

Coding Guidelines

Limit Zeilenlänge

- max. 80 Zeichen

Kommentare / Dokumentation

- Alle Kommentare auf Englisch!
- Kommentare, die gesamte Methoden/Klasse beschreiben: in ganzen Sätzen (Fließtext)
- Kommentar-Format nach IDE-Vorgabe/für automatische Doc-Erstellung (z.B. JavaDoc)
- unnötige Kommentare weglassen
 - v.a. Inline-Kommentare nur sehr sparsam verwenden
 - keine Kommentare für eine einzelne Code-Zeile
- Kommentare immer aktuell halten! (bei relevanten Änderungen am Code)

Einrückungen und Zeilenumbrüche

- Leerzeichen statt Tabs (nach Sprachempfehlung: z.B. 2 bei Ruby, JavaScript, 4 bei Python)
 - --> in IDE einstellen
- Platzierung d. geschw. Klammern **{ }**:
 - Öffnende Klammer hinter Methodendeklaration
 - schließende Klammer mit Einrückung auf der gleichen Einrückungsebene
- optional: lange Zeilen teilen: Zeilenumbrüche für Übersichtlichkeit. Dabei beachten:
 - die neue Zeile weiter einrücken als die darüber, z.B. bis auf Ebene der öffnenden Klammer (wenn vorhanden, sonst eine Einrückungsebene)
 - bei binären Operatoren: Operator an Anfang der neuen Zeile (wie in Mathematik, z.B. schriftlich Addieren)
- Imports: nur einen pro Zeile
- → evtl. mit Hilfe von „Beautifier“/Auto-Format-Funktion in IDE

Leerzeilen

- Klassen- und Methoden-Definitionen mit zwei Leerzeilen umgeben
- logisch zusammengehörige Anweisungen als Block mit Leerzeile von nächstem Block trennen → aber: sparsam verwenden, keine größere Zahl von Einzeilern voneinander trennen

Leerzeichen

- vor und nach binären Operatoren
 - Ausnahme ggf. um unterschiedliche Priorität d. Operatoren deutlich zu machen:
 - `x = x*2 - 1`
- Keine Leerzeichen nach `(` und `[` und vor `)` und `]`
- Keine Leerzeichen vor `,` oder `;`
- Leerzeichen nach `,`
- Keine Leerzeichen am Zeilenende

konsistentes Benennungsschema

- → Überall beachten:
 - Variablen
 - Methoden
 - Klassen
 - -attribute
- deskriptive Namen verwenden!!
- immer Englisch
- Hilfsvariablen möglichst immer gleich benennen (i, j, n, sum, ...)
 - weitere Vorschläge?
 - → Vorsicht reserv. Schlüsselworte d. Sprache
- Boolean: „Namen [...] sollten einen Zustand ausdrücken, der wahr oder falsch sein kann“
- Groß-/Kleinschreibung (z.B. CamelCase) vs. Unterstriche?
 - Unser Vorschlag: `mixedCase`
- Evtl. GROSSSCHREIBUNG für Konstanten, SQL, o.a.
- Konstruktoren von Klassen: Benennung übergebenes Argument vs. aktuelles Objektattribut?
 - `length` VS. `_length` #evtl. (auch) `this.length` falls in Sprache möglich

Datei: Textcodierung

- UTF-8

Best Practice

- keine Globalen Variablen verwenden

- DRY („Don't Repeat Yourself“) / DIE („Duplication is Evil“)
 - (fast) gleicher Code an mehreren Stellen sollte vermieden werden
- Vermeiden von zu tiefer Verschachtelung
- → weitere Regeln je nach Sprache (z.B. C++ Umgang mit Referenzen u. Zeigern, ...)

Quellen:

<https://github.com/github/rubocop-github/blob/master/STYLEGUIDE.md>

<https://www.python.org/dev/peps/pep-0008/#string-quotes>

https://www.w3schools.com/js/js_conventions.asp

https://www.w3schools.com/js/js_best_practices.asp