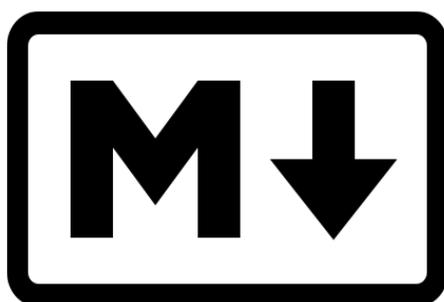
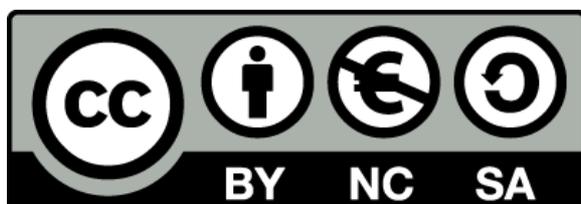


Guida all'uso del linguaggio Markdown



V. 1.3.8 del 20/06/2020

<https://mastodon.uno/@fenix>



Quest'opera è stata rilasciata con licenza Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 3.0 Italia. Per leggere una copia della licenza visita il sito web <http://creativecommons.org/licenses/by-nc-sa/3.0/it/> o spedisci una lettera a Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Indice generale

Crediti.....	4
Capitolo 1.....	5
INTRODUZIONE AL LINGUAGGIO.....	5
Perché usare Markdown?.....	7
A cosa serve Markdown? Anche a fare siti web.....	8
Creazione Documenti.....	9
Creazione Libri.....	9
Applicazioni consigliate che supportano CommonMark.....	10
Applicazioni, editor, online che supportano il linguaggio.....	11
Capitolo 2.....	12
SINTASSI DEL LINGUAGGIO.....	12
Intestazioni o titoli.....	12
Paragrafi.....	17
Enfasi: Scrivere in corsivo (italico).....	19
Enfasi: Scrivere in grassetto.....	20
Enfasi: Scrivere in grassetto corsivo.....	20
Enfasi: Scrivere in barrato.....	21
Liste.....	22
Liste non ordinate (ul).....	22
Liste ordinate (ol).....	23
Linea orizzontale (hr).....	27
Citazioni.....	28
Checklist.....	33
Link e immagini.....	34
Link con etichetta nel collegamento.....	36
Aggiungere una descrizione al link con etichetta.....	37
link con chiave di riferimento.....	37
Inserire una immagine presente in una cartella nel PC.....	39
Inserire una immagine attraverso un URL.....	42
Immagini linkate.....	43
Codice e testo quotato.....	45
Rappresentare il codice per blocchi.....	45
Rappresentare più righe come codice.....	46
Rappresentare il codice inline.....	47
Note a piè di pagina.....	48

Inserire una tabella.....	49
Formatazione celle.....	57
Specifica Markdown Extra per tabelle semplici.....	58
Tabelle mono colonna.....	59
Inserire formule matematiche.....	60
Inserire una formula.....	60
Radicali: $\sqrt{\quad}$	61
Frazioni: $\frac{\quad}{\quad}$	63
Parentesi.....	63
Indici.....	64
Potenze.....	64
Limiti.....	65
Logaritmi.....	65
Derivate (come frazione).....	66
Integrali.....	66
Serie.....	67
Matrici.....	67
Sistemi: lineari e sistemi in generale.....	68
Pandoc: formula matematica inline.....	69
Risorse utili latex.....	70
MathML vs Mathjax.....	71
Sintassi Markdown come caratteri letterali.....	74
Inserire codice in HTML nel documento markdown.....	75
Evidenziare il testo usando il codice HTML.....	76
Scrivere pedici e apici usando il codice HTML.....	76
Immagine a sinistra con testo al lato.....	76
Immagine a destra con testo al lato.....	77

Crediti

A questa guida hanno contribuito le seguenti persone:

Capo progetto e principale ideatore:

Fenix <https://mastodon.uno/@fenix>

Collaborazioni, idee, suggerimenti, critiche, revisioni, supporto morale:

Writefreely italia <https://mastodon.uno/@writefreely> <https://noblogo.org/writefreely/>

collettivo devol <https://mastodon.uno/@devol> <https://noblogo.org/>

Filippodb <https://mastodon.uno/@filippodb> <https://mastodon.uno> <https://diggita.com/>

Nilocram <https://framapiaf.org/@nilocram> [guida markdown nilocram](#)

Bibliografia open o in creative commons, materiale vario:

<https://www.markdownguide.org/getting-started/>

Un manuale in inglese scritto da Mat Cone interamente in markdown

Con licenza: [Creative Commons Attribution-ShareAlike 4.0 International license](#).

<https://github.com/mattcone/markdown-guide-book>

Capitolo 1

INTRODUZIONE AL LINGUAGGIO

L'obiettivo di questo linguaggio all'atto della sua creazione era di permettere alle persone di scrivere usando un formato di testo semplice da scrivere e leggere e opzionalmente, di convertirlo in XHTML (o HTML) strutturalmente valido o pdf.

Dalla sua nascita, Markdown è stato utilizzato in maniera molto ampia: come semplice metodo per pubblicare commenti, per scrivere file ReadMe, per scrivere documentazione di programmi, vedesi Github, OpenStreetMap, SourceForge ecc.. ciò non toglie che viene utilizzato in tantissimi altri ambiti, per citarne alcuni, blog, forum, social-network ecc.

Markdown è il linguaggio di marcatura inventato da Aaron Swartz e John Gruber. **CommonMark** è il dialetto che si propone di superare queste limitazioni e di uniformare le convenzioni.

Alla buona, il **Markdown** è il *formato stretto* creato nel 2004, **CommonMark** è il Markdown esteso supportato da **Pandoc**.

Un testo scritto in CommonMark o per abbreviare in Markdown può essere convertito in un formato di pubblicazione come HTML, PDF, EPUB o LaTeX attraverso dei compilatori appositi o editor specifici per markdown, tra gli strumenti di conversione più versatili e potenti, il mio consiglio non può che ricadere su **pandoc**.

<https://pandoc.org/>

<https://commonmark.org/>

GitHub Flavored Markdown Spec:

<https://github.github.com/gfm/>

Per scrivere in CommonMark si scrive del semplice testo a cui si applicano dei marcatori. Un marcatore è un segno convenzionale per indicare la formattazione. CommonMark è un linguaggio di marcatura leggero che usa espressioni codificate

abbastanza semplici da essere utilizzabili durante la fase creativa di scrittura. Per quanto riguarda il CommonMark o non tutti gli editor per Markdown riconoscono la sintassi più estesa del CommonMark. Quindi il campo della scelta dell'editor o del compilatore per scrivere il nostro documento si restringe.

Quindi iconsiglio vivamente di usare editor che supportano il linguaggio CommonMark .

Altrimenti rischiamo che quando andiamo a esportare il nostro documento dal formato .md (Markdown) nel formato PDF, DOC, ODT, HTML o TeX non vengono interpretate determinate istruzioni di formattazione del testo che ci sono nel linguaggio CommonMark, ma non sono presenti nel Markdown stretto. Perché tutte le codifiche del Markdown sono presenti nel CommonMark, ma non viceversa.

Perché usare Markdown?

Potresti chiederti perché le persone usano Markdown invece di un editor WYSIWYG. Perché scrivere con Markdown quando è possibile premere i pulsanti in un'interfaccia per formattare il testo?

A quanto pare, ci sono un paio di ragioni diverse per cui le persone usano Markdown invece degli editor WYSIWYG.

- Markdown può essere utilizzato per tutto. Le persone lo usano per creare siti Web, documenti, note, libri, presentazioni, messaggi e-mail e documentazione tecnica.
- Markdown è portatile. I file contenenti testo in formato Markdown possono essere aperti praticamente con qualsiasi applicazione. Se decidi che non ti piace l'applicazione Markdown che stai attualmente utilizzando, puoi importare i tuoi file Markdown in un'altra applicazione Markdown. Ciò è in netto contrasto con le applicazioni di elaborazione testi come Microsoft Word che bloccano i tuoi contenuti in un formato di file proprietario.
- Markdown è indipendente dalla piattaforma. È possibile creare testo in formato Markdown su qualsiasi dispositivo che esegue qualsiasi sistema operativo.
- Markdown è a prova di futuro. Anche se l'applicazione che stai utilizzando smette di funzionare ad un certo punto in futuro, sarai comunque in grado di leggere il tuo testo in formato Markdown usando un'applicazione di modifica del testo. Questa è una considerazione importante quando si tratta di libri, tesi universitarie e altri documenti cardine che devono essere conservati indefinitamente.
- Markdown è ovunque. Siti Web come Reddit e GitHub supportano Markdown e molte applicazioni desktop e basate su Web lo supportano.

A cosa serve Markdown? Anche a fare siti web..

Markdown è un modo semplice e veloce per prendere appunti, creare contenuti per un sito Web e produrre documenti pronti per la stampa.

Non ci vuole molto per imparare la sintassi di Markdown, e una volta che sai come usarlo, puoi scrivere usando Markdown praticamente ovunque. La maggior parte delle persone usa Markdown per creare contenuti per il Web, ma Markdown è utile per formattare qualsiasi cosa, dai messaggi di posta elettronica alle liste della spesa.

Ecco alcuni esempi.. creare siti web !

Markdown è stato progettato per il Web, quindi non dovrebbe sorprendere che ci siano molte applicazioni appositamente progettate per la creazione di contenuti di siti Web.

Se hai familiarità con HTML, CSS e controllo delle versioni, dai un'occhiata a [Jekyll](#), un popolare generatore di siti statici che prende i file Markdown e crea un sito Web HTML. Un vantaggio di questo approccio è che GitHub Pages fornisce hosting gratuito per siti Web generati da Jekyll.

Se Jekyll non fa per te, scegli uno dei tanti altri generatori di siti statici disponibili.

<https://www.staticgen.com/>

Creazione Documenti

Markdown non ha tutti i fronzoli e abbellimenti degli elaboratori di testi come Microsoft Word, ma è abbastanza buono per creare documenti di base come appunti e lettere.

È possibile utilizzare un'applicazione di creazione di documenti Markdown per creare ed esportare documenti in formato Markdown in formato PDF o HTML. La parte PDF è fondamentale, perché una volta che hai un documento PDF, puoi fare qualsiasi cosa con esso: stamparlo, inviarlo via email o caricarlo su un sito Web.

In quasi tutti i modi, Markdown è la sintassi ideale per prendere appunti. Purtroppo, Evernote e OneNote, due delle più note applicazioni di note, al momento non supportano Markdown. La buona notizia è che molte altre applicazioni di note open-source supportano Markdown.

Creazione Libri

Stai cercando di pubblicare un romanzo da solo? Prova [Leanpub](#), un servizio che prende i tuoi file in formato Markdown e li trasforma in un libro elettronico. Leanpub pubblica il tuo libro in formato PDF, EPUB e MOBI. Se desideri creare copie tascabili del tuo libro, puoi caricare il file PDF o EPUB su un altri servizi di Direct Publishing.

Oppure utilizza **pandoc** per convertire il tuo libro in formato PDF, EPUB e MOBI.

Applicazioni consigliate che supportano CommonMark

Pandoc: <https://pandoc.org/> (compilatore multi piattaforma)

Pandoc converte tra una moltitudine di formati di documenti, supporta CommonMark come input e output, supporta **latex e matjax**, quindi puoi scriverci formule matematiche. Io personalmente consiglio sempre di installare anche Pandoc qualunque sia l'editor e il sistema operativo usato.

ReText: <https://github.com/retext-project> (Linux, Editor markdown)

Ghostwriter: <https://wereturtle.github.io/ghostwriter/> (editor Windows, Linux)

Remarkable: <http://remarkableapp.github.io> (editor linux, supporta mathjax)

MacDown: <https://macdown.uranusjr.com/> (editor open per macosx)

Mark Text: <https://github.com/marktext/marktext> (editor multi piattaforma)

Markor: [F-droid](#) [github](#) [G.play](#) (Text editor - Notes & ToDo- Android, supporta notazione matematica text e latex, funziona offline, si setta nelle impostazioni!)

Diary: [F-droid](#) [github](#) (diario personale, lavorativo, agenda Calendario – Android)

Notes : [F-droid](#) [github](#) (editor-note Android)

Applicazioni, editor, online che supportano il linguaggio

Il modo migliore per iniziare con Markdown è usarlo. È più facile che mai grazie a una varietà di strumenti gratuiti.

Non è nemmeno necessario scaricare nulla. Esistono diversi editor online di Markdown che puoi usare per provare a scrivere in Markdown.

Dillinger: è uno dei migliori editor online di Markdown. Basta aprire il sito e iniziare a digitare nel riquadro a sinistra. Un'anteprima del documento renderizzato appare nel riquadro a destra. <https://dillinger.io/>

WriteFreely: piattaforma di scrittura e pubblicazione minimalista, federata, incentrata sulla privacy che utilizza il linguaggio markdown per la stesura dei contenuti. <https://writefreely.org/>

La comunità **italiana** di **WriteFreely** ha una sua istanza: <https://noblogo.org>
l'iscrizione è aperta e gratuita!

Laverna: piattaforma web per prendere note, scrivere appunti, editor markdown multi piattaforma mac/win/linux.

Alternativa open **evernote**. <https://laverna.cc/>

Il collettivo devol ha un istanza italiana di tale piattaforma a questo indirizzo: <https://laverna.devol.it/>

Capitolo 2

SINTASSI DEL LINGUAGGIO

Intestazioni o titoli

Con intestazioni o sezioni o titoli mi riferisco a quelli che in HTML sarebbero i vari tag di (Heading) H, quindi H1-H7.

