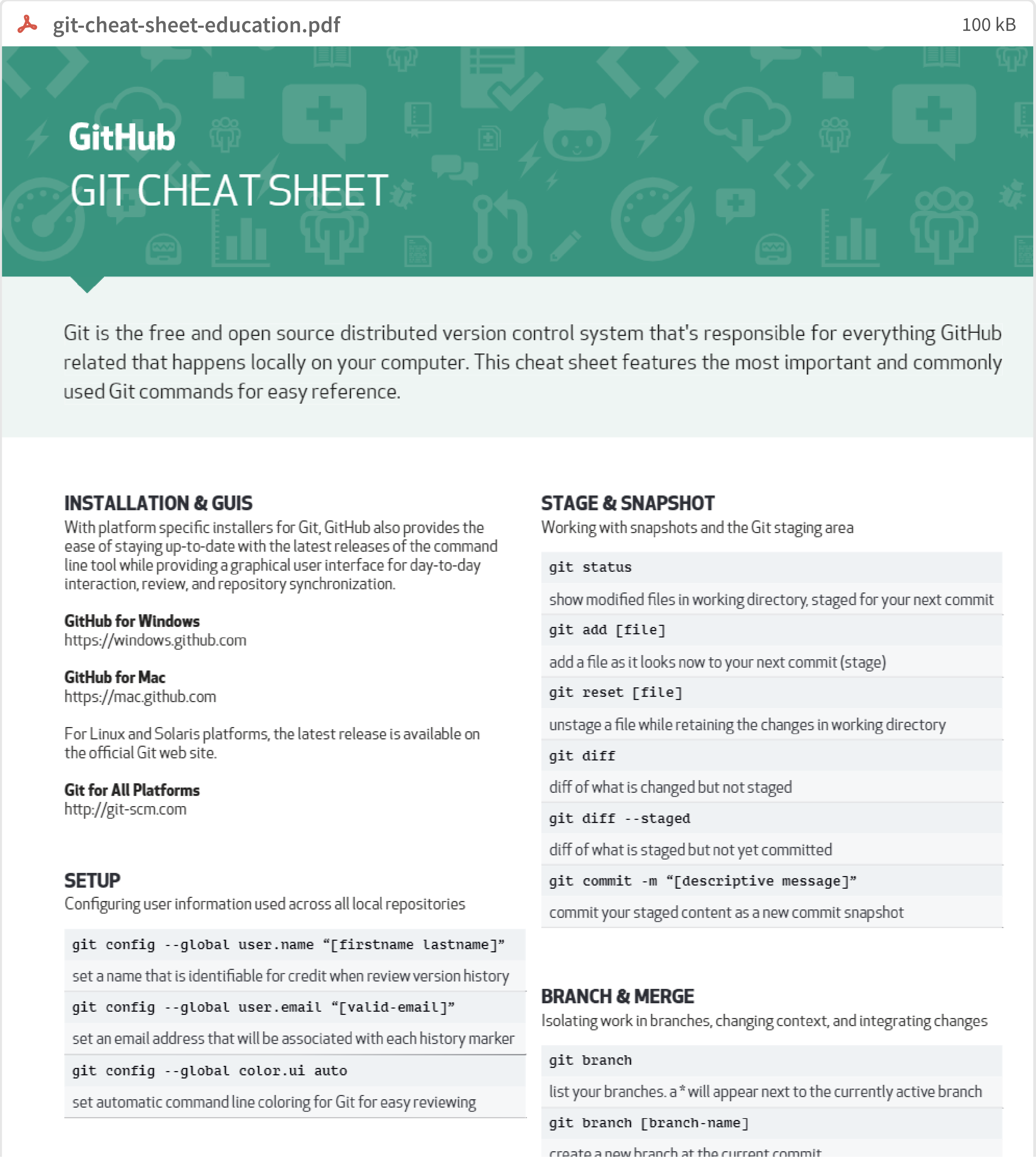
Команды Git



git-cheat-sheet-education.pdf 100 kB

GitHub
GIT CHEAT SHEET

Git is the free and open source distributed version control system that's responsible for everything GitHub related that happens locally on your computer. This cheat sheet features the most important and commonly used Git commands for easy reference.

