

Übung 01: Rekapitulierung von PCL1 und Module der Standard-Library

Programmiertechniken in der Computerlinguistik II, FS 17

Abgabedatum: 7. März 2017, 23:59

Hinweise zur Abgabe

- Bitte gib jedes Python-Programm in einer eigenen Datei ab, die die Dateiendung `.py` hat und *ausführbaren* Python-Code enthält.
- Gib all deine Antworten, welche keine Skripts sind, in einem PDF-Dokument *vorname__nachname__uebung01.pdf* ab
- Geize nicht mit Kommentaren direkt im Programm-Code, wo Erläuterungen angebracht sind. Umfangreiche Erklärungen werden hingegen besser in einer separaten README-Datei mitgeliefert (vorzugsweise Plain-Text oder PDF).
- Um das Hochladen der Abgabe auf OLAT zu erleichtern, kannst du die Dateien mit **zip** (oder einem anderen verbreiteten Format) archivieren / komprimieren.

1 Coding-Stil und Programmierprinzipien

1.1

Lies dir den offiziellen Python Style-Guide ([Link](#)) und die Google-Ergänzungen dazu ([Link](#)) durch. Versuche, deinen Code wenn möglich danach zu gestalten. Alle Übungen in PCL II werden auch nach Coding-Stil bewertet.

Abzugeben: Hier muss nichts abgegeben werden.

1.2

Für die Lesbarkeit eines Programms ist nicht nur das Layout wichtig, sondern auch die Organisation des Codes. Ein wichtiges Prinzip ist das Prinzip der Modularität, also die Aufteilung eines Programms in kleinere, wohldefinierte Bestandteile. Es gibt aber auch andere solche “Prinzipien der Programmierung”. Recherchiere einige dieser Prinzipien im Internet (zum Beispiel auf Wikipedia: [Link](#)) und notiere drei, die dir besonders einleuchten. Weshalb sind sie wichtig?

Abzugeben: Eine PDF Datei.

1.3

Nimm ein altes Python-Skript von dir, zum Beispiel aus PCL I, und kritisiere es. Welche Stil- und Organisationsprobleme findest du? Das Skript sollte ungefähr 50-80 Zeilen lang sein. (Wenn du kein geeignetes Skript hast, darfst du auch eins von jemand anderem nehmen.)

Abzugeben: Dein gewähltes Python-Skript und eine PDF Datei mit deinen Kommentaren zu den Stil- und Organisationsproblemen.

2 Vergleich von Zeitungsartikeln

Im Rahmen eines kleinen Forschungsprojektes willst du Online-Artikel verschiedener Schweizer Zeitungen (Blick, Tages-Anzeiger, 20 Minuten und NZZ) miteinander vergleichen. Schreibe ein Modul welches die Funktion `analyse_newspaper(filename, encoding)` definiert. Diese Funktion sollte einen als Textdatei im lokalen Dateisystem gegebenen Artikel einlesen (gegeben als parameter `filename`) und die folgenden Eigenschaften bestimmt:

- Gesamtanzahl der Wörter
- Das längste und das kürzeste Wort
- Durchschnittliche Wortlänge
- Durchschnittliche Anzahl der Wörter pro Satz
- Median für die Satzlänge in Worten

Das Resultat der Auswertung soll in in eine Textdatei mit dem Namen `analysis_output.txt` geschrieben werden, die lokal gespeichert wird. Die Resultate sollten separat (in der gleichen Reihenfolge wie oben aufgezählt) auf eine neuen Zeile geschrieben werden. Dein Programm muss dabei mit verschiedenen Encodings der Inputdateien umgehen können (gegeben als parameter `encoding`). Es sollen mindestens die folgenden Zeichenkodierungen verarbeitet werden können: UTF-8, ISO-8859-1. Als Anschauungsbeispiel für das Format, in dem du die Eingabedateien erwarten kannst, sowie für Testzwecke sind der Aufgabenstellung einige zu verarbeitende Artikel beigefügt.

Abzugeben: Eine Python Datei.

3 CSV Modul

Du erhältst ein CSV-Datei von Sacramento Police Department welches 7'584 Verbrechen im Januar 2006 aufzeichnet. Deine Aufgabe besteht darin, die Datei zu verkürzen, bzw. die Informationen herauszufiltern, welche für dich wichtig sind und in eine neue CSV-Datei namens `modified_crime_data.csv` zu schreiben. Die neue CSV-Datei soll folgende Informationen enthalten:

- Das Datum und die Uhrzeit (`cdatetime`)
- Die Adresse (`address`)
- Den Stadtteil (`district`)
- Die Beschreibung (`crimedescr`)

Ebenfalls sollten nur Verbrechen aufgelistet werden, die zwischen 2:00 und 13:00 Uhr stattfanden und bei denen es um einen Diebstahl ging (Tipp: Stichwort z.B. 'Burglary'). Schreibe hierzu ein Modul mit der Funktion `modify_csv_file(filename)` wobei der Parameter `filename` die einzulesende CSV-Datei beschreibt. Die Spalten- und Zeilenreihenfolge soll in der neuen Datei gleich bleiben. Verwende für diese Aufgabe die Funktionen/Klassen/Methoden des CSV Moduls. Die Dokumentation dazu findest du [hier](#).

Abzugeben: Eine Python Datei.

Reflexion/Feedback

- Fasse deine Erkenntnisse und Lernfortschritte in zwei Sätzen zusammen.
- Wie viel Zeit hast du in diese Übungen investiert?