Die Kommmandozeile und du

2020-11-18T16:24:58+01:00

Table of Contents

# lernos-git-guide

Platzhalter für künftigen Content # lernos-git-guide

## Platzhalter für künftigen Content

title: “Arbeiten und Lernen in der Digitalität” date: 2020-11-15T16:24:58+01:00 draft: true

# Arbeiten und Lernen in der Digitalität

Gelesen hat es wohl jeder schon, die mahnenden Worte zur digitalen Transformation und wie sich unsere Welt revolutionär in kürzester Zeit ändert. Was heißt das nun konkret für uns als Mitglieder der digitalen Gesellschaft?

Als Konsument sind wir durch Internet und Handy Apps schon längst Teil des weltumspannenden Netzwerks von Services und Communities. Wir können liken, sharen, und unsere Übernachtung oder die nächste Taxifahrt online buchen, ohne dass wir uns dafür groß bemühen müssen.

Wenn wir etwas genauer in dieses Netzwerk schauen, wird schnell klar, dass es eine Ansammlung aus Millionen von einzelnen Bestandteilen ist, die alle “irgendwie” miteinander kommunizieren. Technische Standards sorgen dafür, dass sich diese Komponenten untereinander verstehen oder zumindest dieselbe Sprache sprechen. Wir nennen diese Komponenten der digitalen Welt “Services”.

Einzelne Services können wir uns dabei wie Web-Benutzer vorstellen, die eine Website aufrufen, Informationen erhalten oder versenden und so ihre Arbeit besser verrichten können.

Das Ergebnis der Arbeit ist dann entweder wieder Information (schließlich leben wir im Informationszeitalter), oder ein Produkt, das entsteht, eine Ware, die versendet wird, und so weiter.  
Netzwerk-Services sind also eine Gemeinschaft von Arbeitern, die miteinander kommunizieren - nur viel schneller und regelmäßiger als Menschen das können.

Wie entstehen diese Services? Millionen von Software-Entwicklern auf der ganzen Welt kennen diese Standards, tauschen sich darüber aus und entwickeln sie weiter. Am erfolgreichsten sind die Software-Services, die sich möglichst gut auf die standardisierten Übergabepunkte verstehen, die für andere leicht zu verstehen sind und die man idealerweise für seine eigenen Zwecke benutzen und weiterentwickeln kann.

Open Source hat viel zu dieser Entwicklung beigetragen. Software, die so lizensiert ist, dass jeder sie benutzen, verändern und weitergeben kann, hat geschafft, was die großen kommerziellen Systemanbieter nicht schaffen konnten - eine sich selbst verstärkende weltweite Gemeinschaft zu schaffen, die stetig neue Services erfindet, Bestehendes verbessert und immer wieder Hilfsmittel erfindet, diese vielen Einzelteile neu zu kombinieren.

Die großen Open Source Plattformen, allen voran github, haben sich zu den virtuellen Arbeits- und Marktplätzen weiterentwickelt. Hier lagern Millionen von öffentlich nutzbaren Softwareprojekten im Source Code, stetig begutachtet und weiterentwickelt durch Interessierte Entwickler.

Dieses Prinzip und auch diese Plattformen können wir auch für andere Prozesse als Softwareentwicklung einsetzen. So können wir zum Beispiel Redaktionsprozesse von Texten genauso ablaufen lassen wie Softwareentwickler ihren Code schreiben. Das mag erst einmal befremdlich klingen. Bei genauerem Hinsehen ist ein Redaktionsprozess jedoch sehr ähnlich zu Software-Entwicklung - nur dass eben kein Code erzeugt wird, sondern Texte und Bilder.

Wenn wir die Arbeit an einzelnen kleinen Textbausteinen in dem Mittelpunkt stellen und nicht vom Endergebnis her denken, können wir viel produktiver und effizienter den “Content” erzeugen, den wir haben wollen und die “Endmontage” in einen eigenen Teilprozess verlegen.

Das geht mit Markdown und Git ganz hervorragend. Markdown ist einfach, sehr schnell zu lernen und erlaubt die einfache Bearbeitung von Textteilen in den Montageschritten. Git ist unser Archiv und unser Workflowsystem, über das wir den Prozess ins Laufen bringen.

So wird unsere Textarbeit zu einem digitalen Service, den wir in den großen Strom der digitalen Produktion mit eingliedern können und mit anderen Teilen zu guten Ergebnissen verbinden können.

# lernos-git-guide

## Platzhalter für künftigen Content

title: “Arbeiten und Lernen in der Digitalität” date: 2020-11-15T16:24:58+01:00 draft: true

# Arbeiten und Lernen in der Digitalität

Gelesen hat es wohl jeder schon, die mahnenden Worte zur digitalen Transformation und wie sich unsere Welt revolutionär in kürzester Zeit ändert. Was heißt das nun konkret für uns als Mitglieder der digitalen Gesellschaft?