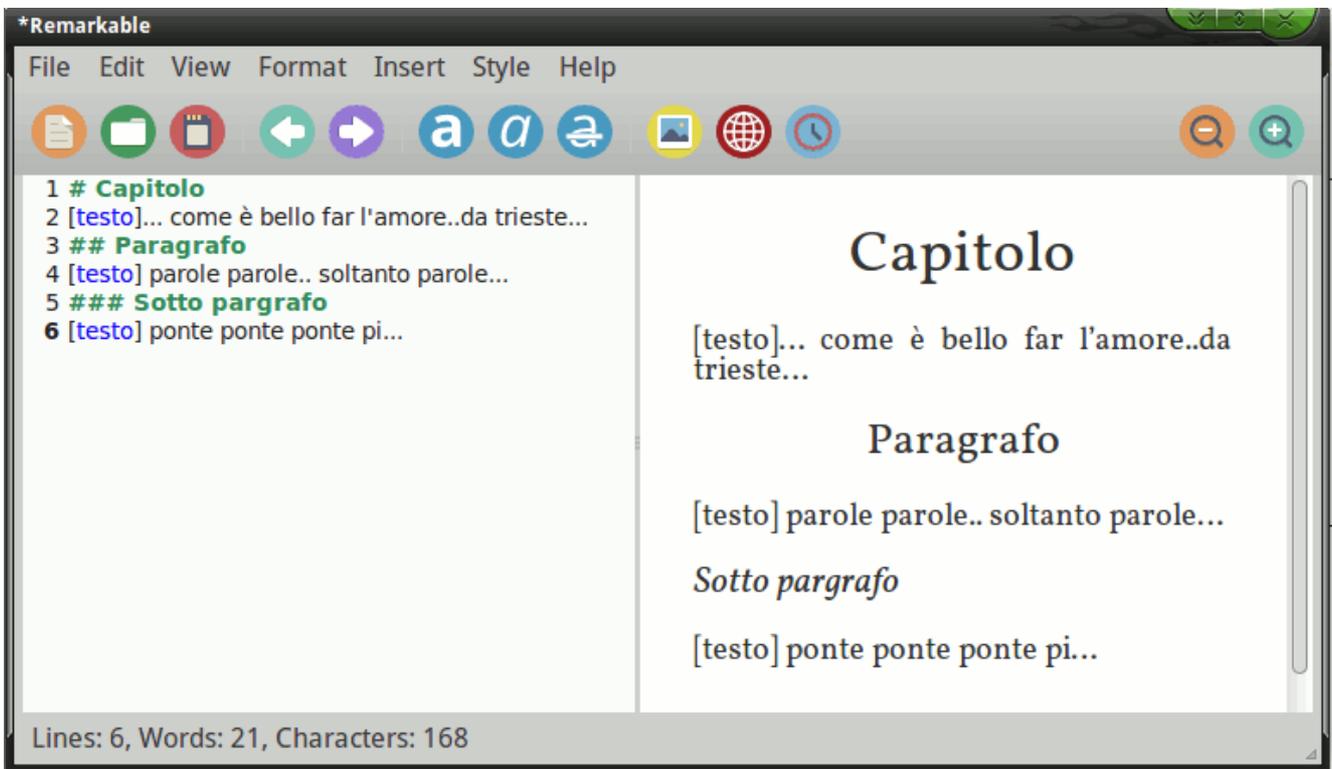
In Markdown creiamo le intestazioni aggiungendo il carattere #

```
# H1
## H2
### H3
#### H4
##### H5
##### H6
```

Si possono creare sezioni di livello inferiore usando un maggior numero di segni # sono disponibili fino a 6 livelli di sezioni, generabili usando da uno a sei segni #.

vediamo un esempio:

```
# Capitolo
[testo]... come è bello far l'amore..da trieste...
## Paragrafo
[testo] parole parole.. soltanto parole...
### Sotto paragrafo
[testo] ponte ponte ponte pi...
```



il risultato è simile a questo

In alternativa, solo per H1 e H2 possiamo usare quest'altro formato:

```
H1
===
H2
---
```

oppure così

```
Capitolo H1
=====
Capitolo H2
-----
```



Ma in questo modo non abbiamo una gerarchia, ma solo il livello più alto.

I livelli di heading sono automaticamente etichettati e ci si può riferire a loro come link interni al documento, questo è utile per generare toc o indici. Riferendoci all'esempio precedente:

```
Capitolo H1
=====
Capitolo H2
-----
clickando questo [link][Capitolo H2] si é indirizzati al Capitolo H2
```

Capitolo H1

Capitolo H2

clickando questo [link](#) si é indirizzati al Capitolo

da notare che questo funziona solo con pandoc, con altri editor potrebbe non funzionare come nel caso di remarkable.

Paragrafi

In Markdown per creare un nuovo paragrafo basta andare a capo due volte.

Cioè lasciando una riga vuota rispetto al paragrafo precedente.

Vediamo un esempio un po stupido..

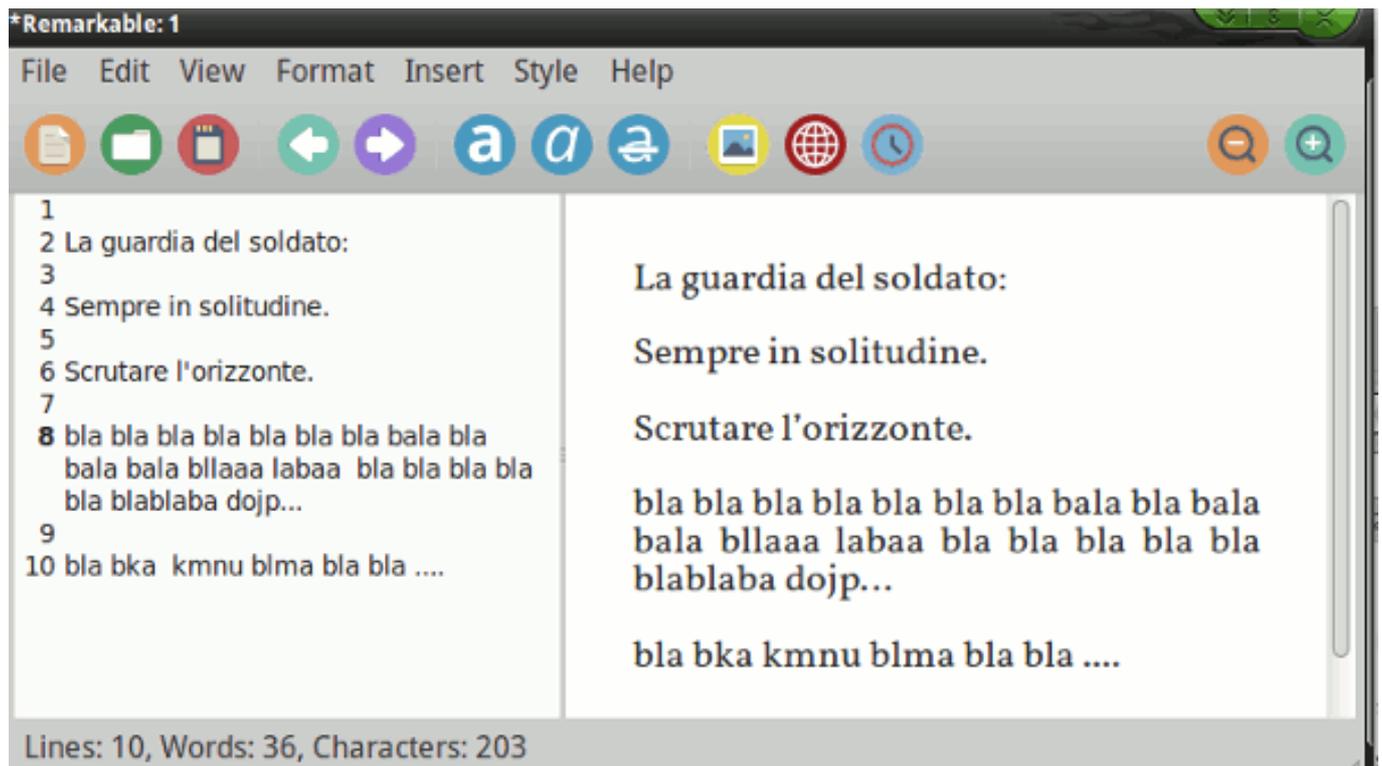
```
La guardia del soldato:
```

```
Sempre in solitudine.
```

```
Scrutare l'orizzonte.
```

```
bla bla bla bla bla bla bla bala bla bala bala bllaaa labaa bla bla  
bla bla bla blablaba dojp...
```

```
bla bka kmnu blma bla bla ....
```

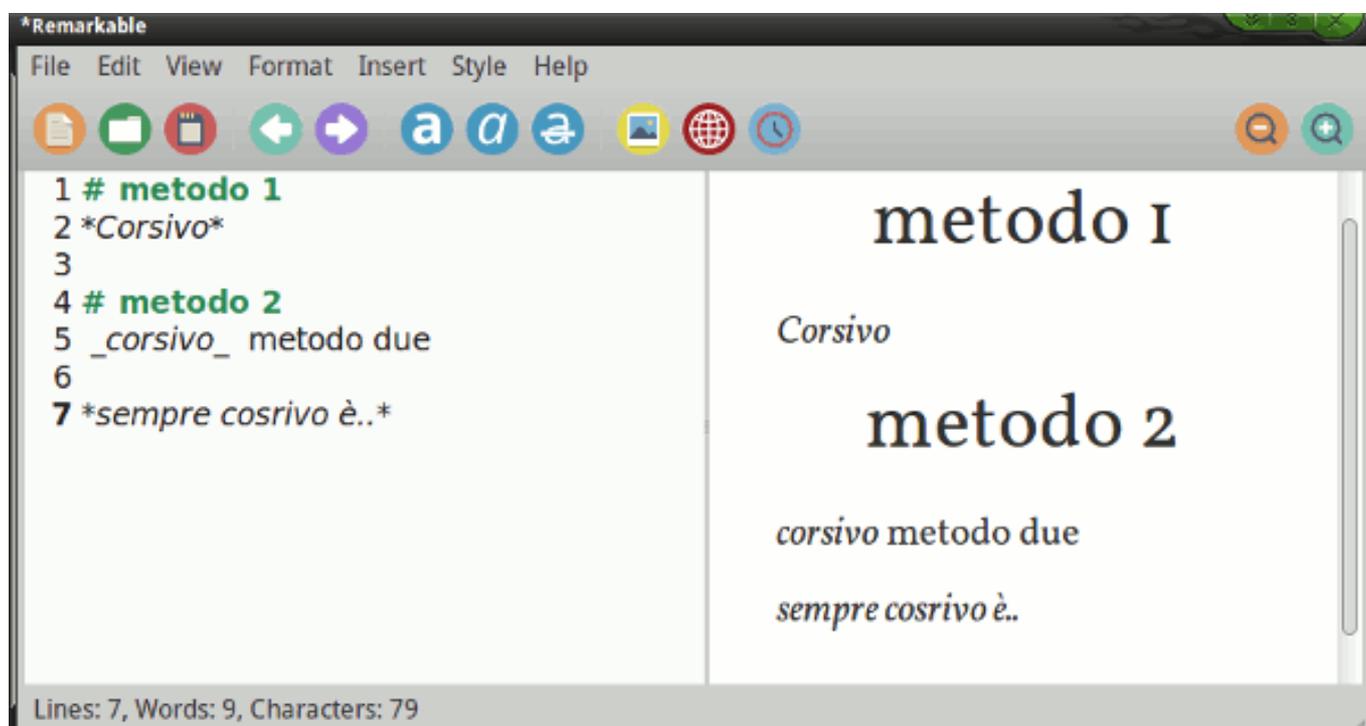


Enfasi: Scrivere in corsivo (italico)

```
# metodo 1
*Corsivo*

# metodo 2
_corsivo_ metodo due

*sempre cosrivo è..*
```

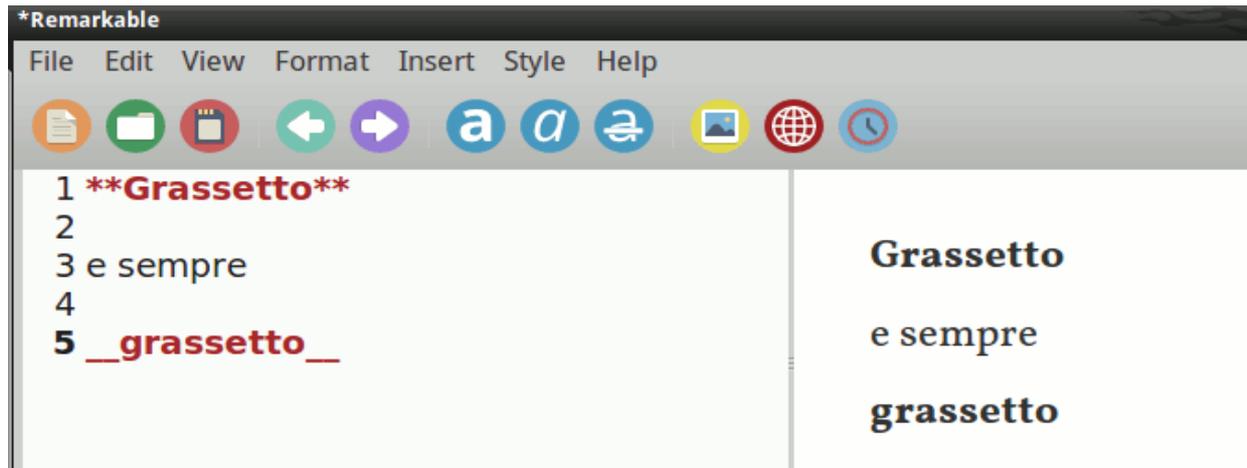


Enfasi: Scrivere in grassetto

```
**Grassetto**
```

è sempre

```
__grassetto__
```



Enfasi: Scrivere in grassetto corsivo

```
***Grassetto corsivo***
```

```
___corsivo grassetto___
```

