

# 1 Python coding style

Es macht Sinn sich auf ein gemeinsames Code-Layout zu einigen, da dies die Lesbarkeit eines Programms ungemein erhöhen kann und sich nicht jeder aufs Neue Gedanken über die 'richtige' Formatierung machen muss.

Für Python gibt es die sogenannten Python Enhancement Proposals. Diese geben Richtlinien für das Formatieren, die Benennung von Variablen, Klassen, Funktionen, etc. an.

Für alle, die nicht die kompletten Proposals lesen wollen, hier in möglichst kompakter Form die (mir) wichtigsten Formatierungen für Python-Code.

## 1.1 Shebang

Es empfiehlt sich am Anfang eines Python Skriptes anzugeben, mit welchem Interpreter das Skript ausgeführt werden soll (macht auch das Korrigieren leichter, da dann klar ist, ob ihr Python 2.X oder 3.X beim Lösen der Aufgaben verwendet habt). Zudem ist es auch immer sinnvoll, die Enkodierung anzugeben (wir bitten euch stets die utf8 Kodierung zu verwenden, da diese die meisten Satzzeichen beinhaltet).

```
#!/usr/bin/env python2
# -*- coding: utf8 -*-
```

## 1.2 Dokumentierung

Bitte versucht alle Funktionen, die ihr schreibt, gleich am Anfang mit einem sogenannten 'docstring' zu versehen. Das hilft einerseits euch selbst, da ihr in Gruppen sicherlich auch den Code von anderen lesen müsst und andererseits uns beim Korrigieren der Abgaben. Zudem ist es beim Entwickeln von größeren Programmen unabdingbar, eine ausführliche Dokumentation direkt im Code zu haben, also was bedeuten die Variablen, was ist der Typ der Variablen, welchen Zweck hat eine Klasse/eine Methode,...

```
def function_with_types_in_docstring(param1, param2):
    """Example function with types documented in the docstring.

    'PEP 484'- type annotations are supported. If attribute,
    parameter, and
    return types are annotated according to 'PEP 484', they do not
    need to be
    included in the docstring:

    Parameters
    -----
    param1 : int
        The first parameter.
    param2 : str
        The second parameter.

    Returns
    -----
    bool
        True if successful, False otherwise.

    .. _PEP 484:
        https://www.python.org/dev/peps/pep-0484/
    """
```

” ”

### 1.3 Namen

Auch für die Namen verschiedener Python Objekte gibt es eine Konvention. Die wichtigsten Regeln:

- Funktionen sollen mit kleinen Buchstaben benannt werden. Wenn mehrere Wörter im Namen enthalten sind, sollen diese via `_` getrennt werden
- Konstanten sollen mit großen Buchstaben benannt werden. Wenn mehrere Wörter im Namen enthalten sind, sollen diese via `_` getrennt werden
- Klassennamen werden nach dem 'CamelCase' Muster benannt, das heißt, dass der Anfangsbuchstabe eines Wortes groß geschrieben wird und nicht mit einem `_` getrennt werden.