Als Konsument sind wir durch Internet und Handy Apps schon längst Teil des weltumspannenden Netzwerks von Services und Communities. Wir können liken, sharen, und unsere Übernachtung oder die nächste Taxifahrt online buchen, ohne dass wir uns dafür groß bemühen müssen.

Wenn wir etwas genauer in dieses Netzwerk schauen, wird schnell klar, dass es eine Ansammlung aus Millionen von einzelnen Bestandteilen ist, die alle “irgendwie” miteinander kommunizieren. Technische Standards sorgen dafür, dass sich diese Komponenten untereinander verstehen oder zumindest dieselbe Sprache sprechen. Wir nennen diese Komponenten der digitalen Welt “Services”.

Einzelne Services können wir uns dabei wie Web-Benutzer vorstellen, die eine Website aufrufen, Informationen erhalten oder versenden und so ihre Arbeit besser verrichten können.

Das Ergebnis der Arbeit ist dann entweder wieder Information (schließlich leben wir im Informationszeitalter), oder ein Produkt, das entsteht, eine Ware, die versendet wird, und so weiter.  
Netzwerk-Services sind also eine Gemeinschaft von Arbeitern, die miteinander kommunizieren - nur viel schneller und regelmäßiger als Menschen das können.

Wie entstehen diese Services? Millionen von Software-Entwicklern auf der ganzen Welt kennen diese Standards, tauschen sich darüber aus und entwickeln sie weiter. Am erfolgreichsten sind die Software-Services, die sich möglichst gut auf die standardisierten Übergabepunkte verstehen, die für andere leicht zu verstehen sind und die man idealerweise für seine eigenen Zwecke benutzen und weiterentwickeln kann.

Open Source hat viel zu dieser Entwicklung beigetragen. Software, die so lizensiert ist, dass jeder sie benutzen, verändern und weitergeben kann, hat geschafft, was die großen kommerziellen Systemanbieter nicht schaffen konnten - eine sich selbst verstärkende weltweite Gemeinschaft zu schaffen, die stetig neue Services erfindet, Bestehendes verbessert und immer wieder Hilfsmittel erfindet, diese vielen Einzelteile neu zu kombinieren.

Die großen Open Source Plattformen, allen voran github, haben sich zu den virtuellen Arbeits- und Marktplätzen weiterentwickelt. Hier lagern Millionen von öffentlich nutzbaren Softwareprojekten im Source Code, stetig begutachtet und weiterentwickelt durch Interessierte Entwickler.

Dieses Prinzip und auch diese Plattformen können wir auch für andere Prozesse als Softwareentwicklung einsetzen. So können wir zum Beispiel Redaktionsprozesse von Texten genauso ablaufen lassen wie Softwareentwickler ihren Code schreiben. Das mag erst einmal befremdlich klingen. Bei genauerem Hinsehen ist ein Redaktionsprozess jedoch sehr ähnlich zu Software-Entwicklung - nur dass eben kein Code erzeugt wird, sondern Texte und Bilder.

Wenn wir die Arbeit an einzelnen kleinen Textbausteinen in dem Mittelpunkt stellen und nicht vom Endergebnis her denken, können wir viel produktiver und effizienter den “Content” erzeugen, den wir haben wollen und die “Endmontage” in einen eigenen Teilprozess verlegen.

Das geht mit Markdown und Git ganz hervorragend. Markdown ist einfach, sehr schnell zu lernen und erlaubt die einfache Bearbeitung von Textteilen in den Montageschritten. Git ist unser Archiv und unser Workflowsystem, über das wir den Prozess ins Laufen bringen.

So wird unsere Textarbeit zu einem digitalen Service, den wir in den großen Strom der digitalen Produktion mit eingliedern können und mit anderen Teilen zu guten Ergebnissen verbinden können.

## Woche 0

### Check in / Who is who? Deine Ziele?

## Lorem Ipsum.

title: “Setup Tools” date: 2020-11-18T16:24:58+01:00 draft: false —

Wilkommen zu Woche 0 im git for kids circle

Zu Beginn müssen ein paar Accounts angelegt und Tools installieren. Hierbei gibt es immer mögliche Alternativen. Ein paar Alternativen werden wir hier vorstellen, Du kannst die kommenden Kapitel aber auch mit Tools Deiner Wahl verfolgen.

## Markdown

GitHub bietet das Textformat Markdown (.md) an. Dieses Format nutzt eine einfache Syntax die in erster Liene dafür gedacht ist Texte fürs Web zu schreiben.