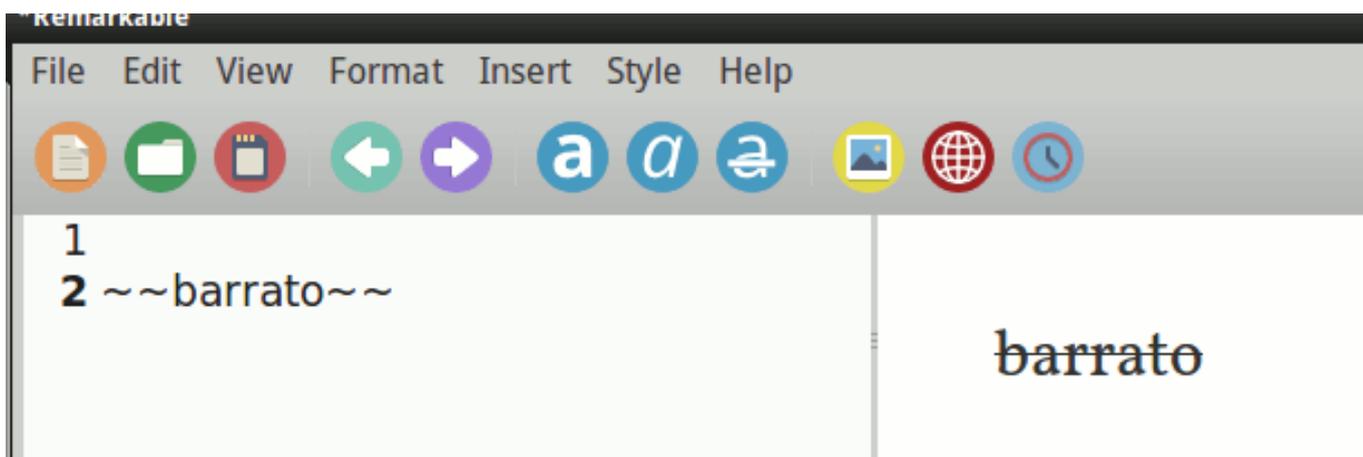
```
**_Grassetto corsivo_**
```



Enfasi: Scrivere in barrato

Scrivere una parola o frase tra quattro tilde (~) serve a barrare il testo. Il codice:

```
~~barrato~~
```



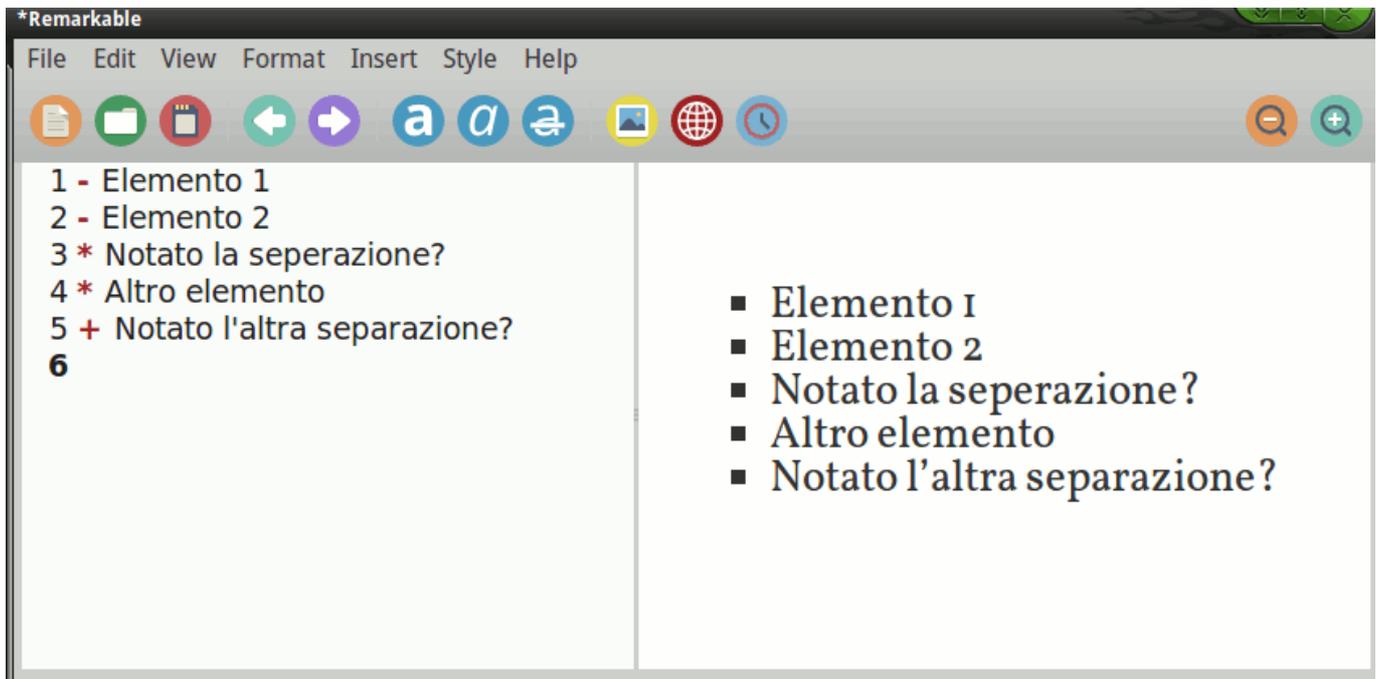
Liste

Scrivere liste con Markdown è tanto intuitivo quanto potente.

Liste non ordinate (ul)

Si possono usare asterischi, meno, più: portano a questo risultato!

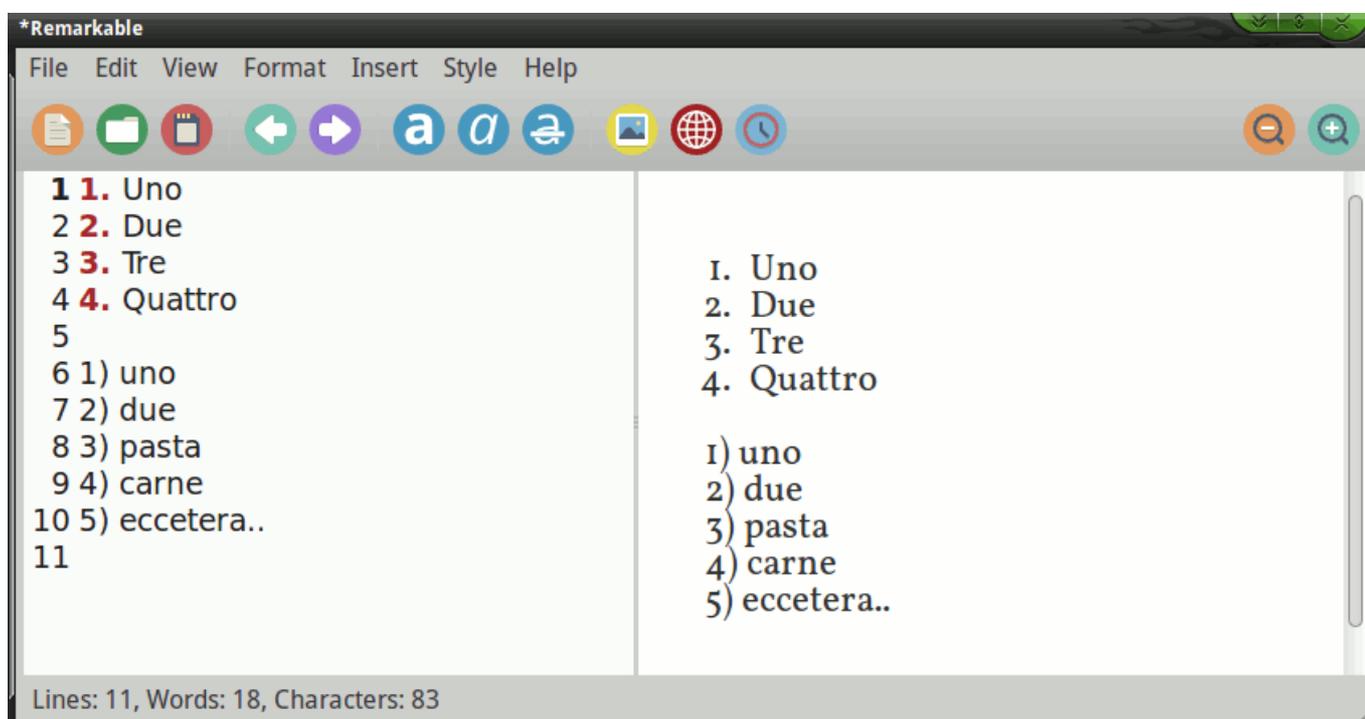
```
- Elemento 1
- Elemento 2
* Notato la seperazione?
* Altro elemento
+ Notato l'altra
```



Liste ordinate (ol)

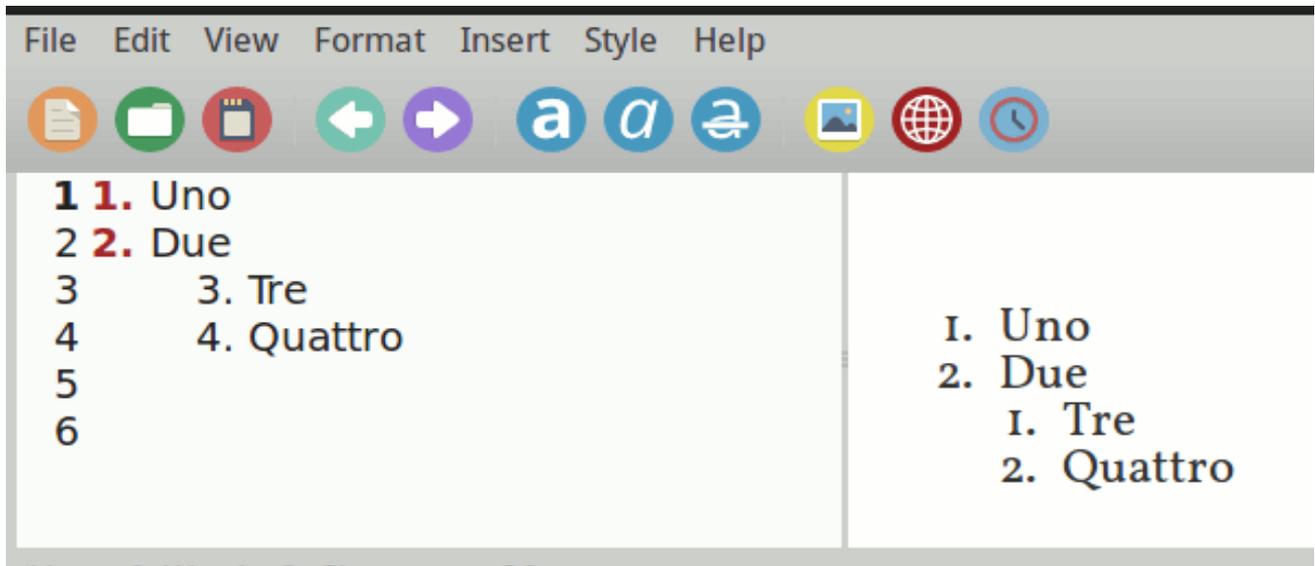
1. Uno
2. Due
3. Tre
4. Quattro

- 1) uno
- 2) due
- 3) pasta
- 4) carne
- 5) eccetera..



Sia per gli elenchi puntati che per quelli numerati, per aggiungere un livello ad una voce è sufficiente scrivere le sotto-voci dopo un TAB. oppure scrivere 3 spazi. Esempio:

1. Uno
2. Due
3. Tre
4. Quattro



da notare che questo funziona bene solo con pandoc, con altri editor potrebbe non funzionare proprio bene come nel caso di remarkable, non si riescono a fare liste annidate con la sintassi dei numeri con la parentesi

- ```

1) uno
 2) due
3) tre ecc..

```

sicuramente non è implementato, si consiglia di evitare questa situazione!!!

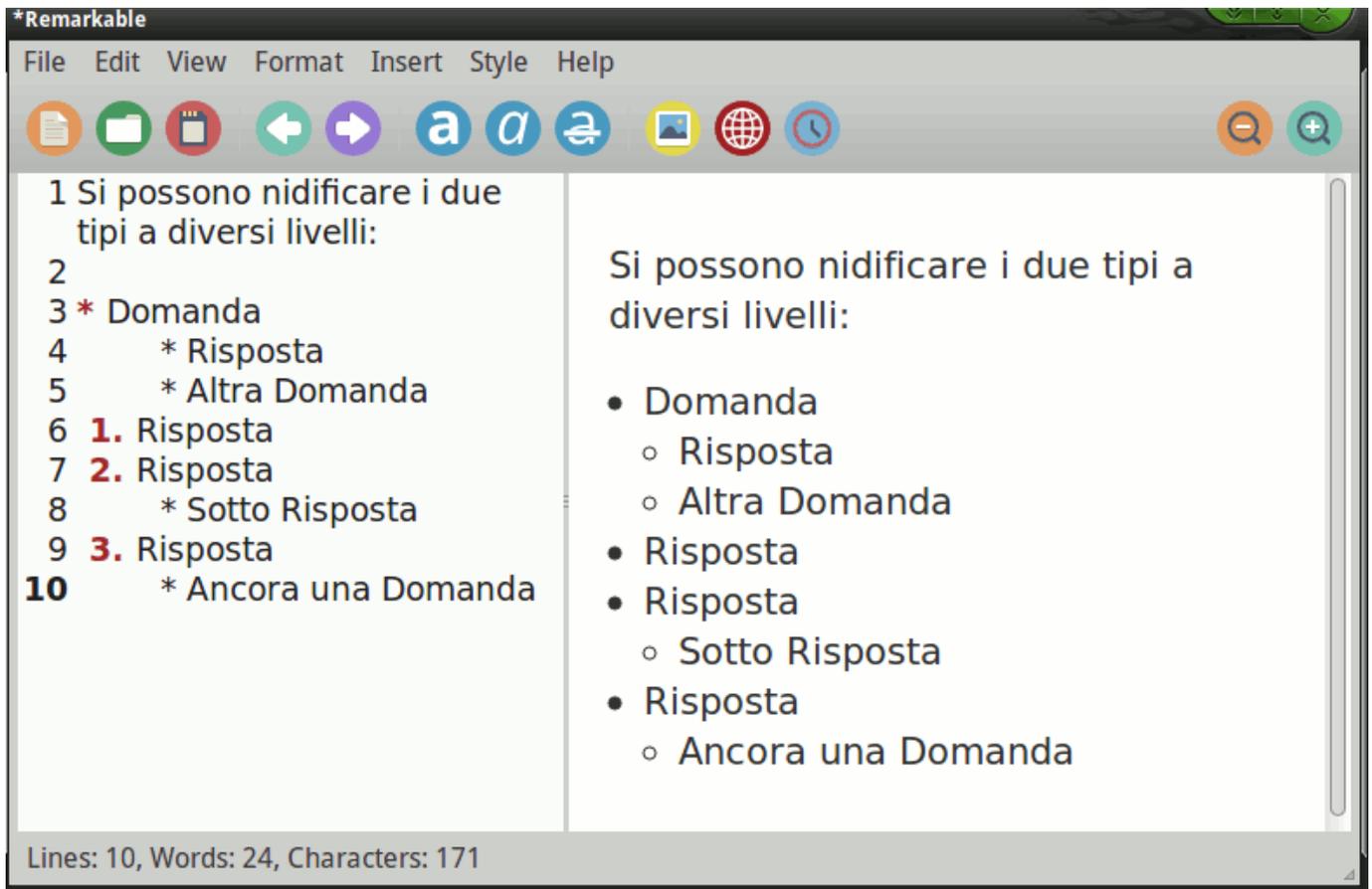
Si possono anche nidificare i due tipi a diversi livelli:

- ```

Si possono nidificare i due tipi a diversi livelli:

* Domanda
  * Risposta
  * Altra Domanda
1. Risposta
2. Risposta
  * Sotto Risposta
3. Risposta
  * Ancora una Domanda

