

/Sei (k)ein Blödmann und nutze **Git!**

or how to git gud



git



/Inhaltsverzeichnis



/01 /GESCHICHTE

> Wo kommt Git her?

/02 /FUNKTION

> Was kann Git?

/03 /GIT HOSTING SERVICES

> GitHub, Gitlab,
Bitbucket



/01

Geschichte

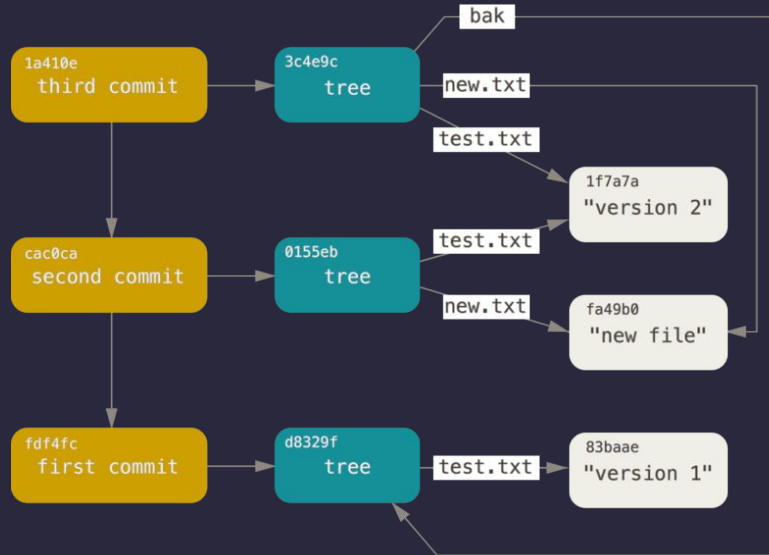
Wo kommt git her?



→ /Blödmann:

Git ([git], engl. *Blödmann*) ist eine **freie Software** zur verteilten **Versionsverwaltung von Dateien**, die von **Linus Torvalds** ursprünglich für die Quellcode-Verwaltung des Linux-Kernels entwickelt wurde.





/02

Funktion

Was kann Git?

**/Git ist eine
Versionsverwaltung.**

/Beispiel <init>



iterator.py



```
1 #should print numbers 0 to 10
2 counter = 0
3 while (counter < 10):
4     counter += 1
5     print(counter)
6
```

cmd



```
$ git init
> Initialized empty Git
  repository in E:/programming/oenings/.git/

$ git add .\iterator.py

$ git commit -m 'add iterator'
> [master (root-commit) e6223e4] add iterator
1 file changed, 5 insertions(+)
create mode 100644 iterator.py
```

/Beispiel <diff>

iterator.py



```
1 #should print numbers 0 to 10
2 counter = 0
3 while (counter ≤ 10):
4     print(counter)
5     counter += 1
6
```



cmd



```
$ git diff .\iterator.py
> diff --git a/iterator.py b/iterator.py
index 36646c3..3d6bc4a 100644
--- a/iterator.py
+++ b/iterator.py
@@ -1,5 +1,5 @@
 #should print numbers 0 to 10
 counter = 0
-while (counter < 10):
-    counter += 1
+while (counter ≤ 10):
+    print(counter)
+    counter += 1

$ git add .\iterator.py
$ git commit -m 'hotfix'
> [master e8311b1] hotfix
1 file changed, 2 insertions(+), 2 deletions(-)
```


/Beispiel <log && restore>

cmd

— □ ✕

```
$ git log
> commit e8311b165d4f499519df6a60c89161ded0c32ea9
Author: Benjamin Boll <github@b-sqr.de>
Date: Mon Feb 14 17:36:38 2022 +0100
```

hotfix

```
commit e6223e4305b1bc2782d10dfd69e7ce39e1a48f3c
Author: Benjamin Boll <github@b-sqr.de>
Date: Mon Feb 14 17:00:15 2022 +0100
```

add iterator

```
$ git restore -s e6223e4305b1bc2782d10dfd69e7ce39e1a48f3c -SW iterator.py
```



iterator.py

— □ ✕

```
1 #should print numbers 0 to 10
2 counter = 0
3 while (counter < 10):
4     counter += 1
5     print(counter)
6
```

/Beispiel <diff>

iterator.py



```
1 #should print numbers 1 to 10
2 counter = 0
3 while (counter < 10):
4     counter += 1
5     print(counter)
6
```



cmd



```
$ git diff
> diff --git a/iterator.py b/iterator.py
index 3d6bc4a..c570625 100644
--- a/iterator.py
+++ b/iterator.py
@@ -1,5 +1,5 @@
-#should print numbers 0 to 10
+#should print numbers 1 to 10
   counter = 0
-while (counter <= 10):
-    print(counter)
+while (counter < 10):
+    counter += 1
+    print(counter)

$ git add ./iterator.py
$ git commit -m 'adapt task'
> [master 53cef7c] adapt task
1 file changed, 3 insertions(+), 3 deletions(-)
```

