

Cloud Style Guide

28. März 2016

1 Quick-Start

Regel Nummer eins: viel Inhalt, wenig Bla Bla! Dies wollen wir hier auch einhalten. Regel Nummer zwei: sinnvolle Kommentare anbringen, Variablen+Dateien nachvollziehbar benennen, insgesamt auf Nachvollziehbarkeit achten. Regel Nummer drei: eigene nützliche Erkenntnisse den Anderen zur Verfügung stellen durch Aktualisieren des Kautschuk Repos. Die verwendeten Programme:

- Python 3.x
- LuaLaTeX
- make
- atom o.ä. Texteditoren (z.B. VIM, notepad++)

Der generelle Ablauf zum Erstellen eines Protokolls sollte sich wie folgt gestalten. Zunächst wird das verwendete Template (also die Vorlage) in einen vernünftig benannten Ordner (vorzugsweise /Praktikum/xxx mit xxx=Versuchsnummer) kopiert. Das Template ist im Kautschuk Repo zu finden und sollte bei sinnvollen Änderungen stets gepusht werden, sodass alle dieselbe Vorlage verwenden.

Die Daten aus den Messungen müssen ins Digitale übertragen werden. Dazu zwei Möglichkeiten:

- Manuelles Anlegen von .txt Dateien im Ordner "/Praktikum/xxx/Messdaten".
- Erstellen von .txt mit Hilfe von Python (der berühmte Datenersteller.py). Aber auch hierbei sollten die Daten in dem "/Praktikum/xxx/Messdaten" Ordner landen.

Das Erstellen der Messdaten sollte ein einmaliger Schritt sein. Wenn der Datenersteller verwendet wird, sollte aber der Ordner Messdaten beim Klonen des Repos mitkopiert werden, sodass dieser Arbeitsschritt im Nachhinein nicht wiederholt werden muss. Die .txt Dateien sollten ein sinnvolles Format haben. Zum Beispiel könnten die Dateien nach Aufgabenteilen benannt werden, z.B. "a.txt". Die erste Zeile einer jeden .txt Datei im Ordner Messdaten muss eine Beschreibung der Spalten samt Einheiten beinhalten, natürlich ist diese Zeile auszukommentieren. Z.B. "#Wert[-] R_2[Ohm] R_3/R_4[-]". Jeder soll wissen, was genau ihr dort eingetragen habt.

Der nächste Schritt ist die Bearbeitung der Datei "/Praktikum/xxx/PythonSkript.py", welche sich durch das Kopieren des Templates bereits an richtiger Stelle befindet. In diesem Skript sind die gängigen Code Examples als Kommentare hinterlegt, sodass diese direkt an die benötigte Stelle kopiert werden können. Nun einige Richtlinien zur Lesbarkeit eures Codes:

- Ganz oben steht, was alles importiert wird. Meistens sollte das reichen, was die Vorlage bereits vorgibt.
- Dann werden ggf. benötigte Funktionen deklariert mit einer sprechenden Namengebung, z.B. `def phi(param_a, Temperatur)`.
- Nun werden globale Konstanten deklariert und in einem Kommentar beschrieben, z.B. `R = const.physical_constants["molar gas constant"] #Array of value, unit, error`
- Erst jetzt werden die Aufgabenteile behandelt. Dazu wird eine kommentierte Abgrenzung verwendet und es folgt der Kommentar `#Aufgabenteil a)`.

- Variablen werden deklariert, häufig durch Einlesen der Messwerte, z.B. `c_W = np.genfromtxt('Messdaten/spezifische_Waermekapazitaet.txt', unpack=True)`. Auch hier sollten so wie überall die Variablennamen sprechend sein, dh. dem Leser anzeigen, was sie darstellen.
- Jetzt sollten ggf. die Variablen in SI Einheiten umgerechnet werden. Die jeweilige Einheit ist durch einen Kommentar anzuzeigen.
- Häufig erfolgt dann ein Curve-Fit. Dieser kann mit den bereit gestellten Funktionen durchgeführt werden, z.B. `params_max = ucurve_fit(reg.reg_linear, t_extern, U_pos_ges_log)`. In der Datei "Regression.py" sind die verschiedenen Fit-Funktionen vorgehalten.
- Die Ergebnisse der Regression werden in neue Variablen geschrieben, z.B. `R_a`, `Offset_a = params_max`. Dann werden sie über die Funktion `write(string filename, string content)` mit Hilfe der Funktion `make_SI(double value, unit, figures)` zur Wiedergabe im Tex-Dokument in den build-Ordner geschrieben, z.B. `write('build/R_daempfung_theo.tex', make_SI(R_1[0], r'\ohm', figures=1))`. Die Funktion `make_SI` ermöglicht die vernünftige Formatierung zur Ausgabe im PDF, vor Allem durch die Wahl der Nachkommastellen mit der Angabe von `figures`.
- Häufig werden Datensätze auch in Tabellen gespeichert. Hierzu empfiehlt sich für eine gute Formatierung die Nutzung der Funktion `make_full_table`, welche die Verwendung von `make_table` voraussetzt. Ein beschriebenes Beispiel hierzu findet sich in der PythonSkript Datei wieder. Es gibt außerdem eine kleine Doku dazu hier im Kautschuk Repo. Wichtig ist, dass `make_full_table` eine komplett fertig formatierte Tabelle ausgibt, die in Latex nur noch durch z.B. `\input build/Tabelle_b_texformat.tex` eingebunden werden muss. Label und Caption sind also jetzt nicht mehr in Latex, sondern in Python einzustellen.
- Ebenso finden sich für das Plotten von Daten eine Reihe von immer wieder verwendeten Funktionen in unserer Vorlage. Es ist z.B. möglich, die Arraygrenzen automatisch zu setzen. Hier gibt es im Rahmen der matplotlib sicher eine ganze Reihe von speziellen Anwendungen. Findet sich bei der Recherche etwas Nützliches, so sollte dies durch entsprechende Ergänzungen in dem Beispielblock allen im Kautschuk Repo verfügbar gemacht werden. Wir nutzen im Übrigen matplotlib, was eine vernünftige Formatierung von Grafiken ermöglicht, aber das Rechenzeit bei der Ausführung des Python Skripts deutlich erhöht. Zum schnellen Testen empfiehlt es sich, die Plots zunächst auszukommentieren. Wie die Plots ins tex Dokument eingebunden werden, kann den Beispielen in `"/Praktikum/xxx/content/auswertung.tex"` entnommen werden. Es gibt verschiedene Anordnungsoptionen.

Das Tex-Dokument soll dann jeder so erstellen, wie er es für richtig hält. Relevant für uns ist in erster Linie "PythonSkript.py". Trotzdem gibt es sicher den ein oder anderen, der euren gesamten Code gerne ausführen würde. Dazu gilt als Regel Nummer vier, dass das Klonen des Repos es einer beinhalteten MAKE-file ermöglicht, zu bauen, OHNE dass man noch irgend etwas tun muss! Die Standard-make file ist in der Vorlage enthalten. Schließlich wäre es noch nett, in den Ordner `"/Praktikum/xxx/final/"` die abtestierten Protokolle in PDF-Form zu legen. Zu guter letzt pusht ihr den xxx-Ordner an die entsprechende Stelle im Kautschuk Repo, sodass nun jeder Zugriff darauf hat und es von jedem Versuch möglichst schnell eine Referenz gibt.