```



questo è il risultato del caso, remarkable non gestisce correttamente le liste ordinate nidificate e miste, con pandoc è un'altra storia, se li diamo in pasto questo listato e apriamo l'html generato abbiamo questo:

```

Si possono nidificare i due tipi a diversi livelli:

* Domanda
  * Risposta
  * Altra Domanda

1. Risposta1
2. Risposta2
  * Sotto Risposta
3. Risposta3
* Ancora una Domanda

```

Si possono nidificare i due tipi a diversi livelli:

- Domanda
 - Risposta
 - Altra Domanda

- 1. Risposta1
- 2. Risposta2
 - Sotto Risposta
- 3. Risposta3
 - Ancora una Domanda

Linea orizzontale (hr)

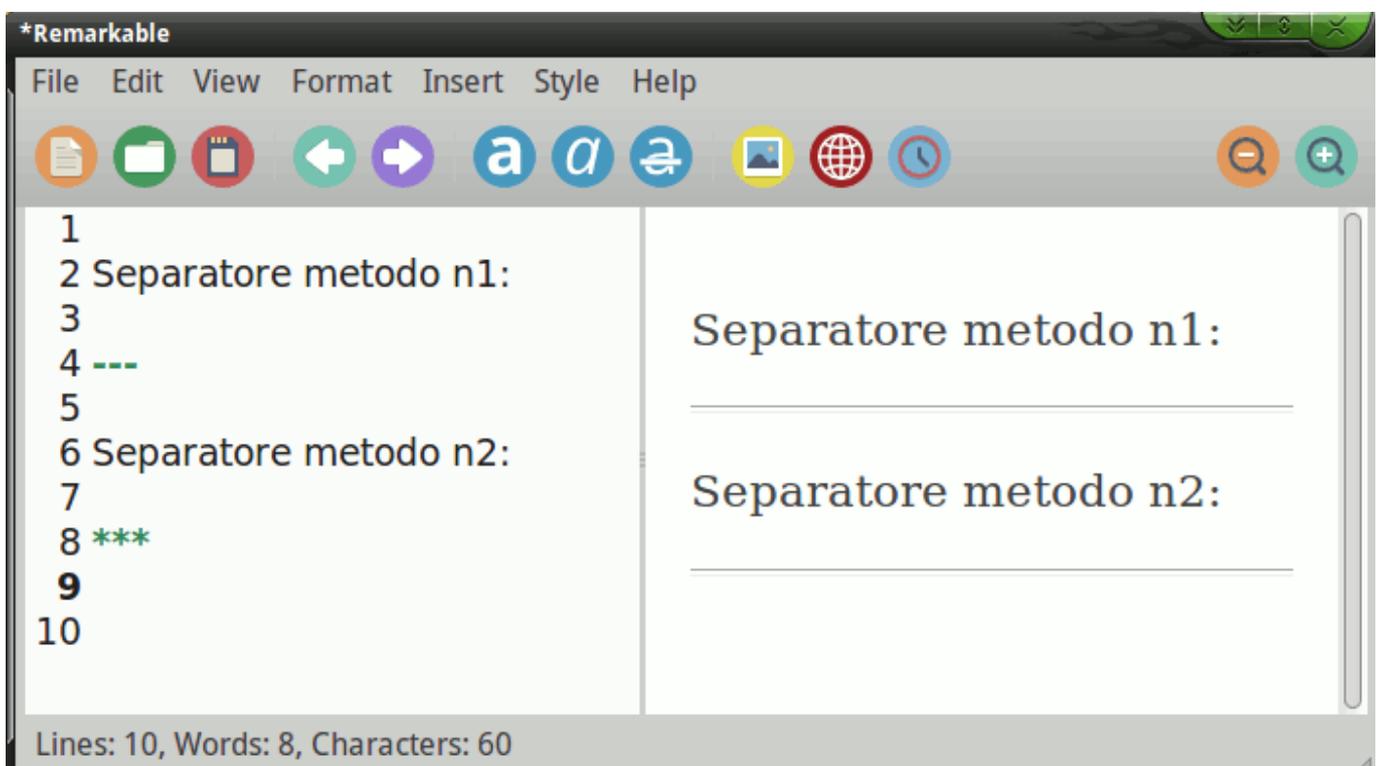
A volte può essere utile dare una separazione netta al testo. Con Markdown possiamo usare:

```
Separatore metodo n1:
```

```
---
```

```
Separatore metodo n2:
```

```
***
```



da notare se utilizziamo i trattini come separatore è bene utilizzare gli accapo per creare un nuovo paragrafo, se non lo facciamo il compilatore lo interpreta come h1.

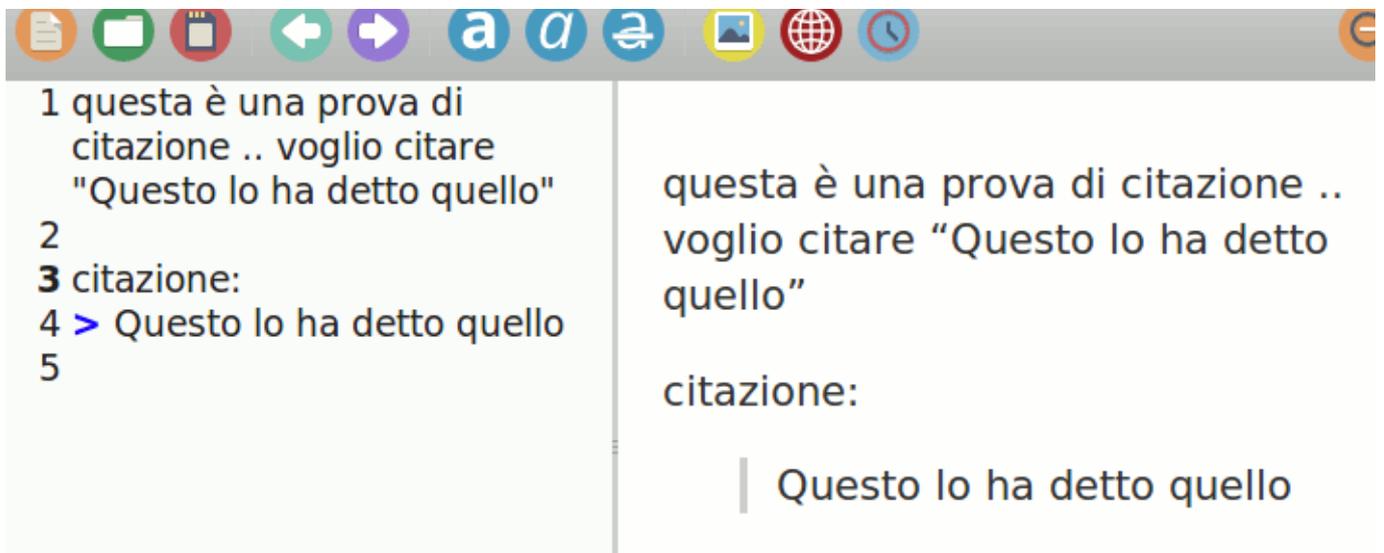
Citazioni

Markdown supporta la sintassi delle email per inserire una citazione.

Per rappresentare una citazione o un blocco di testo che risulti separato dal resto usare il carattere >

```
> Questo lo ha detto quello
```

verrà visualizzato come:



I blockquote o citazioni possono essere nidificati, aggiungendo livelli aggiuntivi di > :

```
> Questo è il primo livello di quotazione.  
>  
>> Questa è una blockquote nidificata.  
>  
> Torna al primo livello.
```

Questo è il primo livello di quotazione.

Questa è una blockquote nidificata.

Torna al primo livello.

I blockquotes o citazioni, possono contenere altri elementi Markdown, inclusi intestazioni, elenchi e blocchi di codice:

```
> ## Questa è un'intestazione.  
>  
> 1. Questa è la prima voce dell'elenco.  
> 2. Questa è la seconda voce dell'elenco.  
>  
> Ecco un esempio di codice:  
>  
> return shell_exec ("echo $ input | $ markdown_script");
```

Questa è un'intestazione.

- i. Questa è la prima voce dell'elenco.
- ii. Questa è la seconda voce dell'elenco.

Ecco un esempio di codice:

```
return shell_exec ("echo $ input | $ markdown_script");
```

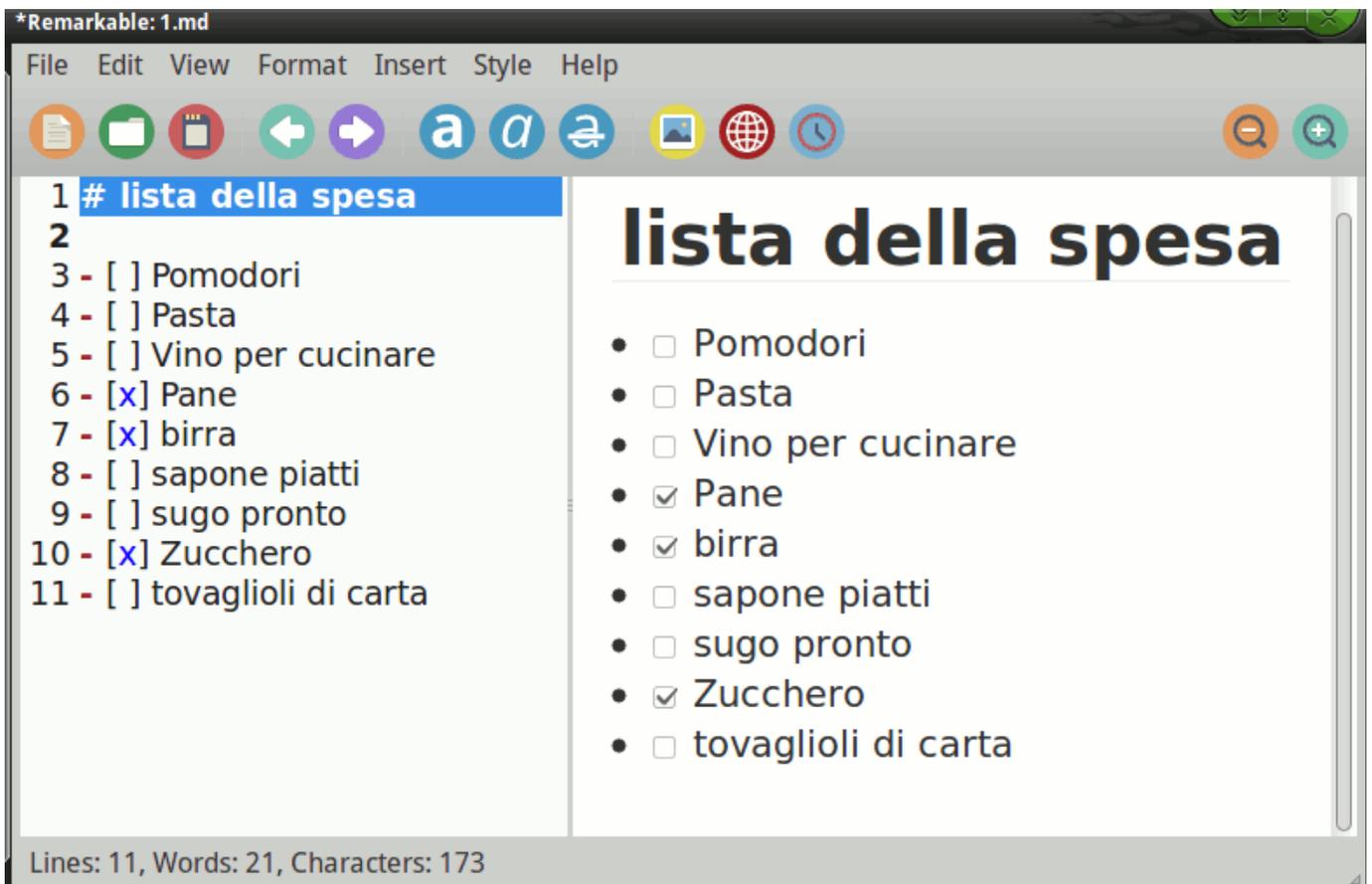
Checklist

Possiamo creare delle liste che hanno il quadratino da spuntare davanti alla voce. Queste liste possono essere usate come elenco di attività da fare, elenchi da spuntare di cose da acquistare, come la spesa o altro.

Oppure una lista di questo si può utilizzare per redigere moduli o questionari a risposte chiuse. Per scrivere un elenco di questo tipo digitare all'inizio della riga, il seguente codice: - [] oppure - [x]

```
# lista della spesa

- [ ] Pomodori
- [ ] Pasta
- [ ] Vino per cucinare
- [x] Pane
- [x] birra
- [ ] sapone piatti
- [ ] sugo pronto
- [x] Zucchero
- [ ] tovaglioli di carta
```



Link e immagini

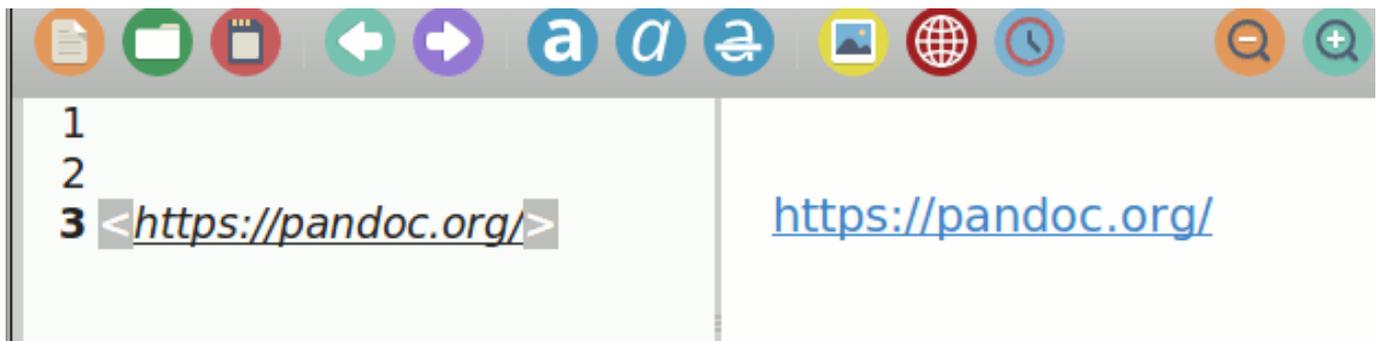
I link possono essere inseriti marcando il testo da linkare con parentesi quadre e inserendo in parentesi tonde il link a cui riferirsi.

