

Übungsblatt:	13	1. Abgabepartner*in:	Ada Lovelace
Aufgabe:	37	2. Abgabepartner*in:	Donald Knuth
Abgabegruppe:	42	3. Abgabepartner*in:	Alan Turing

Dies ist eine \LaTeX -Vorlage für Übungszettelabgaben im Fach Informatik, die das `listingsutf8`-Paket nutzt. Dieses ist in der Regel sofort nutzbar und erfordert keine weitere Einrichtung. Das Syntax Highlighting ist hier jedoch nicht ganz so detailliert wie beim `minted`-Paket, für das in meinem Git-Repository¹ ebenfalls eine Vorlage bereitgestellt wird. Nachfolgend gibt es ein paar Hinweise, wie diese Vorlage zu verwenden ist.

Disclaimer

Ich kann leider grundsätzlich keinen \LaTeX -Support anbieten und verweise daher auf gängige Suchmaschinen und die \TeX -Community von StackExchange². Sucht man den LaTeX-Befehl für ein bestimmtes Symbol, ist Detexify³ praktisch.

Verwendung

Als Test sollte man versuchen, die Datei `abgaben-listings.tex` in einem TeX-Editor zu öffnen und zu kompilieren. Dabei handelt es sich um den TeX-Code zu genau diesem Dokument. Wenn das nicht klappt, kann das folgende Gründe haben:

- In den Umgebungsvariablen des Systems fehlt das Verzeichnis der \LaTeX -Distribution.
- Der Compiler wird nicht mit der Shell-Escape-Option aufgerufen. Ggfs. sollte in den Einstellungen des Editors nachgeschaut werden, dass `pdflatex` mit der Option `-shell-escape` aufgerufen wird.
- Irgendwas anderes. Man prüfe die Ausgabe des Compilers.

Wenn alles geklappt hat, kann man diese Vorlage verwenden. Ich habe als Einstiegshilfe unten verschiedene Beispiele – unter anderem zum Einbinden von Quellcode – angehängt. Die Verwendung der entsprechenden TeX-Befehle sollte anhand der Datei `abgaben-listings.tex` erschießbar sein. Weitere Hinweise und Konfigurationsmöglichkeiten findet man in der Paketdokumentation⁴.

Nicht vergessen: Ein automatisches Syntax Highlighting entbindet **niemals** von der Pflicht, gut lesbaren und gut nachvollziehbaren Code zu produzieren.

Phil Steinhorst

<https://gitlab.com/phist91/latex-templates>

¹<https://gitlab.com/phist91/latex-templates>

²<https://tex.stackexchange.org/>

³<http://detexify.kirelabs.org/>

⁴<http://texdoc.net/texmf-dist/doc/latex/listings/listings.pdf>.

Beispiele

Textformatierungen

Eine Auflistung:

- Dieser Text ist **fett**.
- Dieser Text ist *kursiv*.
- Dieser Text ist unterstrichen.
- Dieser Text ist **rot**.
- Dieser Text ist anklickbar.

Eine Aufzählung:

- (a) Eine Liste,
- (b) aber mit
- (c) Nummerierung!

Eine zentrierte Tabelle:

Etwas	gewöhnungsbedürftig,
	aber vielseitig und machbar!

Die gleiche Tabelle, aber mit Beschreibung:

Etwas	gewöhnungsbedürftig,
	aber vielseitig und machbar!

Tabelle 1: Viele Editoren bieten Assistenten zum Anlegen von Tabellen wie dieser.

Grafiken

Eine eingebundene Grafikdatei:



Die gleiche Grafik, in verschiedenen Größen:

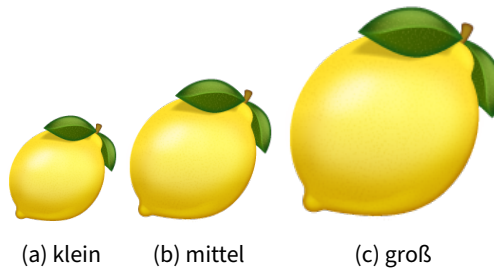


Abbildung 1: Zitronen in verschiedenen Größen.

Eine TikZ-Grafik mit Beschreibung:

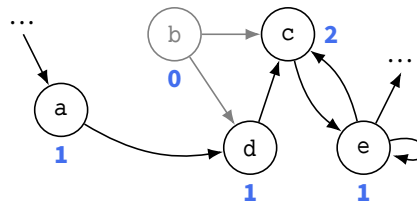


Abbildung 2: TikZ ist ein sehr mächtiges Paket zur Erstellung ansprechender Vektorgrafiken.

