

DSCB 230 - Aufgabenblatt 11

Die Aufgabenblätter werden in dieser Veranstaltung in Jupyter Notebooks veröffentlicht und bearbeitet. Diese finden Sie in der Github Organisation für Data Science 2 unter dem Repository *dscb230-tutorial* (<https://github.com/hka-mmvmv/dscb230-tutorial>). Die Musterlösung wird ebenfalls in Form eines Jupyter Notebook in Github hochgeladen.

Aufgabenteil 1

Heute lernt ihr Git. Erarbeitet folgende Arbeitsschritte in einem Tool (Terminal, Git-Desktop-UI Anwendung, ...) eurer Wahl. Die Musterlösung dieser Aufgabe beinhaltet die Lösung mittels dem Terminal.

1. Erstellt euch ein Github Profil (mit der HKA habt ihr Github Pro)
2. Erstellt euch ein Github Repository mit folgenden Eigenschaften:
 - Name: HelloWorld
 - Description: Das Effizienteste Hello-World Programm dieser Erde
 - Privat auswählen
 - README auswählen
 - .gitignore auswählen
 - Wähle Python3 aus
 - Lizenz auswählen
 - Wähle MIT aus
3. Klont euer neues Repository auf eure Lokale Maschine

```
$ git clone https://github.com/<username>/HelloWorld
```

4. Navigiert mit einem Editor eurer Wahl in den neuen Ordner

```
$ cd HelloWorld
```

5. Erstellt eine Python Datei
 - Name: hello-world.py
 - Inhalt: `print("Hello, World!")`

```
$ touch hello-world.py  
$ echo 'print("Hello, World!")' > hello-world.py
```

6. Pusht euren Inhalt in den Main Branch

```
$ git add .  
$ git commit -m „Create Hello World“  
$ git push
```

7. Erstellt einen neuen Branch und wechselt da hinein

- Name: `Optimized_HelloWorld`

```
$ git branch Optimized_HelloWorld  
$ git checkout Optimized_HelloWorld
```

8. Verändert nun die Datei wie folgt

- Inhalt: `print(„Hello, World!\n“ * 100)`

```
$ echo ,print(„Hello, World!“ * 100)‘ > hello-world.py
```

9. Pusht nun auch diesen Inhalt in den entsprechenden Branch

```
$ git add .  
$ git commit -m „Create a Better Hello World“  
$ git push
```

10. Vollzieht nun einen Merge durch.

```
$ git checkout main  
$ git merge Optimized_HelloWorld  
$ git push
```

11. Verändert nun die `.gitignore` Datei sodass alle Python3 Dateien nicht mehr erkannt werden, denkt dran, alle bisherigen Dateien zu löschen.

```
$ git rm *.py  
$ git commit -m „Remove all Python Files“  
$ git push  
$ echo „*.py“ >> .gitignore  
$ git add .gitignore  
$ git commit -m „Create .gitignore“  
$ git push
```