É possibile utilizzare come link un identificativo e raggruppare tutti i link in un altro punto del documento, per esempio alla fine documento, in stile Wikipedia. Questo permette di aggiornare con una sola modifica tutti i link relativi a uno specifico identificativo, molto comodo se si compila un documento molto lungo o con diversi richiami a uno specifico link.

Per capire meglio vediamo subito qualche esempio pratico:

Automatic Links: URL da soli, o circondati da <> vengono automaticamente tradotti in link.

```
<https://pandoc.org/>
```



Link con etichetta nel collegamento

La sintassi del link con etichetta è: etichetta tra parentesi quadre e URL tra parentesi tonde: [nome](url)

Quindi se scriviamo il codice:

```
[Pandoc](https://pandoc.org/ )
```



Aggiungere una descrizione al link con etichetta

Per accompagnare il link a una descrizione metterla tra virgolette dopo l'URL:

```
[nome](url "descrizione")
```

Se scriviamo il seguente codice:

```
[Markdown Wiki](https://it.wikipedia.org/wiki/Markdown "leggi")
```



link con chiave di riferimento

Come precedentemente accennato è possibile utilizzare come link un identificativo e raggruppare tutti i link in un altro punto del documento, questa è una cosa molto utile, ci permette di gestire i link in maniere molto flessibile.

```
[testo da linkare][id]
```

```
testo del documento...
```

```
immagini
```

```
altro testo
```

```
[id]: http://indirizzo
```

Vediamo subito un esempio:

```
# link con chiave di riferimento
```

```
[Il motore di ricerca Google][google]
```

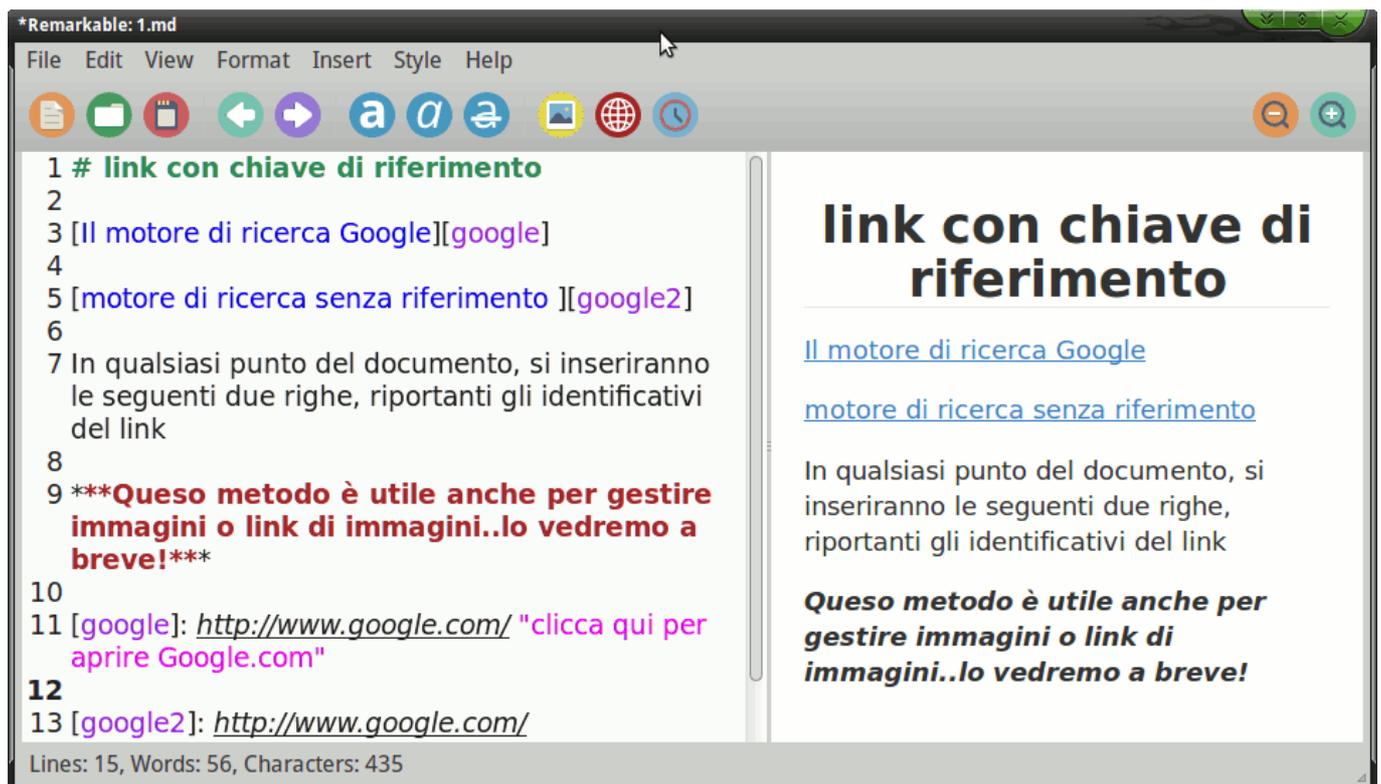
```
[motore di ricerca senza riferimento ][google2]
```

In qualsiasi punto del documento, si inseriranno le seguenti due righe, riportanti gli identificativi del link

Questo metodo è utile anche per gestire immagini o link di immagini..lo vedremo a breve!

```
[google]: http://www.google.com/ "clicca qui per aprire Google.com"
```

```
[google2]: http://www.google.com/
```



Inserire una immagine presente in una cartella nel PC

L'inserimento delle immagini usa la stessa sintassi dei link, ma preceduta da un punto esclamativo.

L'immagine deve essere inserita nella stessa cartella del documento.

Oppure all'interno di una cartella posizionata all'interno della stessa cartella del documento. Le immagini possono essere in formato JPG o formato PNG.

Immaginiamo di aver inserito nella cartella "immagini_markdown" un file col nome "gatto.jpg" con l'immagine di un gatto.

Digitiamo il seguente codice:

```
immagine nella cartella del documento
```

```
![gatto](gatto.jpg)
```

```
immagine dentro cartella di cortesia
```

```
![gatto](immagini_markdown/gatto.jpg)
```



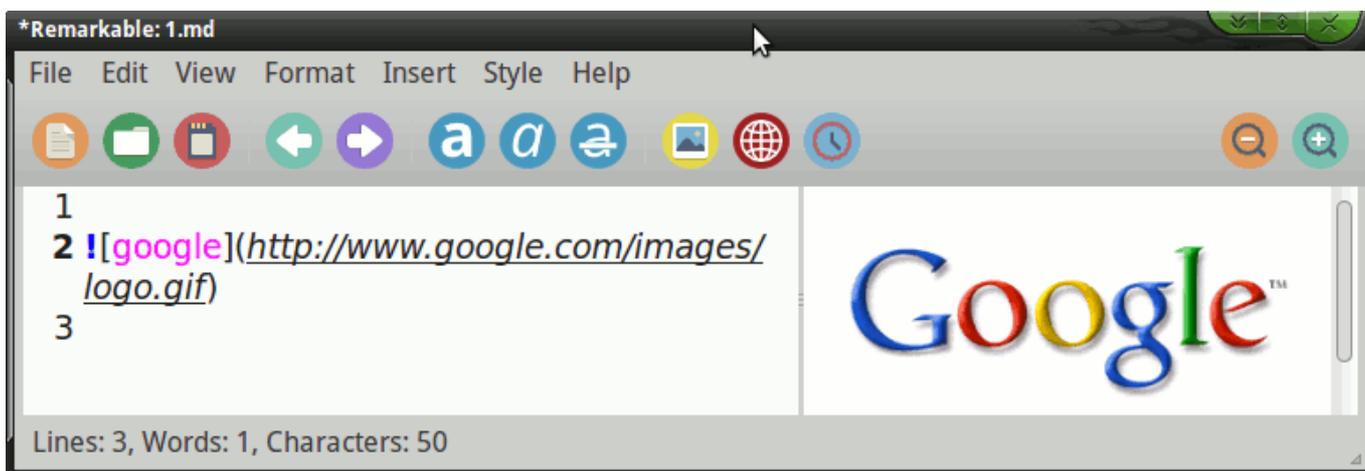
Inserire una immagine attraverso un URL

L'inserimento delle immagini attraverso un URL usa la stessa sintassi dei link, ma preceduta da un punto esclamativo. L'immagine deve essere presente in rete. Quindi ha un URL che permette di raggiungerla. Le immagini possono essere in formato JPG o formato PNG.

```
![nome_immagine](URL_immagine)
```

vediamo un esempio:

```
![google](http://www.google.com/images/logo.gif)
```



Possiamo usare i riferimenti etichetta visti in precedenza con i link anche per le immagini.

```
![google](http://www.google.com/images/logo.gif "pagina google")
```

Immagini linkate

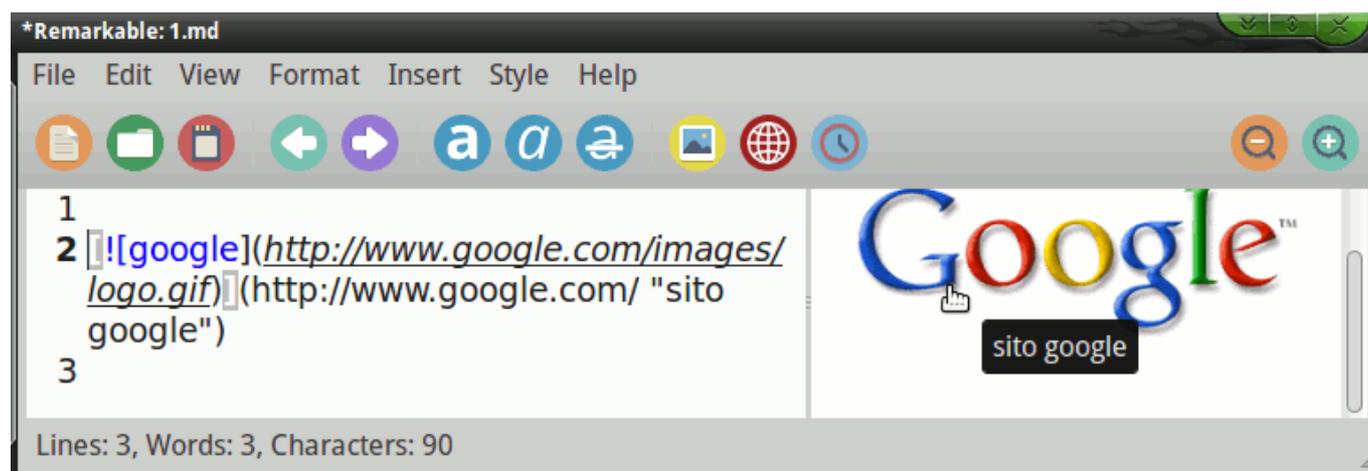
Le due cose possono essere combinate per creare immagini che sono anche link cliccabili.

```
sintassi link  
[link immagine](link)  
combinandola con la sintassi immagine diviene  
[![nome_immagine](URL_immagine)](URL_LINK)
```

vediamo l'esempio precedente come diviene:

```
[![google](http://www.google.com/images/logo.gif)]  
(http://www.google.com/ "sito google")
```

praticamente utilizziamo l'immagine di google come link a google.com



vedete bene che fare i trenini di immagini linkate , con molte immagini può divenire oneroso quindi possiamo gestire il tutto con “i link con chiave di riferimento” visti in precedenza.

In questo modo la gestione diviene molto più elegante.

