

Notwendige Dateien auf <https://github.com/bugfrei/GitDemo>

Git Übungen und Beispiel

2022 Carsten Schlegel

1 Vorbereitungen

Das PowerShell-Script in der PowerShell laden:

```
PS> . ./gitdemo.ps1
```

Wichtig ist der allein stehende Punkt. Damit wird das Script nicht ausgeführt sondern nur geladen. Anschließend sind alle darin enthaltenen Funktionen/Cmdlets jederzeit ausführbar.

1.1 Node.js

Die hier enthaltenen Quellcode-Dateien nutzen Javascript. Diese können (wenn gewünscht) mit node.js ausgeführt werden. Daher muss node.js installiert sein.

Am besten mit Chocolatey: <https://chocolatey.org/install>

Anschließend: `PS> choco install nodejs-lts`

Notwendige Dateien auf <https://github.com/bugfrei/GitDemo>

2 Git Rebase $\leftarrow \rightarrow$ Merge

Vorbereitung:

In einen beliebigen Ordner gehen. Notfalls geht einfach ~ (`cd ~`).

Den Befehl **Demo-Rebase** aus dem gitdemo.ps1-Skript ausführen.

Anschließend ist der aktuelle Ordner bereits der Übungsordner. Zurück setzen mit

```
PS> cd ..
```

```
PS> Demo-Rebase
```

Danach kann man erneut alles ausprobieren.

2.1 Ausgangssituation

Es gibt eine Demo.js in der Quellcode steht der die Zahlen 0...99 ausgibt.

Es gibt bereits 3 Commits, diese sind für den schnellen Zugriff mit den Tag „1“, „2“ und „3“ versehen.

2.2 Ziel

Es sollen die Zahlen 1 bis 10 ausgegeben werden:

```
start = 1;  
ende = 10;  
for (var i = start; i <= ende; i++)
```

2.3 Änderungen

Wir wechseln in den Commit mit Tag „1“. (`git checkout 1`)

Dort wurde der Fehler gemacht.

Wir ändern die oben angegeben Veränderungen in der demo.js. Diese sieht dann so aus:

```
var start = 1;  
var ende = 10;  
for (var i = start; i <= ende; i++)  
{  
    console.log(i);  
}
```

Nun haben wir die Änderung zwar gemacht, aber nicht committed.

Wir adden in den Stage (`git add .`) und comiten (`git commit -m "Bugfix"`).

Unser Log sieht nun so aus:

```
PS> git log --oneline --graph --all
```

Notwendige Dateien auf <https://github.com/bugfrei/GitDemo>

```
* c157fc7 (HEAD) Bugfix
| * 1c8f656 (tag: 3, main) Bereich ausgeben
| * 6bb10f4 (tag: 2) Meldung
|/
* bb66276 (tag: 1) Initial - 0 bis <100
(END)
```

Unser Bugfix befindet sich im oberen Commit, der bezieht sich auf tag: 1. Er enthält also nicht die Änderungen, die in tag: 2 und tag: 3 erstellt wurden.

2.4 Mergen

Um den Bugfix, der in tag: 1 erstellt wurde und die Änderungen von tag: 2 und 3 zu vereinen, kann gemerged werden:

Dies kann mit eines der folgenden Anweisungen durchgeführt werden:

```
PS> git merge 1c8f656
```

```
PS> git merge main
```

```
PS> git merge 3
```

Durchgeführt werden. Also entweder mit dem Hash von Commit tag: 3. Oder mit der Position des main-Branches. Da ein tag existiert, kann diese auch verwendet werden.

Danach verlangt git die Message für den Merge-Commit im Editor. So lassen oder ändern und speichern.

```
* 48247f6 (HEAD) Merge branch 'main' into HEAD
|\
| * 1c8f656 (tag: 3, main) Bereich ausgeben
| * 6bb10f4 (tag: 2) Meldung
| * | c157fc7 Bugfix
|/
* bb66276 (tag: 1) Initial - 0 bis <100
(END)
```

Ist nun das Ergebnis.

Die Datei demo.js beinhaltet nun den Bugfix und die Änderungen von tag 2 und tag 3:

```
var start = 1;
var ende = 10;
for (var i = start; i <= ende; i++)
{
    console.log(i);
}
console.log('Zahlen ausgegeben')
console.log(`Von ${start} bis <${ende}`);
```

Notwendige Dateien auf <https://github.com/bugfrei/GitDemo>

2.5 Rebase

Wie ausgehend von 2.3 Änderungen (direkt nach dem commit „Bugfix“) führen wir ein Rebase anstelle der merge aus.

Einer der folgenden Anweisungen ausführen:

```
PS> git rebase 1c8f656
```

```
PS> git rebase main
```

```
PS> git rebase 3
```

Danach ist auch alles zusammengeführt. Es wird aber kein neuer Commit erstellt. Das Log sieht nun so aus:

```
* 7bae1d8 (HEAD) Bugfix
* 12fe945 (tag: 3, main) Bereich ausgeben
* 740eb8c (tag: 2) Meldung
* 79969c5 (tag: 1) Initial - 0 bis <100
(END)
```

Der Code ist ebenfalls Zusammengeführt:

```
var start = 1;
var ende = 10;
for (var i = start; i <= ende; i++)
{
    console.log(i);
}
console.log('Zahlen ausgegeben')
console.log(`Von ${start} bis <${ende}`);
```