



[SAP Programming Styleguide](#)

SAP hat gerade die neuen Programmierkonventionen vorgestellt, die sich deutlich von vergangenen Empfehlungen unterscheiden und zudem öffentlich auf github.com gepflegt werden.

Clean Code, Unit Tests und mehr

Unter der Internetadresse <https://github.com/SAP/styleguides> kannst du dich über die aktuellen Empfehlungen informieren. Sie fallen sehr umfangreich aus und sind in Englisch verfasst. Die [Empfehlungen](#) geben dir Hinweise zur Benennung und Verwendung von Objekten. In der Regel gibt es einen *Good Case* und ein *Anti-Pattern*.

Schreibweise

Die Empfehlungen sagen deutlich aus, wie in welchen Fällen die Schreibweise sein sollte. Zum Beispiel wird darauf hingewiesen, dass Methoden, die einen RECEIVING-Parameter haben, dieser als direkte Zuweisung verwendet werden sollte:

```
DATA(sum) = aggregate_values( values ).
```

Das Anti-pattern hierzu wäre:

```
aggregate_values(  
    EXPORTING  
        values = values  
    RECEIVING  
        result = DATA(sum) ).
```

Einsatz von Sprachelementen

Es wird sehr detailliert darüber geschrieben, in welchen Fällen Sprachelemente sinnvoll eingesetzt werden sollen. Ein gutes Beispiel hierfür ist die [Inline-DATA-Definition](#). Es werden auch Empfehlungen zu [LOOP](#) und anderen Sprachelementen gegeben. Hierbei wird nicht nur darauf eingegangen, wie LOOP verwendet werden sollte, sondern auch, welche Tabellentypen man benutzen sollte.

Bereiche

Der Styleguide gliedert sich in viele Bereiche: [Formatting](#), [Error Handling](#) und mehr. Es wird dabei sehr auf Kleinigkeiten eingegangen, die – selbst wenn man sie selbst nicht so wichtig findet – doch einmal nachdenken sollte. Zum Beispiel wird empfohlen, LOOP INTO REF zu verwenden und nicht ASSIGNING. Einfach aus dem Grund, um klarzustellen, dass ASSIGNING nicht benötigt wird. Wenn Felder geändert werden sollen und deswegen ASSIGNING benutzt wird, ist das okay, aber es soll eben sicherstellen, dass nicht einfach aus dem Grund Dasmacheichimmerso verwendet wird.

Testing

Ein großer Bereich ist [Testing](#). Auf den letzten Veranstaltungen hat sich immer mehr herauskristallisiert, dass das automatische Testen immer wichtiger wird und immer mehr ins Bewusstsein des gemeinen ABAP-Programmierers rückt. Wer sich also noch nicht damit beschäftigt hat, sollte dies dringend tun. Ich empfehle den kostenlosen, sehr umfangreichen OpenSAP-Kurs [Writing Testable Code For ABAP](#).

Fazit

Der SAP Style Guide ist ungewohnt klar und kompromisslos. Wo in der Vergangenheit häufig ein „Kannst du so machen, aber auch anders“ zu lesen war, wird hier klar definiert, wie es sein sollte. Das finde ich gut, auch wenn ich einiges anders sehe.

Sehr gut finde ich auch, dass an Beispielen auch auf Kleinigkeiten eingegangen wird. Ob man diese Kleinigkeiten dann berücksichtigt oder nicht, bleibt einem natürlich weiterhin selbst überlassen. Die SAP zeigt hier jedoch, dass die Regeln einen guten Grund haben.

Es wird sehr viel Wert auf *sauberen Code* gelegt. An mehreren Stellen wird darauf hingewiesen, dass man Variablen, Methoden etc. so benennen sollte, dass sie eindeutig sind und klar erkennbar machen, wofür sie benutzt werden.

Einem immer wiederkehrenden Streitthema, der Ungarischen Notation, wird eine ganz klare Absage erteilt. In einer Welt, in der man bei Mouseover sofort den Typ einer Variablen oder eines Objektes erkennen kann, ist es nicht mehr notwendig, einen Hinweis auf die Typdeklaration in den Namen aufzunehmen.

Spicker und Goldene Regeln

Die Guidelines sind in einem [Cheat Sheet](#) auf vier Seiten zusammengefasst. Ebenso gibt es eine sehr komprimierte Version als [Golden Rules](#).