La sintassi generale è questa:

```
[![testo_immagine][id_image_link]][id_link]
```

vediamo un esempio pratico:

[Il motore di ricerca Google][google]

![Il logo di Google][logo]

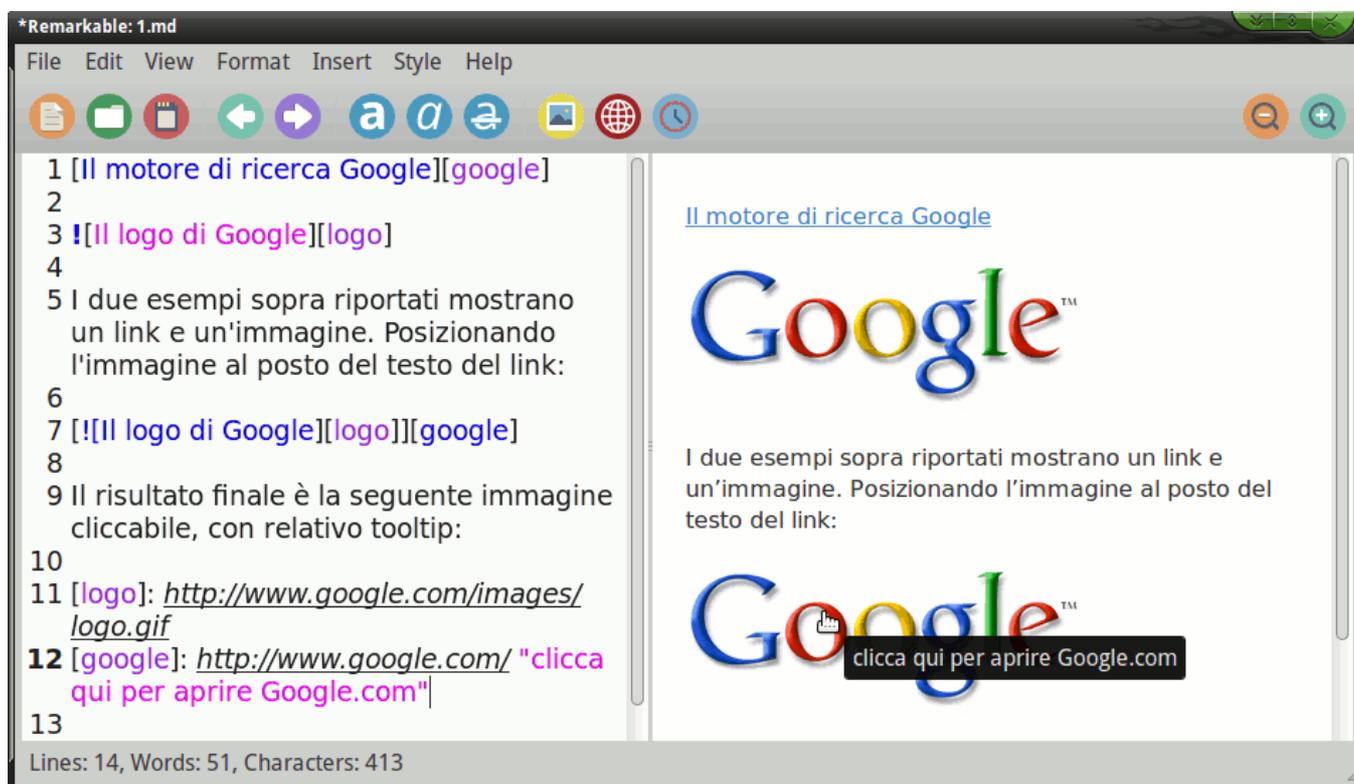
I due esempi sopra riportati mostrano un link e un'immagine. Posizionando l'immagine al posto del testo del link:

[![Il logo di Google][logo]][google]

Il risultato finale è la seguente immagine cliccabile, con relativo tooltip:

[logo]: <http://www.google.com/images/logo.gif>

[google]: <http://www.google.com/> "clicca qui per aprire Google.com"



Codice e testo quotato

Rappresentare il codice per blocchi

Frammenti di codice e di Markdown stesso possono essere visualizzati inserendo quattro spazi all'inizio della linea corrispondente o una tabulazione "tasto tab".

Per rappresentare del codice è possibile usare un blocco di testo con quattro spazi a inizio riga.

Scrivendo il seguente codice:

```
(digitare 4 spazi) blocco di testo rappresentato come codice
    blocco di testo rappresentato come codice
```

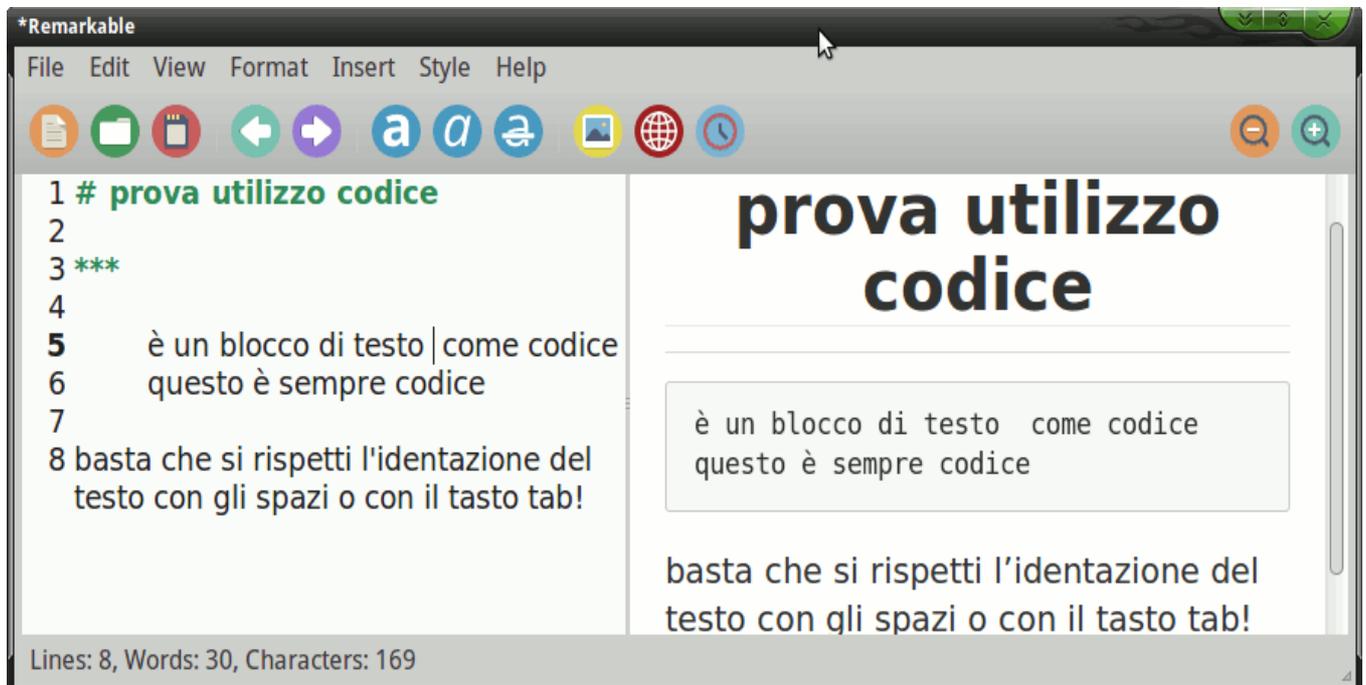
oppure

```
(digitare tab) blocco di testo rappresentato come codice
    blocco di testo rappresentato come codice
```

Si otterrà:

```
blocco di testo rappresentato come codice
```

vediamo un esempio:



con questo metodo il codice viene identificato tramite l'indentazione del testo, molto spesso però il codice da inserire nel nostro documento viene copiato in blocco da altri fonti, e sarebbe molto snervante stare ad indentare righe e righe

manualmente. Fortunatamente la lungimiranza di chi ha ideato il linguaggio ci ha fornito altri metodi per utilizzare i blocchi di codice.

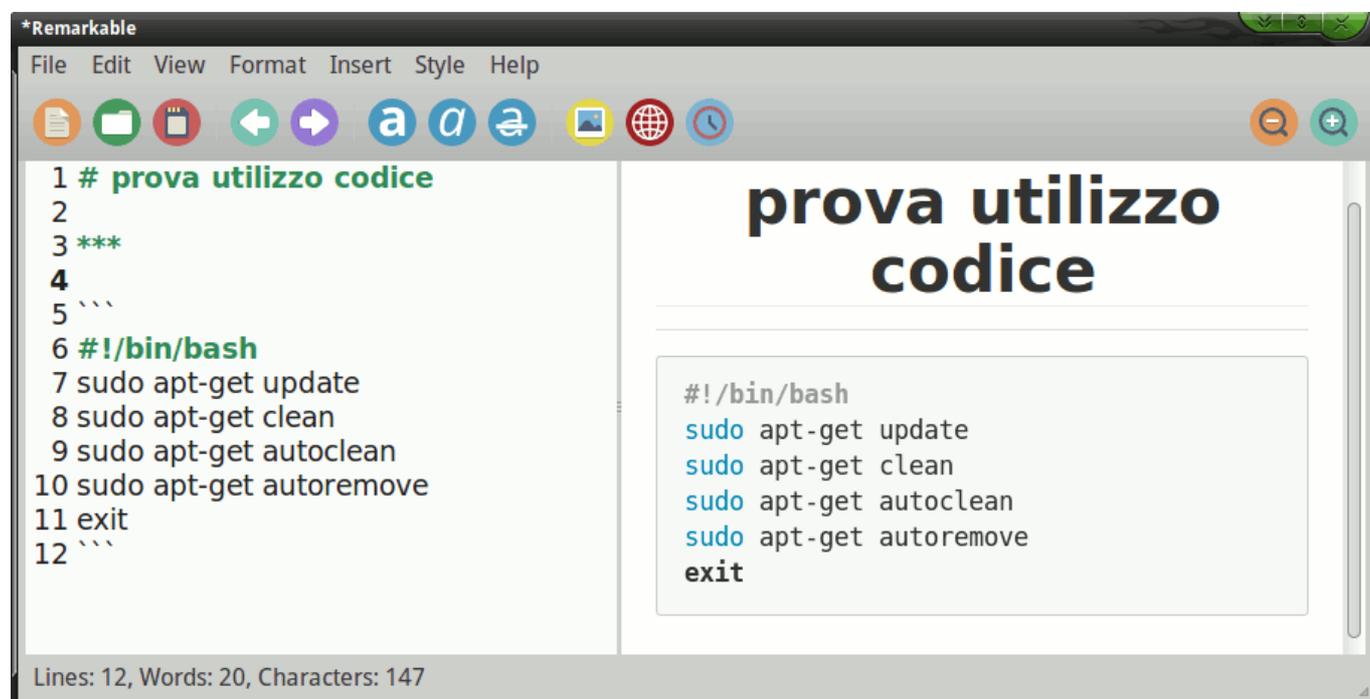
Rappresentare più righe come codice

Per rappresentare più righe o paragrafi come codice in maniera elegante e flessibile scrivere il seguente codice racchiuso tra ```

```
```\n\nrighe o paragrafi da rappresentare come codice\n\n```
```

Quindi per esempio se scriviamo il seguente codice:

```
```\n\n#!/bin/bash\nsudo apt-get update\nsudo apt-get clean\nsudo apt-get autoclean\nsudo apt-get autoremove\nexit\n```
```



Rappresentare il codice inline

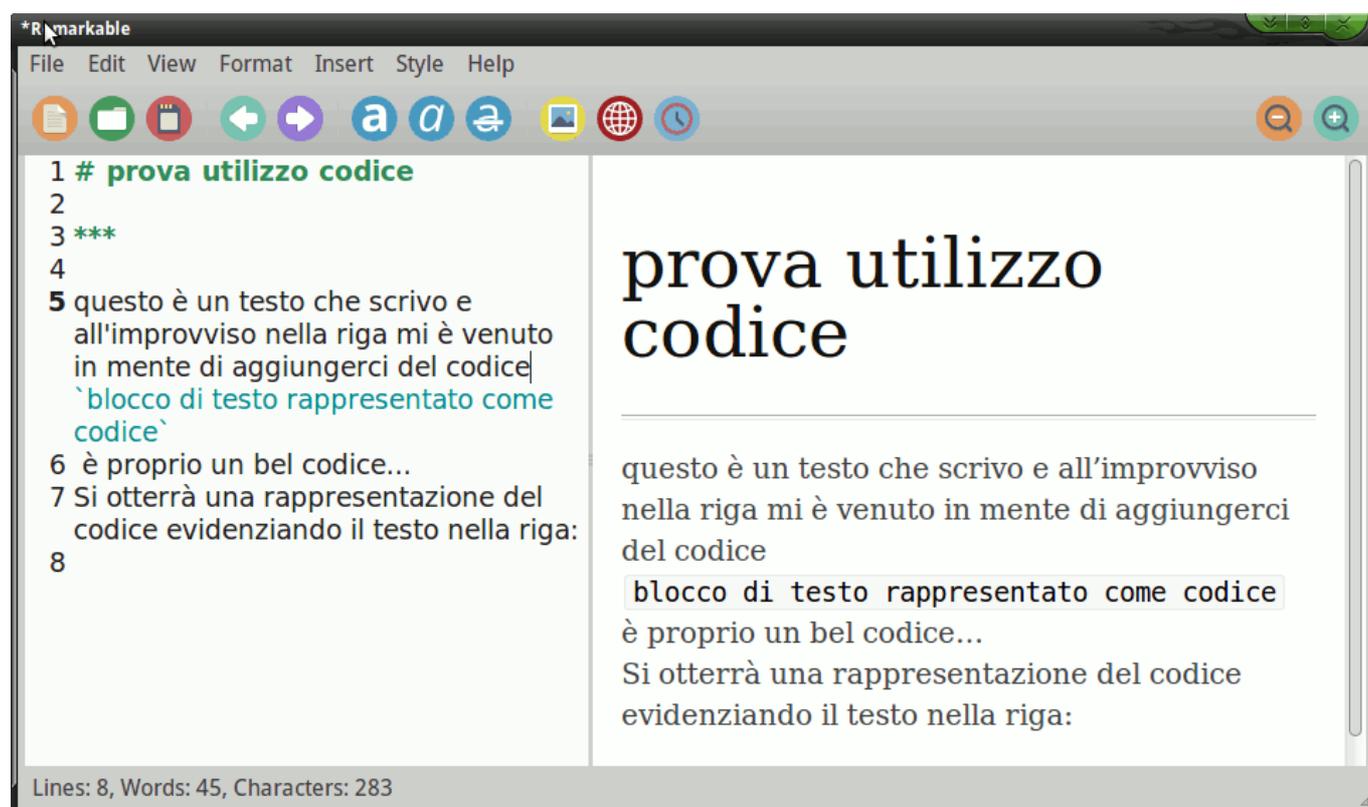
Per rappresentare del codice in linea o in riga è possibile mettere la parola o frase tra accenti a sinistra (backtick).

```
`blocco di testo rappresentato come codice`
```

Per far capire meglio scrivendo il seguente codice:

```
questo è un testo che scrivo e all'improvviso nella riga mi è venuto  
in mente di aggiungerci del codice `blocco di testo rappresentato  
come codice` è proprio un bel codice...
```

```
Si otterrà una rappresentazione del codice evidenziando il testo  
nella riga:
```



Note a piè di pagina

Per mettere una nota a piè di pagina mettere dopo la parola a cui si riferisce la nota.

