

# LU01b - Linting & Formatting

## Formatting

Formatter sorgen dafür, dass der gemeinsame Code einheitlich formatiert wird. Oftmals besitzen IDE bereits eine Standard-Formatierung oder die Möglichkeit, Formatierungsregeln mit Entwicklern mit denselber IDE auszutauschen.

Beispiele von IDE-spezifischen Formatierungen:

- **IntelliJ:** Code Style XML unter `.idea/codeStyles/...`
- **Visual Studio Code:** Settings unter `.vscode/settings.json`
- **Eclipse: Code Style Formatter:** Eclipse Code Formatter XML
- **Diverse IDE:** `.editorconfig` (aber nicht alle Regeln funktionieren aber bei allen IDE's)

IntelliJ Project.xml	VS Code settings.json	Eclipse java-formatter.xml	.editorconfig
<pre> ... &lt;/JavaCodeStyleSettings&gt; &lt;codeStyleSettings language="JAVA"&gt;   &lt;!-- Tab size --&gt;   &lt;indentOptions&gt;     &lt;option name="INDENT_SIZE" value="4"/&gt;     &lt;option name="TAB_SIZE" value="4"/&gt;     &lt;option name="USE_TAB_CHARACTER" value="false"/&gt;   &lt;/indentOptions&gt;   &lt;!-- New line at end of file --&gt;   &lt;option name="INSERT_FINAL_NEWLINE" value="true"/&gt; &lt;/codeStyleSettings&gt; ... </pre>	<pre> {   // Tab size   "editor.tabSize": 4,   "editor.insertSpaces": true,   // New line at end of file   "files.insertFinalNewline": true,   // Java-specific   "editor.formatOnSave": true,   // Remove unused imports "editor.codeActionsOnSave": {   "source.organizeImports": true }, } </pre>	<pre> ... &lt;profile kind="CodeFormatterProfile" name="Project Formatter" version="12"&gt;   &lt;!-- Tab size --&gt;   &lt;setting id="org.eclipse.jdt.core.formatter.tabulation.size" value="4"/&gt;   &lt;setting id="org.eclipse.jdt.core.formatter.indentation.size" value="4"/&gt;   &lt;setting id="org.eclipse.jdt.core.formatter.tabulation.char" value="space"/&gt;   &lt;!-- New line at end of file --&gt;   &lt;setting id="org.eclipse.jdt.core.formatter.insert_new_line_at_end_of_file_if_missing" value="true"/&gt; &lt;/profile&gt; ... </pre>	<pre> ... [*] indent_style = space indent_size = 4 tab_width = 4  end_of_line = lf insert_final_newline = true trim_trailing_whitespace = true ... </pre>

Beispiele von Formatter-Bibliotheken

- Javascript, Typescript, HTML, CSS, JSON → Prettier
- Python → black, pep8, ...
- Java → Google Java Format

## Linting

Linting (dt. „fusseln“) bezeichnet das automatische finden und teilweise sogar korrigieren von möglichen Fehlern o.Ä. anhand von einem definierten Regelset. Entsprechende Bibliotheken sind für diverse Sprachen erhältlich.

Gängige Linter-Bibliotheken

- Javascript, Typescript → ESLint(<https://eslint.org/>) , TSLint wurde eingestellt und ist daher nicht mehr empfohlen
- Python → Pylint (<https://www.pylint.org/>)
- Java → PMD (<https://pmd.github.io/>)
- Diverse Sprachen → SonarQube for IDE (SonarLint)

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/cicd/learningunits/lu01/b?rev=1768917745>

Last update: **2026/01/20 15:02**