INSTALLATION & GUIS

With platform specific installers for Git, GitHub also provides the ease of staying up-to-date with the latest releases of the command line tool while providing a graphical user interface for day-to-day interaction, review, and repository synchronization.

GitHub for Windows
https://windows.github.com

GitHub for Mac
https://mac.github.com

For Linux and Solaris platforms, the latest release is available on the official Git web site.

Git for All Platforms
http://git-scm.com

SETUP

Configuring user information used across all local repositories

```
git config --global user.name “[firstname lastname]”
```
set a name that is identifiable for credit when review version history

```
git config --global user.email “[valid-email]”
```
set an email address that will be associated with each history marker

```
git config --global color.ui auto
```
set automatic command line coloring for Git for easy reviewing

STAGE & SNAPSHOT

Working with snapshots and the Git staging area

```
git status
```
show modified files in working directory, staged for your next commit

```
git add [file]
```
add a file as it looks now to your next commit (stage)

```
git reset [file]
```
unstage a file while retaining the changes in working directory

```
git diff
```
diff of what is changed but not staged

```
git diff --staged
```
diff of what is staged but not yet committed

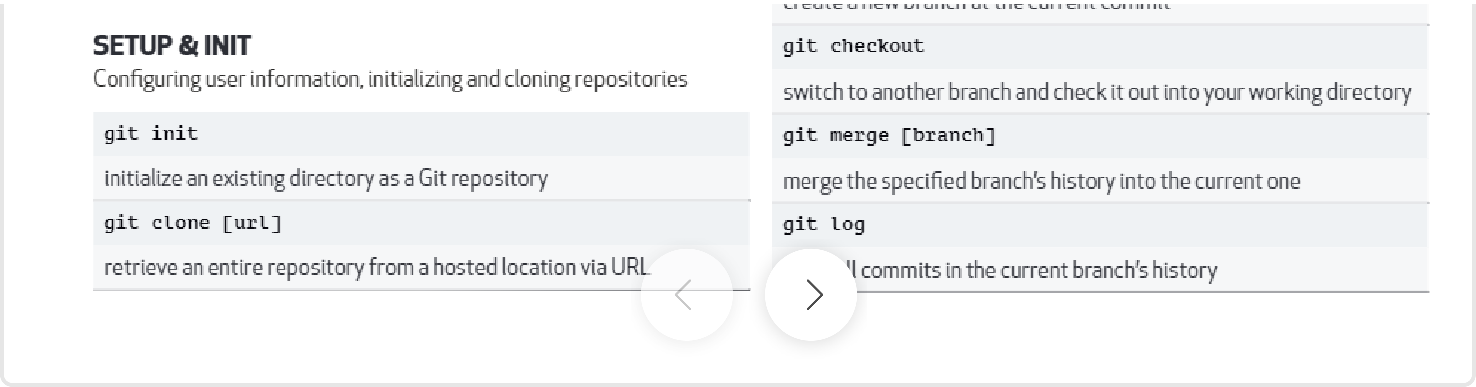
```
git commit -m “[descriptive message]”
```
commit your staged content as a new commit snapshot

BRANCH & MERGE

Isolating work in branches, changing context, and integrating changes

```
git branch
```
list your branches. a * will appear next to the currently active branch

```
git branch [branch-name]
```
create a new branch at the current commit



SETUP & INIT

Configuring user information, initializing and cloning repositories

```
git init
```

initialize an existing directory as a Git repository

```
git clone [url]
```

retrieve an entire repository from a hosted location via URL

create a new branch at the current commit

```
git checkout
```

switch to another branch and check it out into your working directory

```
git merge [branch]
```

merge the specified branch's history into the current one

```
git log
```

ll commits in the current branch's history