

Das ist der Haupt-Titel - M300

Untertitel

Sub-Untertitel

mehr Untertitel

Text Formatieren

Normal, dann **fett** und jetzt *kursiv* und wieder **fett und kursiv**, jetzt wieder normal

Achtung: MD brauch keine `cr/lf` für einen Zeilenumbruch.

Markdown Text in realtime sehen

mit der im Editor installierten Erweiterung **Instant Markdown** können wir die Änderungen am Text direkt im Browser unter dem [URL](#) mitverfolgen

mit einem `>` kann ein **Blockquote** erzeugt werden

Code highlighten

Code kann auf verschiedene Arten highlighted werden. Nachfolgend werden drei Arten beschrieben

- mit Hochkommas
- durch einrücken mit Blanks
- direkt im Text dargestellt

Code Darstellung mit Hochkomma (`)

Folgender Block wird als Code dargestellt. Diese Methode ist effizient, wenn grosse Codeblöcke dargestellt werden. Dazu wird jeweils am Anfang und Ende des Codeblocks je 3 Hochkommas gesetzt. Sie können das sehen, indem sie **README.md** als Text ansehen.

```
pi@kub02:~ $ kga all
NAMESPACE      NAME                                     READY   STATUS    5
RESTARTS      AGE
default        pod/blog-6cf85d9798-gd945             1/1     Running   5
218d
default        pod/jenkins-ui-69d649cf-r9474         1/1     Running   1
74d
default        pod/jenkins-ui-nfs-747979698f-mj9v8    1/1     Running   5
218d
```

Code Darstellung mit einrücken (4 Blanks)

Folgender Text wird als Code dargestellt. Diese Methode eignet sich wenn kurze Codeblöcke dargestellt werden. Dazu wird der Text einfach mit 4 Blanks eingerückt.

```
pi@kube02:~ $ kubectl get node
NAME      STATUS   ROLES    AGE   VERSION
kube02    Ready    master   358d   v1.13.1
kube03    Ready    <none>   358d   v1.13.1
kube04    Ready    <none>   358d   v1.13.1
```

Code direkt im Text darstellen

Oft will man in einem Text ein Kommando darstellen. Dazu eignet sich die Methode mit einfach Hochkommas direkt im Text.

Beispiel:

Gibt man auf einem Unix System `alias` ein, so werden die definierten Aliasbefehle aufgelistet

```
pi@kube02:~ $ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias kc='kubectl create --validate -f'
alias kg='kubectl get'
alias kga='kubectl get --all-namespaces'
alias kgall='kubectl get all --all-namespaces'
alias kgs='kubectl get -n=kube-system'
alias ks='kubectl -n kube-system'
alias ls='ls --color=auto'
```

Trennstrich

--- erzeugen einen Trennstrich

Bilder einfügen

```
![Image](images/denken.png)
```

Denken
IST WIE
GOOGELN
-nur krasser-

oder alternativ mit HTML Code.

```

```



Das hat den Vorteil, dass die Imagegrösse dem Bild angepasst werden kann.

```

```



leider funktioniert das Anpassen der Grösse mit HTML-Code nicht in allen MD Implementationen

Links definieren und wiederverwenden

Links global definieren

Der Link wird einmal definiert ...

```
[1]: http://www.tbz.ch "Die TBZ Homepage"  
[2]: https://sbb.ch "SBB Fahrplan"  
[3]: https://www.markdownguide.org/basic-syntax "Sehr gutes Markdown Tutorial"
```

... und kann danach im ganzen Dokument immer wieder referenziert werden. Dabei ist es egal, an welcher Stelle im Dokument die Definition steht.

wir probieren das hier gleich einmal aus

Die meisten Schüler der [TBZ](#) kommen mit der [Bahn](#) zur [Schule](#).

Viel Wissenswertes über Markdown kann in diesem [Tutorial](#) nachgelesen werden

Links einmalig definieren

Links können folgendermassen für die einmalige Verwendung definiert werden

```
Link zur [TBZ Webseite][1]  
  
[Link zur TBZ Webseite](http://www.tbz.ch)
```

Das sieht dann so aus:

Link zur [TBZ Webseite](#)

[Link zur TBZ Webseite](#)

Dieses Dokument wurde mit [Markdown](#) geschrieben

Tabellen

Tabellen können folgendermassen definiert werden. Die Doppelpunkte in der zweiten Zeile definieren die Ausrichtung.

```
|Model| Farbe|Stückzahl|
|:--:|:--|--:|
|Flott|grün|5|
|Flink|rot|10|
|Hui|blau|0|
```

Model	Farbe	Stückzahl
Flott	grün	5
Flink	rot	10
Hui	blau	0

Aufzählungen

```
* erstens
* zweitens
```

- erstens
- zweitens

oder mit Nummerierung. Bei Eingabe von [Enter](#) wird auf der neuen Zeile die Nummerierung fortgesetzt.

```
1. erstens
2. zweitens
3. drittens <enter>
4.
```

1. erstens
2. zweitens
3. drittens
- 4.