

Kommandozeile

Alle Befehle können `--help`

`ls` - Auflisten der Ordnerinhalte "list"

`pwd` - "Print Working Directory" = "Drucke Arbeitsverzeichnis"

`cd` - "Change Directory" = "Wechsele verzeichnis"

`cd ..` - gehe ein Verzeichnis zurück (`.` ist das Aktuelle Verzeichnis, `..` das darüber)

`find` - alle existierenden Dateien auflisten

`grep -r "===="` - nach `=====` in allen Dateien Suchen (nützlich bei Merge)

`firefox index.html &` - mit Firefox die Datei `index.html` im Hintergrund starten (`&`)

`gedit index.html &` - `index.html` mit `gedit` editieren.

`nano index.html` - `index.html` in der Kommandozeile editieren mit `nano`

`vi index.html` - `index.html` in der Kommandozeile editieren mit `vim`

Git

`git --help` - alle git-Befehle anzeigen. Mach dich damit vertraut

`git commit --help` - Hilfe zu `commit` anzeigen

`git add index.html` - `index.html` dem INDEX hinzufügen

`git add .` - den momentanen Ordner mit allen Änderungen zum INDEX hinzufügen

`git add --all .` - alle Dateien, auch gelöscht, in den INDEX tun

`git remove index.html~` - die Löschung einer Datei auch in den INDEX tun

`git commit -m"Nachricht"` - Den INDEX zu einer neuen Version machen

`git commit -am"Nachricht"` - alle von git erfassten Dateien in den Index tun und mit Nachricht zu neues Version machen

`git push` - ins Quellrepository hochladen

`git push origin master` - in das Repository `origin` in den Branch `master` hochladen

`git pull` - vom Quellrepository Änderungen runterladen

`git branch patch-1` - den Branch `patch-1` erstellen. `master` heißt der Standardbranch

`git checkout patch-1` - in den Branch `patch-1` wechseln

`git branch` - alle Branches anzeigen und auf welchem man ist `git merge master` - den Branch `master` in den momentanen Branch einpflegen

Mergekonflikte beheben

1. `git status` - Zustand, wie ihn git sieht, anzeigen

2. die Dateien mit Mergekonflikten editieren, suchen nach `<<<<`, `====`, `>>>>`

3. `git add` - Dateien hinzufügen

4. `git commit -m"Nachricht"` - Zusammenführung bestätigen und neue Version machen

HTML

Einrückungen sind wichtig für Menschen! Beispielwebseite:

```

<html>
  <head>
    <meta charset="UTF-8" />
    <title>Chemnitzer Linxstage 2016</title>
    <link href="index.css" rel="stylesheet" />
  </head>
  <body>
    <h1>Chemnitzer Linxstage 2016</h1>
  </body>
</html>

```

`<html> </html>` - darin steht alles zur Webseite

`<head> </head>` - darin stehen Informationen über die Webseite

`<body> </body>` - darin stehen Informationen über die Webseite

In head:

`<meta charset="utf-8" />` - UTF-8 als Kodierung der Datei verwenden (alle Unicode Zeichen, beste Wahl)

`<link href="style.css" rel="stylesheet" type="text/css" />` - Cascading Style Sheet in der Datei `style.css` einbinden

`<link rel="icon" type="image/png" href="favicon.png">` - Webseitenlogo

`<script type="text/javascript" src="trallala.js" crossorigin="anonymous"></script>` - die Datei `trallala.js` als JavaScript einbinden

`<title> </title>` - darin steht der Title der Webseite

In body:

`<h1> </h1>` - darin stehen Überschriften (`h1`, `h2`, `h3`, `h4`) `<!-- -->` - darin stehen Kommentare für Programmierer, die nicht angezeigt werden

`<div> </div>` - darin steht ein Abschnitt. Divs eignen sich gut zum Stylen, siehe CSS.

`<p> </p>` - darin steht ein Absatz

`
` - ein Zeilenumbruch

`Bin<wbr />de<wbr />strich` - Bindestriche, wenn das Wort "Bindestrich" umgebrochen werden muss wegen Platz

`Github Repository` - ein Link auf eine entfernte Webseite

`Home` - ein Link auf die unterste `index.html`-Datei (man beachte den `/` am Anfang)

`Test` - ein Link auf eine `test.html`-Datei im selben Ordner (kein `/`).

`` - das Bild `test.png`, hier auch im selben Ordner

Es gibt viele Unicodezeichen: tinyurl.com/cyztmr

CSS

Elemente auswählen

- `<div class="content">` - Jedes Element, `div`, `h1`, `a`, `p`, usw. kann eine Klasse haben. Hier hat ein `div` Element eine Klasse "content". Elemente, die die selbe Klasse haben, nutzen den gleichen CSS-

Style, der hinter der Klasse liegt.

Schriftfarbe auf blau und Schriftgröße auf 20 Pixel für den Inhalt aller "content" setzen:

```
.content {
  color: blue;
  font-size: 20px;
}
```

- `<div id="einmal">` - "id" = "Identifizier" = "Identifizierer" sind nur für ein Element da. CSS zum setzen des Hintergrunds: `#einmal { background: black; }`
- `<h1> </h1>` - Alle Elemente eines Types können einen eigenen Stil bekommen.
Schriftart aller Hauptüberschriften auf "Roboto" setzen, wenn es "Roboto" nicht gibt, auf "Arial":
`h1 { font-family: Roboto,Arial; }`
- Mouseover-effekte wie, wenn die Maus drüber ist die SFarbe ändern
`a:hover { color: darkblue; }`

Eigenschaften `font-size: 15px;` - Schriftgröße auf 15 Pixel setzen

`color: #666A73;` - Schriftfarbe auf `#666A73` setzen

`letter-spacing: .5px;` - Abstand der Buchstaben auf 0,5 Pixel setzen

`width: 66.66667%;` - Breite auf 2/3 setzen

`height: 2em;` - Höhe doppelte Schriftgröße setzen

`padding: 10px;` - Innen größer machen (mit Hintergrund) `margin: 10px;` - Abstand größer machen (ohne Hintergrund)

`margin-right`, `margin-left`, `margin-top`, `margin-bottom` - spezielle Ränder, analog bei padding

Längenangaben

Es gibt z.B. `px`(Pixel), `%`(Anteil am Maximum), `em`(*Schriftgröße), tinyurl.com/ctyzh9z

Von anderen Webseiten abgucken

Rechtsklick auf ein Element → Inspect Element → Inspector → Rules