

Test Markdown

Wuppie Fluppie

Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

GH-Preview

Hello World

Hier ist normaler Markdown-Text, mit **fett** und auch *kursiv*.

- Stichpunkt 1
 - Stichpunkt 2
 - Stichpunkt 3
-
1. Aufzählung 1
 2. Aufzählung 2
 3. Aufzählung 3
 - 3.1 Unterpunkt 3.1
 - 3.2 Unterpunkt 3.2

Hier die **Pandoc-Markdown** mark-Erweiterung.

Inline

$$\mathbf{g} = (g_1, \dots, g_m) \in \{0, 1\}^m$$

- $a^i b^{2^*i}$ ist nicht regulär
- $a^i b^{2^*i}$ für $0 \leq i \leq 3$ ist regulär

Block

$$\Phi(\mathbf{g}_i) = F(\Gamma(\mathbf{g}_i)) - w \cdot \sum_j (Z_j(\Gamma(\mathbf{g}_i)))^2$$

$$p_{sel}(\mathbf{g}_k) = \frac{\Phi(\mathbf{g}_k)}{\sum_j \Phi(\mathbf{g}_j)}$$

$$g_i^{(t+1)} = \begin{cases} \neg g_i^{(t)} & \text{falls } \chi_i \leq p_{mut} \\ g_i^{(t)} & \text{sonst} \end{cases}$$

Known Problems

- VSCode Preview: `\mbox{ tanh }` => tanh => `\text{ tanh }` => tanh
- GH Preview:
 - `` => => ?? => ??
 - `\operatorname{tanh}` => tanh => `\mathop{\text{tanh}}` => tanh

$$g_i^{(t+1)} = \begin{cases} \neg g_i^{(t)} & \text{falls } \chi_i \leq p_{mut} \\ g_i^{(t)} & \text{sonst} \end{cases}$$

Schwierig: In Pandoc-Markdown muss Mathe mit `$` oder `$$` eingeschlossen werden, unabhängig vom konkreten Inhalt. In LaTeX ist aber `\begin{eqnarray}` bereits der Beginn einer Mathe-Umgebung, d.h. hier wären extra `$$ falsch`. Das muss per Filter korrigiert werden!

```
$$\begin{eqnarray}
S & \rightarrow & a \ A & \nonumber \\
A & \rightarrow & d \ B \mid \ b \ A \mid \ c \ A & \nonumber \\
B & \rightarrow & a \ c \mid \ b \ C \mid \ c \ A & \nonumber \\
C & \rightarrow & \epsilon & \nonumber \\
\end{eqnarray}$$
```

Links

Link to WWW

craftinginterpreters.com/the-lox-language.html

Internal Links

selbe ebene: [readme.md](#)

unterordner: [subfolder/foo.md](#)

zurück nach oben I: [../02-parsing/antlr-parsing.md](#)

zurück nach oben II: [../../homework/sheet01.md](#)

Code

```
grammar Hello;

start : stmt* ;

stmt  : ID '=' expr ';' | expr ';' ;
expr  : term ('+' term)* ;
term  : atom ('*' atom)* ;
atom  : ID | NUM ;

ID     : [a-z][a-zA-Z]* ;
NUM    : [0-9]+ ;
WS     : [ \t\n]+ -> skip ;
```

Java-Code kompilieren: `javac *.java`

```
import org.antlr.v4.runtime.CharStreams;
import org.antlr.v4.runtime.CommonTokenStream;
import org.antlr.v4.runtime.tree.ParseTree;

public class Main {
    public static void main(String[] args) throws Exception {
        HelloLexer lexer = new HelloLexer(CharStreams.fromStream(System.in));
        CommonTokenStream tokens = new CommonTokenStream(lexer);
        HelloParser parser = new HelloParser(tokens);

        ParseTree tree = parser.start(); // Start-Regel
        System.out.println(tree.toStringTree(parser));
    }
}
```


Code ohne alles

```
import org.antlr.v4.runtime.CharStreams;
import org.antlr.v4.runtime.CommonTokenStream;
import org.antlr.v4.runtime.tree.ParseTree;

public class Main {
    public static void main(String[] args) throws Exception {
        HelloLexer lexer = new HelloLexer(CharStreams.fromStream(System.in));
        CommonTokenStream tokens = new CommonTokenStream(lexer);
        HelloParser parser = new HelloParser(tokens);

        ParseTree tree = parser.start(); // Start-Regel
        System.out.println(tree.toStringTree(parser));
    }
}
```

Images

Figures (w/ caption) should be centered like in LaTeX. Inline images will appear as is (also like in LaTeX).

