

Coding Style

Allgemeine zu beachten für guten Code

- leicht zu lesen und zu verstehen, auch von anderen
- gut zu testen
- keine Redundanzen
- so wenig abhängig von anderem Code wie möglich
- keine Überraschungen
- eine Klasse nur eine Verantwortlichkeit
- Klasse offen für Erweiterungen, aber geschlossen gegenüber Veränderungen/Modifikationen
- abgeleitete Klassen immer anstelle ihrer Basisklasse einsetzbar: abgeleitete Klasse erweitert Basisklasse ohne Einschränken oder Veränderungen

Variablenbenennungen

- englische Bezeichnungen
- Klassen: Substantive (großer Anfangsbuchstabe)
- Funktionen: Verben (klein geschrieben)
- Variablen: (klein geschrieben)
- eindeutige Beschreibung der Variable, Klasse, Funktion durch Namen
- nie mals l (kleines L), O (großes o), I (großes i) als Variablennamen

Python Code Layout

- 4 Leerzeichen zum Einrücken
- Imports: immer am Anfang der Datei (nach Modul Kommentaren, docstrings,...), jeweils eigenen Zeile für neue Bibliothek
- Argumente für Funktionen: bei Zeilenumbrüchen mit obigen Argumenten alignen

- Zeilenumbruch vor binären Operatoren (z.B. +, -, *, /), Zeilen alignen
- Leerzeilen: zur logischen Unterteilung des Codes, je 2 vor und nach Klassendefinitionen
- vermeiden von unnötigen Leerzeichen in Ausdrücken
- keine Leerzeichen in Ausdrücken vor Kommata, Semikolons, Doppelpunkten, offenen Klammern für Argumentenlisten, Slicing Parametern
- je ein Leerzeichen vor und nach Zuweisungen (=, +=, ...), Vergleichen (==, !=, ...), Booleans (and, or, not)

Kommentare

- auf englisch
- nur aktuelle Kommentare behalten
- vollständige Sätze
- Block-Kommentare für den gesamten Code: Einrückungenwie Code, # vor jeder Zeile (auch Leerzeilen)
- Inline-Kommentare: mindestens 2 Leerzeichen Abstand zum Code, Starten mit # und einem Leerzeichen (z.B. # Variable for counting)
- Docstrings für alle Module, Funktionen, Klassen und Methoden (z.B. """Return an integer""")
- bei mehrzeiligem Docstring: schließendes """ auf neuer Zeile