Il codice della nota sarà il seguente:

```
parola [^numero nota]
```

E il codice del riferimento che scriviamo subito sotto sarà:

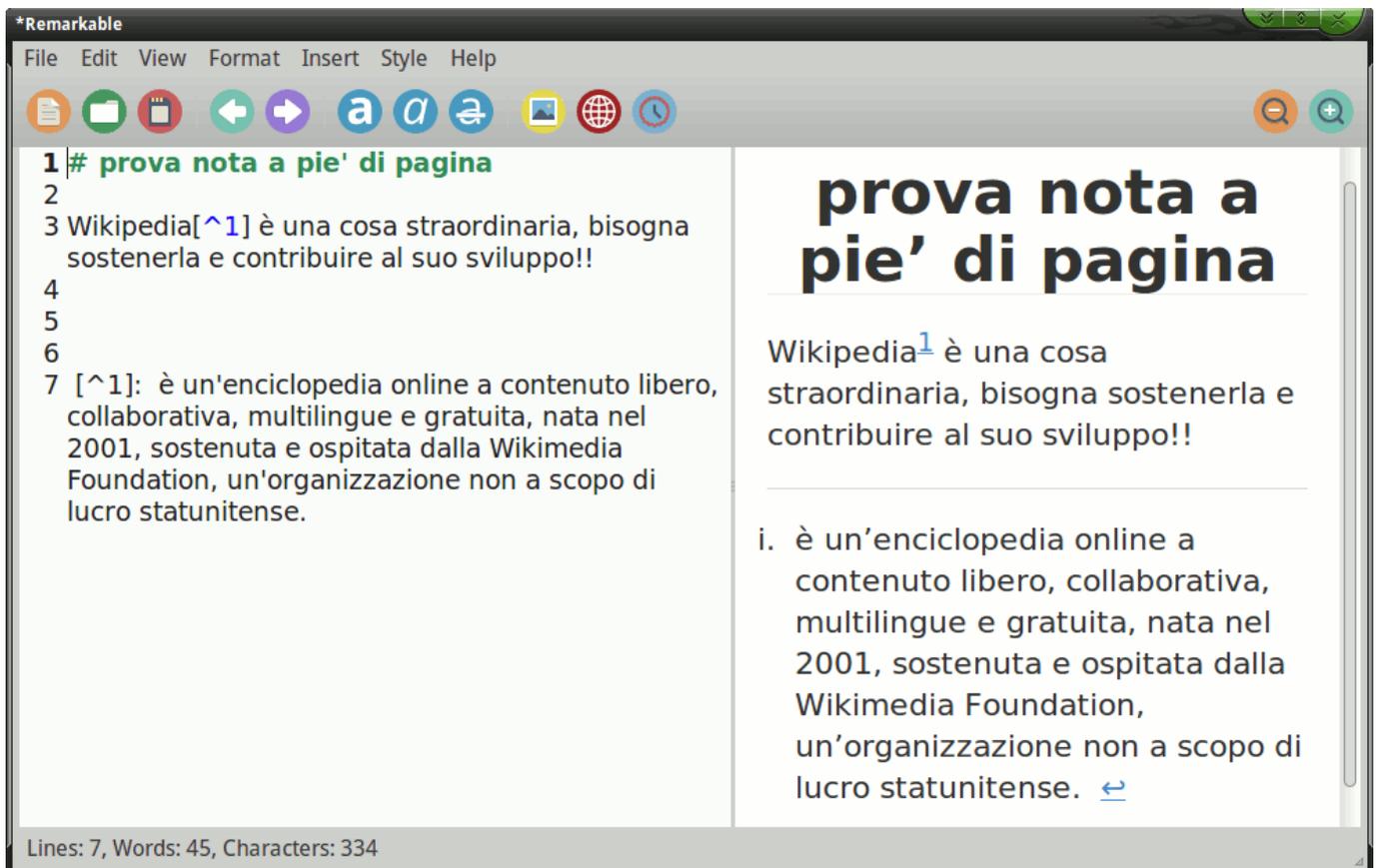
```
[^numero nota]: descrizione nota
```

vediamo subito un esempio:

```
# prova nota a pie' di pagina
```

```
Wikipedia[^1] è una cosa straordinaria, bisogna sostenerla e contribuire al suo sviluppo!!
```

```
[^1]: è un'enciclopedia online a contenuto libero, collaborativa, multilingue e gratuita, nata nel 2001, sostenuta e ospitata dalla Wikimedia Foundation, un'organizzazione non a scopo di lucro statunitense.
```



The screenshot shows the Remarkable application interface. The top menu includes File, Edit, View, Format, Insert, Style, and Help. Below the menu is a toolbar with various icons for editing and navigation. The main content area is split into two panels. The left panel shows the raw Markdown code for a document with a footnote. The right panel shows the rendered preview of the document, where the footnote is displayed as a separate block of text at the bottom of the page.

```
1 # prova nota a pie' di pagina
2
3 Wikipedia[^1] è una cosa straordinaria, bisogna
  sostenerla e contribuire al suo sviluppo!!
4
5
6
7 [^1]: è un'enciclopedia online a contenuto libero,
  collaborativa, multilingue e gratuita, nata nel
  2001, sostenuta e ospitata dalla Wikimedia
  Foundation, un'organizzazione non a scopo di
  lucro statunitense.
```

prova nota a pie' di pagina

Wikipedia¹ è una cosa straordinaria, bisogna sostenerla e contribuire al suo sviluppo!!

i. è un'enciclopedia online a contenuto libero, collaborativa, multilingue e gratuita, nata nel 2001, sostenuta e ospitata dalla Wikimedia Foundation, un'organizzazione non a scopo di lucro statunitense. ↵

Lines: 7, Words: 45, Characters: 334

Inserire una tabella

Le specifiche fondamentali di Markdown (stretto) non includono le tabelle; tuttavia, alcuni siti e applicazioni utilizzano varianti di Markdown che possono includere tabelle e altre funzioni speciali. [GitHub Flavored Markdown](#) è una di queste varianti e viene utilizzato sul sito GitHub per il rendering di file .md nel browser. Questa specifica è anche comunemente utilizzata in molti siti internet, blog, in alcuni editor moderni, in pandoc.

E' possibile inserire una tabella nel documento. La tabella si inserisce scrivendo:

SULLA PRIMA RIGA: l'intestazione delle colonne. Ogni colonna è separata da una pipe (|). Per esempio:

```
|prima colonna|seconda colonna|terza colonna|
```

SULLA SECONDA RIGA: il codice che indica per ogni colonna l'allineamento (a sinistra, a destra o al centro).

L'allineamento si indica aggiungendo al codice --- (3 segni consecutivi del trattino di sottrazione) il segno (dei due punti) secondo la seguente sintassi:

```
Codice per l'allineamento al centro: |:---:|  
Codice per l'allineamento a sinistra: |:---|  
Codice per l'allineamento a destra: |---:|
```

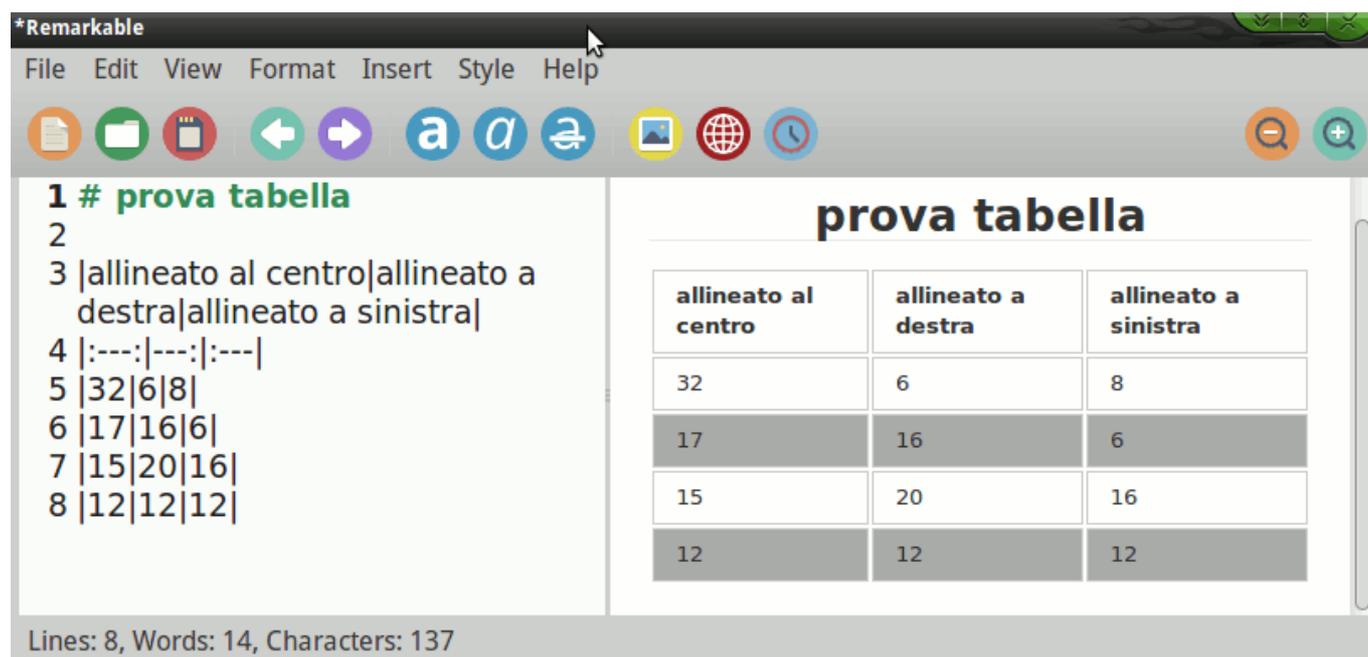
SULLE RIGHE SUCCESSIVE: si riportano i dati della tabella. I dati di ogni cella sono tra le due pipe:

```
Cella con contenuto: |contenuto cella|  
Cella vuota (priva di contenuto): | |
```

Vediamo ora con un esempio. Il seguente codice:

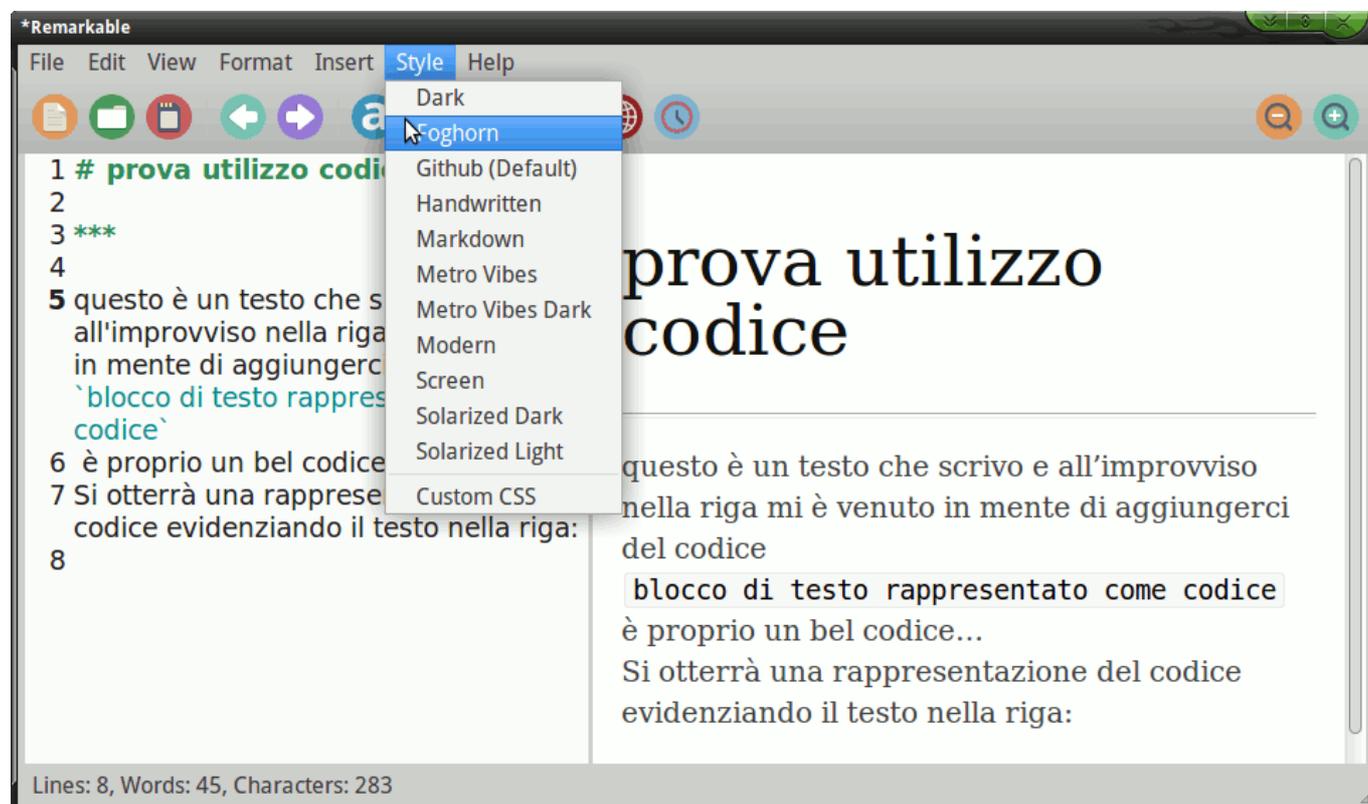
```
|allineato al centro|allineato a destra|allineato a sinistra|  
|:---:|---:|:---|  
|32|6|8|  
|17|16|6|  
|15|20|16|  
|12|12|12|
```

In alcuni editor del linguaggio markdown, le tabelle non sono perfettamente implementate e quindi producono un risultato simile a questo:



come vedete non è stato implementato l'allineamento della tabella che è una prerogativa di GFM ([GitHub Flavored Markdown](#)), o di [PHP Markdown Extra tables](#).

Come puoi notare dalle immagini precedenti di solito gli editor markdown come remarkable hanno una funzione integrata per iniettare lo stile dentro l'html generato, questo riguarda anche la visualizzazione delle tabelle!



usando **pandoc** abbiamo questo risultato:

prova tabella con pandoc senza custom css

allineato al centro allineato a destra allineato a sinistra

32	68
17	166
15	2016
12	1212

La struttura visiva non è delle più allettanti ma il codice html è perfettamente “allineato” con le istruzioni markdown è solo una questione di stile. Infatti nei moderni browser web non è più implementata una css style specifica per le tabelle visto che ormai le tabelle in xhtml e html5 si fanno in altro modo, quindi per abbellirla dobbiamo caricare all'interno un custom css per le tabelle oppure linkarlo in un file css esterno. Con il primo metodo ci tocca inserirlo manualmente per ogni file generato, con il secondo metodo invece basta costruire un file css apposito e linkarlo in automatico attraverso pandoc per tutti i nostri file.

```
table {
    font-family: verdana,sans,sans-serif;
    #font-size:12px;
    color:black;
    border-width: 1px;
    border-color: #3A3A3A;
    border-collapse: collapse;
}
th {
    border-width: 1px;
    padding: 10px;
    border-style: solid;
    border-color: #3A3A3A;
    background-color: #B3B3B3;
}
td {
    border-width: 1px;
    padding: 10px;
    border-style: solid;
    border-color: #3A3A3A;
    background-color: #ffffff;
```

```
}
```

questo in alto è un semplice codice di stile per le tabelle, vediamo come usarlo con pandoc (ovviamente il file css creato deve “seguire” sempre i nostri file html creati).