/03

GIT HOSTING SERVICES

Die *verteilte* Versionsverwaltung



/Vergleich



/GitHub

+85M User

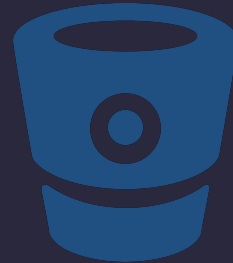
+282M Repositories
(+50M davon sind public)



/Gitlab

+0.1M User

+0.5M Repositories



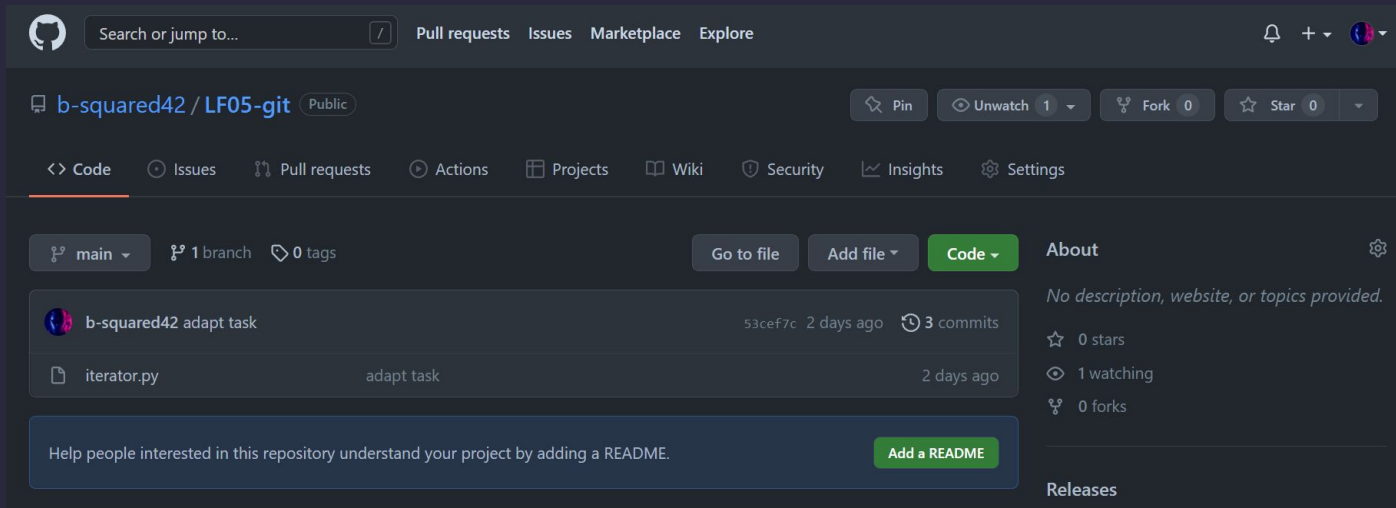
/Bitbucket

+10M User

+28M Repositories



/Working with GitHub



/Working with GitHub

\$ clone

\$ push

\$ pull

/Working with GitHub

\$ clone →

\$ push

\$ pull

```
cmd
$ git clone git@github.com:b-squared42/LF05-git.git
> Cloning into 'LF05-git'...
  remote: Enumerating objects: 9, done.
  remote: Counting objects: 100% (9/9), done.
  remote: Compressing objects: 100% (5/5), done.
  Receiving objects: 100% (9/9), done.
  Resolving deltas: 100% (1/1), done.
  remote: Total 9 (delta 1), reused 9 (delta 1), pack-reused 0
```

/Working with GitHub

\$ clone

\$ push



\$ pull

```
cmd
$ git push
> Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 307 bytes | 307.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:b-squared42/LF05-git.git
53cef7c..1f1c397 main -> main
```


/Working with GitHub

\$ clone

\$ push

\$ pull



```
cmd
$ git pull
> remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 678 bytes | 1024 bytes/s, done.
From github.com:b-squared42/LF05-git
    1f1c397..d4c198d  main      -> origin/main
Updating 1f1c397..d4c198d
Fast-forward
 iterator.py | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

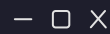
**/Sei (k)ein
Blödmann und
stelle Fragen!**

now u git gud.





BENJAMIN BOLL



<zwischenpeicher>

https://carbon.now.sh/?bg=rgba%28171%2C184%2C195%2C0%29&t=material&wt=boxy&l=python&width=680&ds=false&dsoff=20px&dsblur=68px&wc=true&wa=true&pv=56px&ph=56px&ln=true&fl=1&fm=JetBrains+Mono&fs=14px&lh=133%25&si=false&es=4x&wm=false&code=

https://carbon.now.sh/?bg=rgba%28171%2C184%2C195%2C0%29&t=vscode&wt=boxy&l=application%2Ftext%2Fsh&width=364&ds=false&dsoff=20px&dsblur=68px&wc=true&wa=true&pv=56px&ph=56px&ln=false&fl=1&fm=JetBrains+Mono&fs=14px&lh=133%25&si=false&es=4x&wm=false&code=%2524%2520git%2520init%250A%253E%2520initialized%2520empty%2520git%250A%2520repository%2520in%2520E%253A%252Fprogramming%252Foenings%252Fgit%252F%250A%2520%2520%250A%2524%2520git%2520add%2520%255Citerator.py