Mathematische Formeln

Die Formel $a^2 + b^2 = c^2$ ist kompakt und passt in den Fließtext.

Die Formel $x := \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}$ sprengt hingegen die Zeilenhöhe, was innerhalb eines größeren Absatzes sehr unschön aussehen kann, wie man hier sieht.

Besser: Einzeilige Formeln absetzen und zentrieren:

$$x := \frac{1}{1 + \frac{1}{1 + \frac{1}{2}}}$$

Und mehrzeilige Formeln? Bitte nicht so:

$$f(x) = x^2 + 2x - 7$$

$$f'(x) = 2x + 2$$

$$f''(x) = 2$$

Viel besser: Abgesetzt und ausgerichtet:

$$f(x) = x^2 + 2x - 7$$

$$f'(x) = 2x + 2$$

$$f''(x) = 2$$

Formeln können auch nummeriert werden:

$$\sum_{i=0}^n x^i = \frac{1 - x^{n+1}}{1 - x} \quad (1)$$

$$\sum_{i=0}^n \frac{1}{2^i} = 2 - \frac{1}{2^n} \quad (2)$$

...um sie später zu referenzieren: Formel (2) ist ein Spezialfall von Formel (1).

Eine Funktion:

$$\begin{aligned} f: \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto x^2 \end{aligned}$$

Geht auch mit Fallunterscheidung:

$$\begin{aligned} \text{sgn}: \mathbb{R} &\longrightarrow \mathbb{R} \\ x &\longmapsto \begin{cases} 1, & \text{falls } x > 0 \\ 0, & \text{falls } x = 0 \\ -1, & \text{falls } x < 0 \end{cases} \end{aligned}$$

Eine Matrix:

$$A := \begin{pmatrix} 1 & a & a^2 & a^3 \\ 1 & b & b^2 & b^3 \\ 1 & c & c^2 & c^3 \\ 1 & d & d^2 & d^3 \end{pmatrix}$$

Wichtig! Niemals Fließtext ohne Befehle wie `\text{...}` in Formel-Umgebungen setzen:

$$x\text{Sta}ff\text{el}$$

Das führt zu Problemen beim Zeichenabstand und sieht komisch aus.

Beispiel für direkt eingegebenen Java-Code:

```
1 import java.util.*;
2 public class Hallo{
3     public static void main( String[] args ) {
4         System.out.print("Hallo Welt!");
5     }
6 }
```

Beispiel für direkt eingegebenen C-Code:

```
1 #include <stdio.h>
2
3 int main(int argc, char** argv){
4     int i;
5     for(i = 0; i < argc; i++){
6         printf("%s \n", argv[i]);
7     }
8     return 0;
9 }
```

Beispiel für Java-Code, der aus einer eigenen Datei eingebunden wird:

```
1 package pst.gcsim.GarbageCollectors;
2 import java.util.*;
3
4 /**
5  * Klasse zur Realisierung eines Mark-Sweep-Kollektors.
6  */
7 public class MarkSweep<T extends Allocator> extends GarbageCollector<T> {
8     public static final int MS_WHITE = 0;
9     LinkedList<HeapObject> toScan;
10
11     public MarkSweep(CollectionController<T> controller) {
12         this.toScan = new LinkedList<>();
13         this.controller = controller;
14     }
15
16     /**
17      * Ausführung der Bereinigungsphase (ohne Animation).
18      */
19     void sweep() {
20         controller.getObjects().sort(new AddressComparator());
21         // Verhinderung von ConcurrentModificationExceptions
22         ArrayList<HeapObject> toRemove = new ArrayList<>();
23         for (HeapObject obj : controller.getObjects())
24             if (obj.getMark() == MS_WHITE) toRemove.add(obj);
25             else obj.setMark(MS_WHITE);
26         toRemove.forEach(obj -> {
27             controller.getAllocator().free(obj);
28             controller.getObjects().remove(obj);
29         });
30     }
31 }
```

Beispiel für C-Code, der aus einer eigenen Datei eingebunden wird:

```
1 #include <stdio.h>
2 int main(int argc, char** argv){
3     int i;
4     for(i = 0; i < argc; i++){
5         printf("%s \n", argv[i]);
6     }
7     return 0;
8 }
```

Beispiel für Inline-Java-Code: `this.toScan = new LinkedList<>();`

Beispiel für Inline-C-Code: `int main(int argc, char** argv)`