R. Herzog, G. Müller
Universität Heidelberg

Worksheet 1 (Solution)

Date issued: 19th October 2026

Problem 1.1. (Hello World using local TeX engines)

(a) Install a TeX LIVE distribution on your operating system no older than TeX LIVE 2025[1].

(b) Create a folder for this course and a subfolder for the first week. In the week's folder, create a file *hello.tex* and *type* (actually type, no copy-pasting) the content of the "Hello, World!" example of the lecture.

(c) Use a minimum of two different engines to typeset the corresponding *hello.pdf*. Which line number of *hello.log* tells you the engine that produced the logfile?

Solution.

The TeX engine writes its version into the first line of the log file. Otherwise I can not provide an example solution for setting up a working local TeX environment.

Problem 1.2. (Overleaf and TeX LIVE versions)

(a) Sign up and log in to Overleaf.

(b) Create an "empty" project for this exercise and share it with your team member.

(c) Add the line

> Trüge der müde Müller prüde Dübel, so säße er häufiger auf bärtigen Äpfeln.

to the document body of *main.tex* and typeset the document using *pdflatex* of TeX LIVE 2025 or 2026.

[1] If you are worried about breaking existing configurations, that you may have already set up, then you may set up a docker container for this class' exercises instead.

(d) Select the oldest underlying TeX LIVE version available in Overleaf (which is it?) and confirm that this does not typeset correctly but no error or warning appears. Do the log files tell you anything useful?

(e) Set the underlying TeX LIVE version to the oldest one that compiles the sentence cleanly. Which version is it and why?

Solution.

(a) No example solution for setting up Overleaf.

(b) No example solution for creating projects in Overleaf.

(c) The document builds cleanly and prints exactly the input line as viewed in the editor to the pdf. No errors or warnings show in Overleaf.

(d) At the time of writing this (16th October 2026) the oldest available Version on my Overleaf is 2014 (marked "Legacy"). When selecting this version and building the document, the rendered line in the pdf reads as:

> Trge der mde Mller prde Dbel, so se er hufiger auf brtigen pfeln.

I. e., all 10 non-ASCII characters are not printed. There are no errors or warnings shown. In the log file, I see 20 instances of

> Missing character: There is no ^^c3 in font cmr10!

(the byte shown in ^^xx notation varies from line to line, one per leftover byte)

What is happening here is that the file *main.tex* is stored as bytes on the harddrive. The non-ASCII characters have to be split up into multiple (two) bytes. Recognizing multiple bytes in a series as a single character requires an (UTF-8) interpreter. The editor has that interpreter, thus rendering the .tex-file as you would it expect it. For your engine, whether that interpreter is loaded by default (enabling the engine to typeset these characters correctly), depends on the engine itself and the release date of the engine. The first version of pdfLaTeX that has default UTF-8 interpretation is 2018.

When the UTF-8 interpreter is unavailable for the engine, it reads single bytes without triggering the interpreter at the lead byte, so each non-ASCII character is read as two single bytes. For each byte, the engine typesets the glyph the loaded font associates with that byte. The font used in this example (cmr10) has no glyphs for those bytes, so the characters are simply *dropped* (hence the "Missing character" messages and the empty

gaps in the output). Other fonts may instead have glyphs at those byte positions and then produce incorrect, garbled characters.

(e) That would be 2018, see the explanation above.