

Web-Anwendungen

Meilenstein 1

01.10.2014

Abgabe: bis zum 06.10.2014, 18 Uhr.

Lernziele

- Anlegen eines Repository
- Verwalten eines Repository (Commit, Push, Pull, Clone...)
- Erste Erfahrungen mit HTML

Hinweise zur Bearbeitung und Abgabe aller Übungsblätter

- Sie arbeiten in Teams. D.h., ihr arbeitet zusammen und erstellt zusammen selbstständig eine Lösung.
- Am besten ist es, wenn Sie während der Labor-Übungen einen eigenen Laptop (bzgl. der Admin-Rechte) mitbringen.
- Ihre Lösungen werden von mir regelmäßig überprüft und ein Review durchgeführt.
- Unzureichende Lösungen sind nachzuarbeiten. Bei wiederholtem Nachbearbeitungsbedarf oder identischen Lösungen ist die Erbringung der Studienleistung aller beteiligten Studierenden gefährdet.

Aufgabe 1

Erstellen Sie sich ein Repository (Pro Gruppe eines).

Entweder Git:

<https://github.com/>

<https://bitbucket.org>

Oder Subversion:

<http://projectlocker.com>

Falls Sie Git oder SVN noch nicht auf Ihrem Rechner installiert haben müssen Sie das als erstes tun. (Auf den Pool Rechnern wird das leider nicht funktionieren, da Sie dort keine Admin Rechte haben).

Legen sie für jedes Gruppenmitglied entsprechende Rechte an.

Zum Nachlesen:

Git: <http://git-scm.com/book/de>

SVN: <http://svnbook.red-bean.com/>

Hinweise:

- Um den Umgang mit einem Repository zu erleichtern dürfen sie auch gerne Plugins für eine IDE ihrer Wahl verwenden.
- Ebenso können graphische Benutzeroberflächen benutzt werden: <http://tortoisesvn.net/>
- Überlegen Sie sich eine geeignete Ordner Struktur für ihr Repository.
- Das Repository sollte über das Semester hinweg stabil bleiben. Es ist maximal eine Änderung erlaubt. Diese ist per E-Mail zu übermitteln.

Aufgabe 2

Üben Sie den Umgang mit einem Repository: (siehe dafür Anhang als Hilfe)

Rechner A: Fügen Sie eine Datei *test.html* dem Repository zu und checken Sie es ein.

Rechner B: Checken Sie das gesamte Repository nun aus (inklusive Datei *test.html*).

Öffnen Sie die Datei *test.html* und fügen sie dort folgende Zeilen ein:

```
<!DOCTYPE html>
  <html>
    <head>
      <title></title>
    </head>
    <body>
      Hallo Welt!
    </body>
  </html>
```

Und speichern Sie die Datei.

Überprüfen Sie anhand `svn/ git diff` Ihre Änderungen und mit `svn / git status` welche Datei geändert wurde.

Checken Sie die Änderungen ein (Denken Sie an eine sinnvolle Commit-Message!).

Rechner A: Updaten Sie Ihr Repository.

Aufgabe 3

Erstellen Sie eine HTML-Seite namens *index.html* und checken Sie diese in Ihr Repository ein.

Verzichten Sie derzeit noch auf umfangreiche Gestaltungen auf der Basis von Vorkenntnissen. Die Seite muss vom W3C-Validator (<http://validator.w3.org/>) als gültig („grün“) validiert werden. Als Seitentitel wählen Sie Ihre Matrikelnummern.

Der Inhalt soll eine Überschrift (ihr Gruppenname) beinhalten und einen Text (Ihrer Wahl) indem Abschnitte enthalten sind und verschiedene Wörter **fett** und *kursiv* formatiert sind.

Erstellen Sie zudem darunter eine nummerierte Aufzählungsliste mit Ihren Namen und Geburtstagen.

Optional: Recherchieren Sie nach geeigneten Browser-Erweiterungen, die Web-Entwicklern das Leben leichter machen.

Anhang:

Git:

The most commonly used git commands are:

add	Add file contents to the index
branch	List, create, or delete branches
checkout	Checkout a branch or paths to the working tree
clone	Clone a repository into a new directory
commit	Record changes to the repository
diff	Show changes between commits, commit and working tree, etc
fetch	Download objects and refs from another repository
grep	Print lines matching a pattern
init	Create an empty git repository or reinitialize an existing one
log	Show commit logs
merge	Join two or more development histories together
mv	Move or rename a file, a directory, or a symlink
pull	Fetch from and merge with another repository or a local branch
push	Update remote refs along with associated objects
reset	Reset current HEAD to the specified state
rm	Remove files from the working tree and from the index
show	Show various types of objects
status	Show the working tree status
tag	Create, list, delete or verify a tag object signed with GPG

SVN:

add	Add file contents to the index
checkout (co)	Checkout a branch or paths to the working tree
commit (ci)	Record changes to the repository
copy (cp)	Copy the file or directory
delete (del, rm)	Remove files from the working tree and from the index
diff (di)	Show changes between commits, commit and working tree, etc
help (?, h)	Show the help File for svn commands
merge	Join two or more development histories together
mkdir	Add directory to the index
move (mv, ren)	Move or rename a file, a directory, or a symlink
revert	Reset current HEAD to the specified state
status (st)	Show the working tree status
update (up)	Update the local files from the repository