Images with Caption

kleines Bild, keine Breiteangabe:



kleines Bild, mit Titel und Breite:



Abbildung 2: "B", width="5%"

breites Bild, keine Breiteangabe:

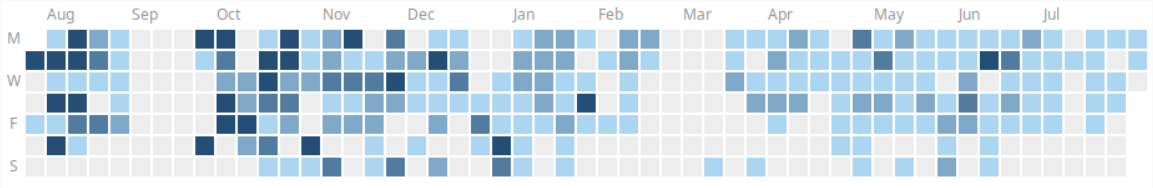


Abbildung 3: "wuppie" (wide)

breites Bild, mit Breiteangabe:

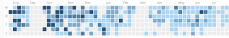


Abbildung 4: "wuppie", width="20%"

breites Bild über HTTP, keine Breiteangabe:

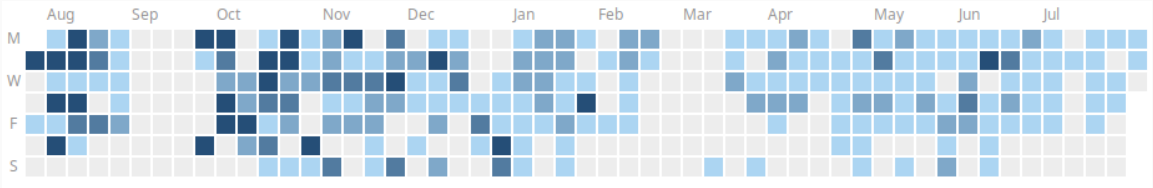


Abbildung 5: “wuppie” via web (raw)

breites Bild über HTTP mit `credits`-Span, keine Breiteangabe:

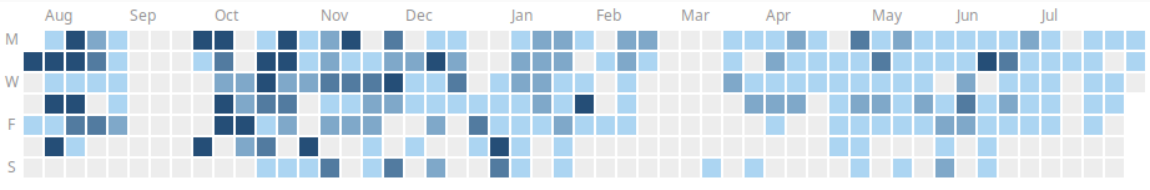


Abbildung 6: “wuppie” via web (raw) (Quelle: “FooFOOOO” by me on void.intern.com)

Quelle: “Foo” by me on void.extern.com

Images w/o Caption

ohne alles:



mit breitenangabe:

B

mit breitenangabe (“width”) und titel:

B

mit breitenangabe ("web_width") und titel:

B

mit breitenangabe (“width” und “web_width”) und titel:

B

Images w/ `img`-Code

using “5%”:

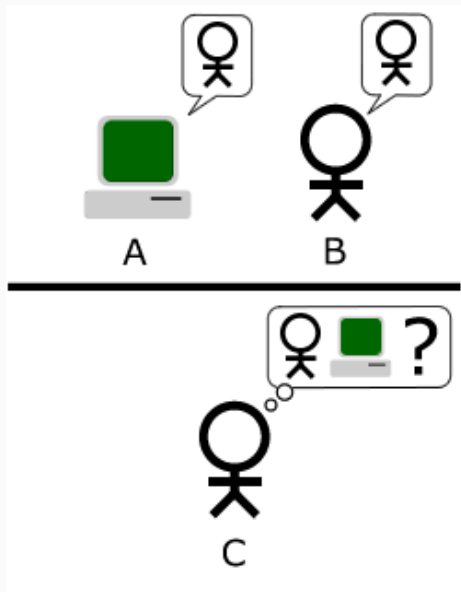
using “80px”:

mit div drumherum:

B

Images from Web (LaTeX backend)

When converting to LaTeX (PDF, Beamer), Pandoc will attempt to download the referred images. However, recently a lot of sites deny this. It seems we need to set an user-agent.



mit caption:

Tabelle 1: Tabelle als Markdown-Pipe-Table, vgl. (Abelson, Sussmann, und Sussmann 1996)

Rechtsbündig	Linksbündig	Default	Zentriert
foo	foo	foo	foo
123	123	123	123
bar	bar	bar	bar

ohne caption:

Rechtsbündig	Linksbündig	Default	Zentriert
foo	foo	foo	foo
123	123	123	123
bar	bar	bar	bar

Normales Zitieren (Siek 2023) ...

Mit Seitenangabe (Siek 2023, 111) oder Kapitel (Siek 2023, Kap. 111) ...

Als Author-Zitat Siek (2023) ...

Details

Zusammenfassung: NIX :)

Lalelu ...

Alert Extension

GH introduced “alerts” with distinctive styling, like

[!NOTE] Foo bar, wuppie fluppie!

[!TIP] Foo bar, wuppie fluppie!

[!IMPORTANT] Foo bar, wuppie fluppie!

[!WARNING] Foo bar, wuppie fluppie!

[!CAUTION] Foo bar, wuppie fluppie!

(see <https://github.blog/changelog/2023-12-14-new-markdown-extension-alerts-provide-distinctive-styling-for-significant-content/>)

Let's stick with Pandocs divs in Markdown content and use filters for export:

Foo bar, wuppie fluppel! Blablabla third line of nonsense ...

Foo bar, wuppie fluppel!

Foo bar, wuppie fluppel!

Foo bar, wuppie fluppel!

Foo bar, wuppie fluppel!

- Export to GH Markdown using “distinctive alerts”
- Export to Hugo using notice shortcode
- Export to Beamer using beamercolorbox (also beameruserguide.pdf; or `block`, `alertblock`, `examples` - cf. https://www.overleaf.com/learn/latex/Beamer%23Creating_a_table_of_contents)

This should probably be in line with #180 ...

THIS IS SLIDE-ONLY CONTENT, STUPID!

Footnotes

Sometimes¹ we need some² footnotes.

Test for Docsify

~~Docsify seems to recognize footnotes even in inline code:~~

Zeichenkette	Beschreibt
<code>^</code>	Zeilenanfang
<code>[abc]</code>	“a” oder “b” oder “c”
<code>[^abc]</code>	alles außer “a”, “b” oder “c” (Negation)
<code>^abc</code>	test
<code>[^abc]</code>	test
<code>[^abc]</code>	test
<code>[^ abc]</code>	test
<code>[a-zA-Z]</code>	alle Zeichen von “a” bis “z” und “A” bis “Z” (Range)
<code>[a-z&&[def]]</code>	“d”, “e” oder “f” (Schnitt)
<code>[a-z&&[^bc]]</code>	“a” bis “z”, außer “b” und “c”: <code>[ad-z]</code> (Subtraktion)
<code>[a-z&&[^m-p]]</code>	“a” bis “z”, außer “m” bis “p”: <code>[a-lq-z]</code> (Subtraktion)

Handling of TeX Shenanigans

Zustand: (Formale) Beschreibung eines Zustandes der Welt

Aktion: (Formale) Beschreibung einer durch Agenten ausführbaren Aktion

- Anwendbar auf bestimmte Zustände
- Überführt Welt in neuen Zustand (“Nachfolge-Zustand”)

LaTeX-Befehle wie `\bigskip` etc. sollten automatisch entfernt werden:

Hier nach den LaTeX-Befehlen.

