examples.md

Create new repository

```
$ git init
Initialized empty Git repository in D:/test/repo/.git/
```

Add first file to repository

Add file "hello.txt"

```
$ git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        hello.txt

nothing added to commit but untracked files present (use "git add" to track)
```

```
$ git add hello.txt
```

```
$ git commit -m "My first commit"
[master (root-commit) 15cc061] My first commit
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
```

```
$ git status
On branch master
nothing to commit, working tree clean
```

Modify a file

Modify hello.txt

```
$ 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:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git add hello.txt
```

```
$ git commit -m "Modify a file"
[master 38b223d] Modify a file
 1 file changed, 1 insertion(+), 1 deletion(-)
```

Branching

```
$ git log --oneline
25632ea Modify a file
3dc35b9 My first commit
```

```
$ git branch
* master
```

```
$ git checkout -b my-test
Switched to a new branch 'my-test'
```

Create "test.txt"

```
$ git add test.txt
```

```
$ git commit -m "Add a test file"
[my-test e728269] Add a test file
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
```

```
$ git log --oneline
e728269 Add a test file
25632ea Modify a file
3dc35b9 My first commit
```

```
$ git checkout master
Switched to branch 'master'
$ git log --oneline
25632ea Modify a file
3dc35b9 My first commit
```

Create "slave.txt"

```
$ git add slave.txt
$ git commit -m "Add slave to master"
[master 9589a2d] Add slave to master
 1 file changed, 1 insertion(+)
 create mode 100644 slave.txt
```

Fast-forward

```
$ git checkout -b to-be-fast-forwarded
Switched to a new branch 'to-be-fast-forwarded'
```

Modify "hello.txt"

```
$ git commit -am "Improve greeting"
[to-be-fast-forwarded 4ebf636] Improve greeting
 1 file changed, 1 insertion(+)
```

```
$ git checkout master
Switched to branch 'master'
```

```
$ git merge to-be-fast-forwarded
Updating 9589a2d..4ebf636
```

```
Fast-forward
 hello.txt | 1 +
 1 file changed, 1 insertion(+)
```

Recursive

```
$ git merge my-test
Merge made by the 'recursive' strategy.
 test.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt
```

```
$ git log --oneline --graph
*   2e32353 Merge branch 'my-test'
|\
| * e728269 Add a test file
* | 4ebf636 Improve greeting
* | 9589a2d Add slave to master
|/
* 25632ea Modify a file
* 3dc35b9 My first commit
```

Rebase examples

```
$ git rebase -i HEAD~2
```

```
pick 3efacf6 Improve greeting
pick 6e6ad01 Add a test file
```

```
$ git log --oneline --graph --all
* 418ac30 Add a test file
* 3efacf6 Improve greeting
* fcfa7b8 Add slave to master
| * 6e6ad01 Add a test file
|/
* a6d57e3 Modify a file
* 0c55337 My first commit
```

```
$ git branch -D my-test
Deleted branch my-test (was 6e6ad01).
$ git log --oneline --graph --all
* 418ac30 Add a test file
* 3efacf6 Improve greeting
* fcfa7b8 Add slave to master
* a6d57e3 Modify a file
* 0c55337 My first commit
```

Correct last commit

Create file "error.txt" containing errors

```
$ git add error.txt
$ git commit -m "Introduse errorz"
[master e115c2e] Introduse errorz
 1 file changed, 1 insertion(+)
 create mode 100644 error.txt
```

```
Some errors here
```

```
$ git add error.txt
```

```
$ git commit --amend -m "Introduce errors"
[master 979003e] Introduce errors
 Date: Fri Nov 3 15:52:26 2017 +0100
 1 file changed, 1 insertion(+)
 create mode 100644 error.txt
```

```
$ git log --oneline --graph --all
* 979003e Introduce errors
* 7e8d078 Add a test file
* 4ebf636 Improve greeting
* 9589a2d Add slave to master
* 25632ea Modify a file
* 3dc35b9 My first commit
```