Wir empfehlen Dir für alle Textdateien in GitHub Markdown zu verwenden.

Diese Taxte kannst Du direkt in GitHub anlegen oder in einem MarkDown Editor schreiben. Wir empfehlen z.B. [Mark Text](https://marktext.app/).

Eine Einführung zu Markdown erhälst Du in Woche 2.

## Die Kommandozeile und du

Wenn du im Kino eine Film siehst, in dem Computerspezialisten irgendetwas Geheimnisvolles erledigen, klicken sie meist nicht mit der Maus herum, sondern arbeiten mit Fenstern mit vielen Textausgaben, die am Bildschirm durchlaufen, nachdem sie kryptische Kommandos eingegeben haben. Das sieht spektakulär und kompliziert aus. Im echten Leben ist es meist nicht so geheimnisvoll. Die durchlaufenden Texte sind meist Statusmeldungen von Systemen und das Tastaturgeklapper kommt von der Eingabe von Befehle auf der so genannten Kommandozeile ein, wodurch die Profis ein Anwendungsprogramm oder eine bestimmte Datenverarbeitung starten.

Wieso gibt es überhaupt noch Kommondozeilen? Ist nicht die Arbeit mit der Maus viel einfacher und effizienter? Programme mit grafischen Benutzeroberflächen sind erst einmal intuitiver zu bedienen. Wer aber viele verschiedene Aufgaben durchführen muss und zudem viel schreibt, wird feststellen, dass es viel Zeit kostet, die Hände von der Tastatur zu nehmen und mit der Maus Aktionen durchzuführen. Und wenn zu viele Fenster verschiedener Apps geöffnet sind, wird es schnell unübersichtlich.

Daher benutzen viele Software-Arbeiter die Kommandozeile. Ein Befehl ist schnell getippt und die Liste vergangener Aktionen kann schnell abgerufen und erneut genutzt werden. Man erreicht alle Kommandos von der selben Stelle und kann bei einem geteilten Bildschirm schnell hin und her wechseln. Vielleicht am allerwichtigsten sind aber zwei weitere Eigenschaften:

1. Kommandos lassen sich verknüpfen. Die Ausgabe eines Programms kann direkt als Eingabe des nächsten Programms genutzt werden und so können ganze Workflows auf einer Zeile ablaufen. Diese Pipeline-Verarbeitung trifft man vor allem auf Linuxsystemen, die heute einen Großteil der Cloud-Computer ausmachen. So liefert die folgende Kette von Kommandos die 10 häufigsten Wörter dieses Artikels

cat kommandozeile.md | tr -cs A-Za-z ' ' | tr A-Z a-z | tr ' ' '\n' | sort | uniq -c | sort -rn | sed 10q

ergibt

16 und  
 16 die  
 11 der  
 8 mit  
 7 ist  
 7 es  
 6 von  
 5 viele  
 5 so  
 5 sind

1. Kommandos kann man nicht nur auf der Kommandozeile aufrufen, sondern in einer Textdatei sammeln und als Script immer wieder ablaufen lassen. so entstehen ganz einfach mächtige Programme, die viele Aufgaben automatisieren können - ähnlich wie Office-Makros, nur viel klarer strukturierbar.

Mit diesen Konzepten ist es in den letzten 50 Jahren gelungen, unsere weltumfassende digitale Vernetzung aufzubauen und weiterzuentwickeln. Auch deine Redaktionsaufgaben kannst du mit Kenntnissen aus diesem Bereich besser, einfacher und schneller erledigen. Nur Mut, es ist einfacher als es zunächst aussieht!

“Die Kenntnis dieser Werkzeuge markiert wohl die Trennlinie zwischen Techies und anderen Leuten (oder, um es mit magischen Begriffen zu sagen - zwischen Zauberern und Muggels)” – Michael Hartl

Dieses Zitat kommt aus dem (englischen) Selbstlern-Text “Learn Enough Command Line to be Dangerous”, der zwar leider nicht kostenfrei zugänglich ist, dessen offene erste Kapitel jedoch einen guten Einstieg in das Thema Kommoandozeile geben.

Tipp: Wenn ihr einen Windows-Rechner habt, installiert euch `git bash`. Das ist eine Kommondozeilen-App, die viele Linux-Befehle ohne großen Aufwand auch auf Windows-Rechnern zur Verfügung stellt. Die Windows-Tools "cmd" oder "Powershell" sind eher Exoten im Bereich der Kommandozeilen.

# lernos-git-guide

Platzhalter für künftigen Content # lernos-git-guide

Platzhalter für künftigen Content # lernos-git-guide

Platzhalter für künftigen Content