Geeignete Abstraktionen wählen für Zustände und Aktionen!

Pandoc Shenanigans

Pandoc sometimes transforms whitespace into UTF8 whitespace, which results in pdftex yelling at me, when this appears in lstlistings.

Example:

```
cf. https://www.overleaf.com/learn/latex/Beamer%23Creating\_a\_table\_of\_contents
```

normal white space:

```
cf. https://www.overleaf.com/learn/latex/Beamer%23Creating\_a\_table\_of\_contents
```

pandoc replacement:

```
cf. https://www.overleaf.com/learn/latex/Beamer%23Creating\_a\_table\_of\_contents
```

Also ... hmmm. And “wuppie”?

Columns

blablabla

Zum Parsen von Ausdrücken (*Expressions*) könnte man diese einfache Grammatik einsetzen. Ein Ausdruck ist dabei entweder ein einfacher Integer oder eine Addition oder Multiplikation zweier Ausdrücke.

```
expr : e1=expr '*' e2=expr      # MUL
      | e1=expr '+' e2=expr      # ADD
      | INT                     # NUM
      ;
```

wuppie! fluppie! foo? bar ...

Beim Parsen von "5*4+3" würde dabei der folgende Parsetree entstehen:



Credits

Typische Regeln und Konventionen tauchen überall auf, beispielsweise bei Tim Pope (siehe nächstes Beispiel) oder bei “How to Write a Git Commit Message”.

Short (50 chars or less) summary of changes

More detailed explanatory text, if necessary. Wrap it to about 72 characters or so. In some contexts, the first line is treated as the subject of an email and the rest of the text as the body. The blank line separating the summary from the body is critical (unless you omit the body entirely); tools like rebase can get confused if you run the two together.

Further paragraphs come after blank lines.

- Bullet points are okay, too
- Typically a hyphen or asterisk is used for the bullet, preceded by a single space, with blank lines in between, but conventions vary here

Use `[bla]{.credits nolist=true}` to put a nicely formatted reference to the original sources in the text without adding it to the list of exceptions to our licence (just giving credits):

Quelle: Test 1: Eigenes Material basierend auf einer Idee nach XYZ.

Note: Using the attribute “nolist” with any value would prevent this span from being included in the exceptions list since values will be read as string in the filter. So even

`[bla]{.credits nolist=false}` will work:

Quelle: Test 2: Eigenes Material basierend auf einer Idee nach XYZ.

Quelle: Test 3: Eigenes Material basierend auf einer Idee nach XYZ.

Quelle: Test 4: Eigenes Material basierend auf einer Idee nach XYZ.

Quelle: Test 5: Eigenes Material basierend auf einer Idee nach XYZ.

Do not use the old `origin` span anymore - superceded by `credits`. This should emit a warning. . .

ShowMe

Hier ein ShowMe-Test:

this is hidden content ...

(but not used anymore)

Use `details` instead:

this is hidden content ...

wuppie

this is a show-me w/ title :)

CBOX

this is content to be centered (and put into a box)...

(but not used anymore)

Center

this content should be centered

Alert

This will be highlighted. (but not used anymore)

Use Pandoc's mark extension instead: This **will** be highlighted. Even **with bold** text.

Hinweis

This is a hint. (but not used anymore)

Thema

The topic of this task or ... (but not used anymore)

BSP

Lalalelu

Simple Beispiel-Button X (but not used anymore)

lalelu

use `ex` instead:

Simple Beispiel-Button X

Beispiel-Button w/ link

Vor \pause...

Vor \pause...

Nach \pause... (“neue” Slide)

should be added automatically and in `\scriptsize` or `<sup><sub>`



Unless otherwise noted, this work is licensed under CC BY-SA 4.0.

Exceptions:

- “FooFOOOO” by me on void.intern.com
- Turing Test version 3.png by Bilby on Wikimedia Commons (Public Domain)
- “A Note About Git Commit Messages” by Tim Pope on tbagery.com
- “Foo” by me on void.extern.com

Last modified: 81fea6d (test: use unbalanced single and double quotes mit backticks, 2025-09-02)