Delete files out of the complete history

Create VS console project "hello"

```
$ git init
Initialized empty Git repository in D:/test/hello/.git/
$ git add .
$ git commit -m Initialize
[master (root-commit) 590fa0c] Initialize
 15 files changed, 163 insertions(+)
 create mode 100644 .vs/hello/v14/.suo
 create mode 100644 hello.sln
 create mode 100644 hello/App.config
 create mode 100644 hello/Program.cs
 create mode 100644 hello/Properties/AssemblyInfo.cs
 create mode 100644 hello/bin/Debug/hello.exe.config
 create mode 100644 hello/bin/Debug/hello.vshost.exe
 create mode 100644 hello/bin/Debug/hello.vshost.exe.config
 create mode 100644 hello/bin/Debug/hello.vshost.exe.manifest
 create mode 100644 hello/hello.csproj
 create mode 100644 hello/obj/Debug/DesignTimeResolveAssemblyReferencesInput.cache
 create mode 100644 hello/obj/Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs
 create mode 100644 hello/obj/Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs
 create mode 100644 hello/obj/Debug/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs
 create mode 100644 hello/obj/Debug/hello.csproj.FileListAbsolute.txt
```

Modify "Program.cs":

```
using System;

namespace hello
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, world!");
        }
    }
}
```

```
$ git add .
$ git commit -m "First approach to solve this complex problem"
[master 02f7599] First approach to solve this complex problem
 7 files changed, 6 insertions(+), 4 deletions(-)
 create mode 100644 hello/bin/Debug/hello.exe
 create mode 100644 hello/bin/Debug/hello.pdb
 create mode 100644 hello/obj/Debug/hello.csprojResolveAssemblyReference.cache
 create mode 100644 hello/obj/Debug/hello.exe
 create mode 100644 hello/obj/Debug/hello.pdb
```

```
$ git filter-branch --index-filter 'git rm --cached --ignore-unmatch hello/bin/* hello/obj/* .vs/*' HEAD
Rewrite 590fa0c172c1abb942f2a65b70650a19b439c2ec (1/2) (0 seconds passed, remaining 0 predicted)    rm '.vs/hello
/v14/.suo'
rm 'hello/bin/Debug/hello.exe.config'
rm 'hello/bin/Debug/hello.vshost.exe'
rm 'hello/bin/Debug/hello.vshost.exe.config'
rm 'hello/bin/Debug/hello.vshost.exe.manifest'
rm 'hello/obj/Debug/DesignTimeResolveAssemblyReferencesInput.cache'
rm 'hello/obj/Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs'
rm 'hello/obj/Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs'
rm 'hello/obj/Debug/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs'
rm 'hello/obj/Debug/hello.csproj.FileListAbsolute.txt'
Rewrite 02f759993479e45f75c588023e87fbe9c12e248d (2/2) (1 seconds passed, remaining 0 predicted)    rm '.vs/hello
/v14/.suo'
rm 'hello/bin/Debug/hello.exe'
rm 'hello/bin/Debug/hello.exe.config'
rm 'hello/bin/Debug/hello.pdb'
rm 'hello/bin/Debug/hello.vshost.exe'
rm 'hello/bin/Debug/hello.vshost.exe.config'
rm 'hello/bin/Debug/hello.vshost.exe.manifest'
rm 'hello/obj/Debug/DesignTimeResolveAssemblyReferencesInput.cache'
rm 'hello/obj/Debug/TemporaryGeneratedFile_036C0B5B-1481-4323-8D20-8F5ADCB23D92.cs'
rm 'hello/obj/Debug/TemporaryGeneratedFile_5937a670-0e60-4077-877b-f7221da3dda1.cs'
rm 'hello/obj/Debug/TemporaryGeneratedFile_E7A71F73-0F8D-4B9B-B56E-8E70B10BC5D3.cs'
rm 'hello/obj/Debug/hello.csproj.FileListAbsolute.txt'
rm 'hello/obj/Debug/hello.csprojResolveAssemblyReference.cache'
rm 'hello/obj/Debug/hello.exe'
rm 'hello/obj/Debug/hello.pdb'

Ref 'refs/heads/master' was rewritten
```

```
$ git log --patch
```

Add ".gitignore" containing:

```
bin/
obj/
.vs/
```

```
$ git add .gitignore
$ git commit -m "Ignore generated files"
[master 4013b9a] Ignore generated files
 1 file changed, 3 insertions(+)
 create mode 100644 .gitignore
```

Handling merge conflicts

Make sure, being in branch "master"

