Erläutere die folgenden Fragen:

1. Folgende vier Zustände kann eine Datei in Git haben (Tracked, Untracked, Modified, Unmodified, Staged)

Zeichne den zeitlichen Ablauf inkl. der Aktionen auf damit eine Datei durch diese Zustände läuft.

Tracked = Datei die in der Git Bash hinzugefügt wurde

Untracked = wenn Dateien von außen hinzugefügt werden

Modified = Datei wurde verändert

Unmodified = datei wurde nicht verändert

Staged = Dateien die mit git add in den staged bereich hinzugefügt wurden

1. Mit welchem Befehl kommt eine Datei in dem Staging Bereich

git add „Dateiname“

git add . wenn mehrere Dateien

1. Was versteht man unter dem „commit“ und wohin kommen dann diese Dateien und wie kann man diese mit einem Befehl „sichtbar machen“

Commit speichert Dateien im Repository

git lock

1. Was könnte ein Branch sein

Abteil

1. Warum gibt es die Befehle git diff und git diff –staged

Mit git diff sieht man den Unterschied zwischen einer alten und neuen Dateiversion wird angezeigt

Mit git diff -staged vergleichst du die stageded Variante mit der im repository

Praktischer Teil – bitte im ersten Schritt nur mit der Konsole „GitBash“ erfüllen (kopiere deine Ergebnisse hier her) – Markiere wenn unklar oder unvorhersehares passiert

1. Erstelle ein neues Verzeichnis „meinTestGit“
2. Intitalisiere das Verzeichnis für Git
3. Lege eine index.html und eine myCss.css hinein (notepad)
4. Schau dir den Status an (Status kopieren)
5. Bring die Dateien in den Stagebereich (Status kopieren)
6. Hol eine Datei wieder aus den Stagebereich (Status kopieren)
7. Ändere, Adde die Datei (Status kopieren)
8. Comitte die Datei (Status kopieren)
9. Ändere die beiden Dateien , Adde die Datei(en) (Status kopieren)
10. Comitte die Dateien
11. Git log ansehen
12. Checkout von ersten Commit
13. Änderungen durchführen und dann verwerfen
14. Checkout in den zweiten Commit

1. Lege einen Github User an (sofern nicht vorhanden)
2. Lege lokal den User in die config Datei von Git und deine Mailadresse
3. Lege ein leeres Repository in Github an
4. Führe die Schritte in der Konsole aus das Repository mit deinen Dateien zu füllen
5. Ändere im Repository mit dem dortigen Editor die Dateien
6. Pulle die Dateien wieder nach Lokal
7. Gitlog

HÜ:

1. Definiert mit dem Nachbar eines als eurer gemeinsames Repsoitory
2. Ändert jeder eine Datei und Pushed die Dateien
3. Pulled die Dateien und ihr solltet keine Konflikte haben
4. Ändert jeder die gleiche Datei. Einer pushed.
5. Der andere „Pulled“ die Dateien und ihr solltet einen Konflikte haben
6. Gemeinsames Lösen der Probleme