

# Git

Patrick Bucher

## Contents

<b>Git Basics</b>	<b>1</b>
Creating a Repository . . . . .	1
Making Changes . . . . .	1
Getting Information . . . . .	2
<b>Configuration (git config)</b>	<b>2</b>
<b>Help (git help)</b>	<b>3</b>
<b>Miscellaneous</b>	<b>3</b>
Switches . . . . .	4

## Git Basics

### Creating a Repository

Create a new Git repository from an existing working directory (move into the working directory first):

```
git init
```

Cloning an existing Git repository from GitHub:

```
git clone https://github.com/[username]/[repository].git
```

### Making Changes

Display changes in the working directory to be staged:

```
git add --dry-run .  
git add -n .
```

Add a file to the staging area:

```
git add [file]
```

Commit changes to a repository (with a message):

```
git commit -m "[message]"
```

Add and commit changes at the same time:

```
git commit --all  
git commit -a
```

## Getting Information

Show a repository's state:

```
git status
```

List a repository's—not the working directory's!—files:

```
git ls-files
```

Show revisions of a file:

```
git blame [file]
```

Show the repository's history of commits (also in one line, with statistics and a combination of those with short statistics):

```
git log  
git log --oneline  
git log --stat  
git log --shortstat --oneline
```

Show changes between files in working directory and the repository—or the staging area, if changes have been staged already:

```
git diff
```

Show changes between files in staging area and in the repository:

```
git diff --staged  
git diff --cached
```

## Configuration (`git config`)

Set global configuration (name and email):

```
git config --global [option] [value]
```

```
git config --global user.name "Patrick Bucher"
```

```
git config --global user.email "patrick.bucher@stud.hslu.ch"
```

Show all configuration:

```
git config --list
```

Show a specific configuration item (name and email):

```
git config [option]
```

```
git config user.name
```

```
git config user.email
```

## Help (git help)

Show the help page (most important commands):

```
git help
```

Show all commands (with pager):

```
git -p help -a
```

Show all available guides:

```
git help -g
```

Getting help on a specific command or read a guide (help itself, the glossary and the tutorial guide):

```
git help [command/subject]
```

```
git help help
```

```
git help glossary
```

```
git help tutorial
```

## Miscellaneous

Starting Git GUI (the package `tk` is required under Linux):

```
git gui
```

Starting Git GUI to commit changes (`citool`):

```
git citool
```

Starting the Git log viewer (gitk):

```
gitk
```

## Switches

Display the installed version of git:

```
git --version
```

Use a pager (usually `less`) for the output:

```
git -p [command]
```

```
git --paginate [command]
```