Content of "hello.txt":

```
Hello, people
Nice to meet you
```

```
$ git checkout -b to-be-conflicted
Switched to a new branch 'to-be-conflicted'
```

Replace "Hello" with "Hallo"

```
$ git status
On branch to-be-conflicted
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:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git commit -am "Say hallo instead of hello"
[to-be-conflicted 2459d02] Say hallo instead of hello
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
$ git checkout master
```

Replace "Hello" with "Hola"

```
$ git commit -am "Say hola instead of hello"
[master 7861168] Say hola instead of hello
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
$ git merge to-be-conflicted
Auto-merging hello.txt
CONFLICT (content): Merge conflict in hello.txt
Automatic merge failed; fix conflicts and then commit the result.
```

```
$ git status
On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   hello.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

```
$ git add hello.txt
$ git status
On branch master
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:

        modified:   hello.txt

```

```
$ git commit
[master 669d43d] Merge branch 'to-be-conflicted'
$ git log --oneline --graph
*   669d43d Merge branch 'to-be-conflicted'
|\
| * 1c14344 Say hallo instead of hello
* | 5c9a7bc Say hola instead of hello
|/
* 92eb6a5 Introduce errors
* 418ac30 Add a test file
* 3efacf6 Improve greeting
* fcfa7b8 Add slave to master
* a6d57e3 Modify a file
* 0c55337 My first commit
```

Git stash

Add file "stash-example.txt"

```
$ git add stash-example.txt
$ git commit -m "Add example file for stashing"
[master a2f1a7c] Add example file for stashing
 1 file changed, 1 insertion(+)
 create mode 100644 stash-example.txt
$ git show
commit a2f1a7cb85aaeb9f6d812230d27eb38d2ac17ac6
Author: Christian Dreier <christian.dreier@csa-germany.de>
Date:   Wed Nov 22 18:45:44 2017 +0100

    Add example file for stashing

diff --git a/stash-example.txt b/stash-example.txt
new file mode 100644
index 0000000..c5c58b3
--- /dev/null
+++ b/stash-example.txt
@@ -0,0 +1 @@
+This is an example
```

Modify "stash-example.txt"

```
$ 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:   stash-example.txt

no changes added to commit (use "git add" and/or "git commit -a")
$ git diff
diff --git a/stash-example.txt b/stash-example.txt
index c5c58b3..b355674 100644
--- a/stash-example.txt
+++ b/stash-example.txt
@@ -1 +1,2 @@
 This is an example
+Hello!
```

```
$ git stash
Saved working directory and index state WIP on master: a2f1a7c Add example file for stashing
HEAD is now at a2f1a7c Add example file for stashing
$ git status
On branch master
nothing to commit, working directory clean
```

```
$ git stash pop
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:   stash-example.txt

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (105c57bab11141a585f05a089d23627c69187f2a)
```

Working with remote repositories

Make a new repository named "personal1":

```
$ mkdir personal1

# ...

$ cd personal1
$ git init
```

```
Initialized empty Git repository in D:/test/personal1/.git/
```

New file ""hello.txt""

```
$ git add hello.txt
$ git commit -m "Initialize"
[master (root-commit) 9e347be] Initialize
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
```

```
$ cd ..
$ mkdir central.git
$ cd central.git
$ git init --bare
Initialized empty Git repository in D:/test/central.git/
```

```
$ cd ..\personal1
$ git remote add origin D:/test/central/
```

```
$ git push origin master
Counting objects: 3, done.
Writing objects: 100% (3/3), 238 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To D:/test/central/
 * [new branch]      master -> master
```

`git log` in central and personal1

```
git clone --bare personal1 central-repo2.git
```

```
$ git clone central.git personal2
Cloning into 'personal2'...
done.
```

```
$ cd personal2
$ git remote show origin
* remote origin
  Fetch URL: D:/test/central.git
  Push  URL: D:/test/central.git
  HEAD branch: master
  Remote branch:
    master tracked
  Local branch configured for 'git pull':
    master merges with remote master
  Local ref configured for 'git push':
    master pushes to master (up to date)
$ git log
commit 9e347be6c7d02c7790d98a66cdce45941e841eb4
Author: Christian Dreier <christian.dreier@csa-germany.de>
Date:   Mon Nov 6 11:19:52 2017 +0100

    Initialize
```