://github.com/nbalandin/RTSoftGitTraining.git (fetch)
origin https://github.com/nbalandin/RTSoftGitTraining.git (push)
```

`$ git remote add <location>` - добавление удаленного репозитория

Создаем новый репозиторий на github и добавляем его:

git push

После проведения работы в экспериментальной ветке, слияния с основной, необходимо обновить удаленный репозиторий (удаленную ветку). Для этого используется команда `git push`

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git remote add temp https://github.com/nbalandin/RTSoftTemp.git

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git push temp master
Username for 'https://github.com': nbalandin
Password for 'https://nbalandin@github.com':
Counting objects: 54, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (54/54), done.
Writing objects: 100% (54/54), 14.82 KiB | 0 bytes/s, done.
Total 54 (delta 31), reused 0 (delta 0)
To https://github.com/nbalandin/RTSoftTemp.git
 * [new branch]      master -> master
```



```
nick@NICK-PC /c/students/temp/RTSoftGitTraining (master)
$ git push --force origin master
Username for 'https://github.com': nbalandin
Password for 'https://nbalandin@github.com':
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/nbalandin/RTSoftGitTraining.git
+ 245ecdb...951667a master -> master (forced update)
```

git fetch

git fetch origin извлекает все наработки, отправленные (push) на этот сервер после того, как вы клонировали его (или получили изменения с помощью fetch). Важно отметить, что команда fetch забирает данные в ваш локальный репозиторий, но не сливает их с какими-либо вашими наработками и не модифицирует то, над чем вы работаете в данный момент. Вам необходимо вручную слить эти данные с вашими, когда вы будете готовы.

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git fetch origin master
From https://github.com/nbalandin/RTSoftGitTraining
 * branch                master      -> FETCH_HEAD

nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git log origin/master
commit 245ecdbf33cb61efd9aae4484b2cb850bf58d7c2
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 27 14:15:20 2015 +0300

    add using statement

commit 951667a6bd6cfd56629bcb9b6535e4a7b271430
Author: Nikolay Balandin <nbalandin@dev.rtsoft.ru>
Date:   Fri Mar 20 17:04:38 2015 +0300
```

git pull

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git pull origin master
From https://github.com/nbalandin/RTSoftGitTraining
* branch      master      -> FETCH_HEAD
Already up-to-date.
```

Выполнение git pull, извлекает (fetch) данные с сервера, с которого вы изначально клонировали, и автоматически пытается слить (merge) их с кодом, над которым вы в данный момент работаете

```
nick@NICK-PC /c/students/RTSoftGitTraining (master)
$ git pull
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 4 (delta 3), reused 4 (delta 3), pack-reused 0
Unpacking objects: 100% (4/4), done.
From https://github.com/nbalandin/RTSoftGitTraining
 951667a..245ecdb master    -> origin/master
Auto-merging WindowsFormsApplication1/ImageFilter.cs
Merge made by the 'recursive' strategy.
 WindowsFormsApplication1/ImageFilter.cs | 1 +
1 file changed, 1 insertion(+)
```

Темы для второго тренинга (углубленный курс)

- Стратегии организации работы с репозиторием
- Теги
- Поиск коммитов, вызвавших ошибку (поиск автора) :-)
- Формирование и наложение патчей
- Переупорядочивание коммитов
- Восстановление данных после `reset --hard` :-)
- Прятанье изменений
- Filter branch
- Git hooks

Литература - «Pro Git», by Scott Chacon and Ben Straub.

Спасибо за внимание!