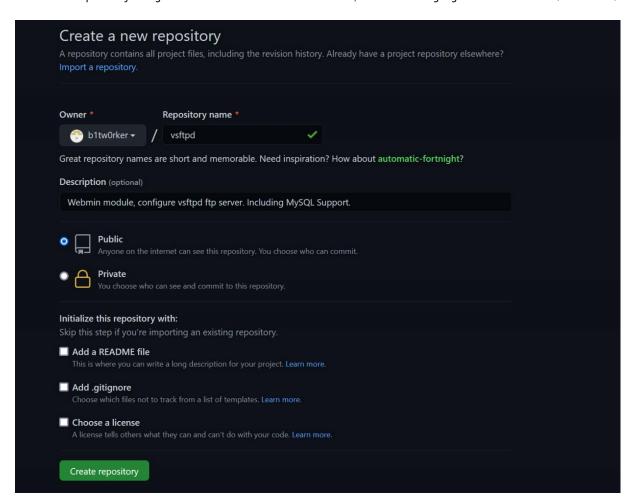
## Ein existierendes Projekt nach Github migrieren

Ein neues Repository auf github.com erstellen. Nicht README, license oder .gitignore initialisieren (Siehe Bild)



**Achtung:** github liefert nach Erstellung eine Anleitung mit "main" als branch. Wir wollen aber "master" als branch. Entsprechend ist das Script angepasst.

cd /lokales/projektverzeichnis

```
$ echo "# vsftpd" >> README.md
$ git init
$ git add README.md
$ git commit -m "first commit"
$ git branch -M master
$ git remote add origin https://github.com/bltw0rker/name-des-repository.git
$ git push -u origin master
```

Em Ende fragt github.com nach Usernamen und Password, falls es nicht im askpass verfügbar ist.

```
[root@centos deploy]# echo "# docs" >> README.md
[root@centos deploy]# git init
Leeres Git-Repository in /root/docs/deploy/.git/ initialisiert
[root@centos deploy]# git add README.md
[root@centos deploy]# git commit -m "first commit"
[master (Root-Commit) a810736] first commit
1 file changed, 1 insertion(+)
    create mode 100644 README.md
[root@centos deploy]# git branch -M master
[root@centos deploy]# git remote add origin https://github.com/bltwOrker/docs.git
[root@centos deploy]# git push -u origin master

(gnome-osh-askpass:29628): Gtk-WARNING **: 16:55:50.003: cannot open display:
    error: unable to read askpass response from '/usr/libexec/openssh/gnome-ssh-askpass'
Username for 'https://github.com': bltwOrker

(gnome ssh-askpass:29629): Gtk-WARNING **: 16:55:56.708: cannot open display:
    error: unable to read askpass response from '/usr/libexec/openssh/gnome-ssh-askpass'
    Password for 'https://bltwOrker@github.com':
Objekte aufzählen: 3, Fertig.
```

## Git.sh

Mit diesem kleinen Script kannst du Änderungen automatisch nach github.com auschecken.

```
$ cd /dein/loklaer/projektordner
$ git add *
$ git commit -m "initial version"
$ git push origin master
$ exit 0
```

Füge in .gitignore ,git.sh' zu, damit die Datei nicht mit ins Repository ausgecheckt wird.

Wenn das Passwort nicht erneut eingegeben werden soll, kann das mit folgendem Befehl gelöst werden:

```
$ git config credential.helper store
```

Git wird die benötigten Passwörter aus dem loklane, globalen Cache holen.