Supponiamo di salvarlo in un file chiamato “tabella.css”, lo invochiamo con pandoc in questo modo:

```
pandoc -s 1.md -c tabella.css -o 1.htm
```

1.md è il mio file di esempio, output del programma sarà 1.html ; vediamo il risultato!

prova tabella con pandoc e custom css

allineato al centro	allineato a destra	allineato a sinistra
32	6	8
17	16	6
15	20	16
12	12	12

Ovviamente ci possono essere una varietà di stili diversi, rispetto a quello usato da me, il limite è solo la fantasia e il gusto. .

Vediamo anche di ricreare uno stile simile a quello utilizzato da remarkable, chiamiamolo tabella3.css

```
table {
    font-family: verdana, arial, sans-serif;
    font-size: 11px;
    color: #333333;
    border-width: 1px;
    border-color: #3A3A3A;
    border-collapse: collapse;
}
th {
    border-width: 1px;
    padding: 8px;
    border-style: solid;
    border-color: #FFA6A6;
```

```

        background-color: #D56A6A;
        color: #ffffff;
    }
    tr:hover td {
        cursor: pointer;
    }
    tr:nth-child(even) td{
        background-color: #F7CFCF;
    }
    td {
        border-width: 1px;
        padding: 8px;
        border-style: solid;
        border-color: #FFA6A6;
        background-color: #ffffff;
    }
}

```

CSS Table Style 3

Grid Table With Interchange Row Color

ID	First Name	Last Name
235312	John	Doe
453123	Mark	Jones
998332	Jonathan	Smith
345612	Andrew	McArthur
453123	Adam	Fuller
998332	Tyler	Watt

Formattazione celle

Puoi applicare la formattazione al contenuto di ogni cella usando la normale sintassi Markdown.

Vediamo un esempio:

```
| Nome Funzione | Descrizione |
| ----- | ----- |
| `help()` | visualizza l' help window. |
| `destroy()` | **Distruggi il computer!** |
```

compiliamo con pandoc:

```
pandoc -s 1.md -c tabella.css -o 1.htm
```

Nome Funzione	Descrizione
<code>help()</code>	visualizza l' help window.
<code>destroy()</code>	Distruggi il computer!

notiamo che la colonna funzione è formattata come “codice” e in descrizione una delle celle è in grassetto!

Specifica Markdown Extra per tabelle semplici

Markdown Extra ha una sua sintassi per le tabelle semplici. Una tabella "semplice" si presenta così:

```
First Header | Second Header
----- | -----
Content Cell1 | Content Cell 2
Content Cell 3 | Content Cell4
```

compiliamo con pandoc e il css visto in precedenza:

```
pandoc -s 1.md -c tabella.css -o 1.htm
```

First Header	Second Header
Content Cell 1	Content Cell 2
Content Cell 3	Content Cell 4

Se lo desideri, puoi aggiungere una pipe all'inizio e alla fine di ciascuna riga della tabella. Usa il modo che ti piace di più. A titolo di esempio, questo darà lo stesso risultato di sopra:

```
| First Header | Second Header |
| ----- | ----- |
| Content Cell | Content Cell |
| Content Cell | Content Cell |
```

Tabella mono colonna

First Header

Content Cell
Content Cell

Nota: una tabella necessita di almeno una pipe su ogni riga affinché **Markdown Extra** l'analizzi correttamente. Ciò significa che l'unico modo per creare una **tabella a una colonna** è quello di aggiungere una pipe iniziale o di coda, o entrambi, in ciascuna riga. Questa particolare tabella **mono colonna** così formata non sempre è ben interpretata dai programmi, si potrebbero avere comportamenti inaspettati, quindi di solito si **sconsiglia il suo utilizzo**.

Inserire formule matematiche

Inserire una formula

Per scrivere le formule matematiche bisogna scriverle tra dei delimitatori:

2 segni del dollaro (\$): per scrivere una formula matematica all'interno del testo.

NB: questo metodo non è supportato da tutti gli editor o siti web, si consiglia nel dubbio, per compatibilità il metodo: formula staccata dal testo! NB: pandoc supporta pienamente questa notazione!

4 segni del dollaro (\$): per scrivere una formula staccata dal testo. (supportato da tutti!)

6 segni del dollaro (\$): per scrivere una formula di matematica composta da più blocchi di formule . (non supportato da tutti)

Ovviamente il linguaggio usato per le formule matematiche è quello di latex, interpretato poi da mathjax, o pandoc. Ovviamente questa è solo una guida di markdown e non di latex quindi faremo solo qualche esempio per le formule matematiche più comuni, che possono sempre tornar utili!

Scriviamo il codice di esempio:

```
$$ E = mc^{2} $$  
$$ A = \pi * r^{2} $$
```

$$E = mc^2$$

$$A = \pi * r^2$$

Produciamo le formule matematiche staccate dal testo:

Radicali: \sqrt

Vediamo subito un esempio:

```
$$ \sqrt{256}=16 $$
```

```
$$ \sqrt[5]{32}=2 $$
```

```
$$ \sqrt{y-w^{x}} $$
```

```
$$ \sqrt[89]{x+4} $$
```

$$\sqrt{256} = 16$$

$$\sqrt[5]{32} = 2$$

$$\sqrt{y - w^x}$$

$$\sqrt[89]{x + 4}$$

Frazioni: `\frac`

La sintassi generale è la seguente:

```
\frac{numeratore}{denominatore}
```

Ad esempio:

```
$$ \frac{x+1}{x-1} $$
```

$$\frac{x + 1}{x - 1}$$

Parentesi

La sintassi generale è la seguente:

```
\left( oggetto \right)
oppure
\left[ oggetto \right]
oppure
\left\{ oggetto \right\}
```

Ad esempio :

```
$$ \left(\frac{x}{y}\right) $$
```

$$\left(\frac{x}{y}\right)$$

Indici

La sintassi generale è la seguente:

```
termine_{indice}
```

Ad esempio:

```
$$ x_{n} $$
```

$$x_n$$

Potenze

La sintassi generale è la seguente:

```
base^{argomento}
```

Ad esempio:

```
$$ e^{x+4y} $$
```

$$e^{x+4y}$$

```
$$ r^{\frac{x}{u}} $$
```

$$r^{\frac{x}{u}}$$

Limiti

La sintassi generale è la seguente:

```
\lim_{variabile\to valore}{argomento del limite}
```

Ad esempio :

```
$$ \lim_{x\to 5}{x+1} $$
```

$$\lim_{x \rightarrow 5} x + 1$$

Logaritmi

La sintassi generale è la seguente:

```
\log_{base}{argomento}
```

Logaritmi naturali

```
\ln{argomento} | \log{argomento}
```

Ad esempio:

```
$$ \log_{7}{\frac{x-3}{2x}} $$
```

$$\log_7 \frac{x - 3}{2x}$$

```
$$ \ln\{3^{\{5\}}\} $$
```

$$\ln 3^5$$

Derivate (come frazione)

La sintassi generale è la seguente:

```
\frac{d}{dx}\left(funzione\right)
```

Ad esempio

```
$$ \frac{d}{dx}\left(\frac{x+4}{x}\right) $$
```

$$\frac{d}{dx} \left(\frac{x + 4}{x} \right)$$

Integrali

La sintassi generale è la seguente:

```
\int_{estremo basso}^{estremo alto}{integrand}
```

Ad esempio:

```
$$ \int_a^b x dx $$
```

$$\int_a^b x dx$$

Senza estremi di integrazione:

```
$$ \int x dx $$
```

$$\int x dx$$

Serie

La sintassi generale è la seguente:

```
\sum_{dicitura sotto}^{dicitura sopra}{argomento sommatoria}
```

Ad esempio :

```
$$ \sum_{i=1}^{5000}{5^i} $$
```

$$\sum_{i=1}^{5000} 5^i$$

Matrici

La sintassi generale è la seguente:

```
\left[\begin{matrix} ... & ... & ... \\ ... & ... & ... \\ ... & ... & ... \end{matrix}\right]
```

dove il simbolo "\\" separa una riga dalla successiva.

Se ad esempio volessimo scrivere:

```
$$ \left[\begin{matrix} a & b \\ c & d \end{matrix}\right] \left[\begin{matrix} x \\ y \end{matrix}\right] = \left[\begin{matrix} e \\ f \end{matrix}\right] $$
```

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}$$

Sistemi: lineari e sistemi in generale

La sintassi generale è la seguente:

```
\begin{cases} ... \\ ... \\ ... \end{cases}
```

dove il simbolo "\\" separa una riga dalla successiva.

Se ad esempio volessimo scrivere:

```
$$ \begin{cases} y=x \\ y=x+1 \end{cases} $$
```

$$\begin{cases} y = x \\ y = x + 1 \end{cases}$$

Pandoc: formula matematica inline

Come accennavo qualche pagina fa, la notazione text o latex per le formule matematiche inline non è implementata da tutti gli editor o piattaforme, e soprattutto non potendo testarlo su ogni programma esistente, mi limiterò ad indicarvi dove funziona senza alcun problema.. su **Pandoc**.

Per rinfrescare la memoria vi ricordo che la sintassi per la produzione inline di formule matematiche nel testo si esegue attraverso due marcatori di dollaro:

```
$FORMULA$
```

per vedere un esempio utilizziamo il seguente codice on un file 1.md da dare in pasto a pandoc:

```
puoi usare le inline-equations in questo modo: $y_i = \alpha + \beta x_i + e_i$
```

questa è una formula matematica staccata dal testo:

```
$$ 3^2+4^2=5^2 $$
```

la cosa importante è che la inline-equations deve essere delimitata senza spazi anteposti tra i marcatori di dollaro, altrimenti non viene interpretata!

```
$y_i = \alpha + \beta x_i + e_i$
```

ovviamente compiliamo con pandoc nel solito modo:

```
pandoc -s 1.md --mathjax -o 1.htm
```

puoi usare le inline equations in questo modo: $y_i = \alpha + \beta x_i + e_i$

questa è una formula matematica staccata dal testo:

$$3^2 + 4^2 = 5^2$$

Risorse utili latex

Per il momento gli esempi sull'uso di formule sono terminati, per riferimenti o approfondimenti di questo “fantastico mondo” vi lascio alla documentazione ufficiale di latex o le varie guide sparse online.

A scopo di studio o critica, qui sotto vi lascio i link alla documentazione che ho utilizzato per questi pochi esempi:

https://it.wikipedia.org/wiki/Aiuto:Formule_matematiche_TeX

<https://www.youmath.it/come-scrivo-le-formule-matematiche.html>

http://wwwcdf.pd.infn.it/AppuntiLinux/latex_ambienti_matematici.htm

L'Editor delle equazioni di CodeCogs è un simpatico servizio online che ti consente di scrivere con dei pulsanti le formule matematiche, per poi poter copiare il codice generato latex, o creare delle semplici immagini di formule:

<https://www.codecogs.com/latex/eqneditor.php>

MathML vs Mathjax

Il MathML (acronimo di Mathematical Markup Language, linguaggio di markup matematico) è un linguaggio web usato per rappresentare simboli e formule matematiche, che permette anche di attribuire un significato semantico alle formule. È stato derivato dall'**XML** come una specifica del gruppo di lavoro sulla matematica del W3C.

La struttura **XML** del linguaggio MathML permette una facile visualizzazione in applicazioni come i browser web, ma d'altro canto è di difficile comprensione rispetto al linguaggio TEX o LATEX.

Esempio:

```
$$ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} $$
```

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Può essere scritta in **MathML** nel seguente modo:

```
<math xmlns="http://www.w3.org/1998/Math/MathML" display="block">
  <mi>x</mi>
  <mo>=</mo>
  <mfrac>
    <mrow>
      <mo>&#x2212;<!-- - --></mo>
      <mi>b</mi>
      <mo>&#x00B1;<!-- ± --></mo>
      <msqrt>
        <msup>
          <mi>b</mi>
          <mn>2</mn>
        </msup>
        <mo>&#x2013;</mo>
        <mn>4</mn>
        <mi>a</mi>
        <mi>c</mi>
      </msqrt>
    </mrow>
    <mrow>
      <mn>2</mn>
      <mi>a</mi>
    </mrow>
  </mfrac>
</math>
```

```
</mrow>
</mfrac>
</math>
```

Come potete notare.. ci si capisce poco, però è uno standard consolidato, è qualsiasi browser moderno lo supporta, in oltre è ottimo per creare **documentazione off-line** perché non ha bisogno di internet e javascript per funzionare! Cosa di cui **mathjax** è dipendente.

Sono disponibili molti programmi in grado di convertire espressioni matematiche verso MathML, inclusi convertitori fra TEX e MathML. Uno di questi è **pandoc** !

Con **pandoc** è semplicissimo, supponiamo che il listato dell'esempio precedente sia stato salvato in un file chiamato 1.md

```
$$ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2^a} $$
```

il comando per per generare un file html in linguaggio MathML è il seguente:

```
pandoc 1.md -s --mathml -o 1.html
```

In oltre il formato MathML è supportato dai grandi software per l'ufficio come la suite open/libreoffice, infatti è il formato grezzo di archiviazione interno utilizzato per le formule matematiche. La cosa curiosa che la suite di programmi citata ha sviluppato un linguaggio tutto suoi per scrivere in MathML, quindi a meno che non utilizziamo qualche estensione possiamo scordarci di utilizzate il linguaggio text per le formule in tali suite di programmi per ufficio!

Se invece con **pandoc** vogliamo utilizzare **mathjax** l'esempio precedente diviene:

```
pandoc 1.md -s --mathjax -o 1.html
```

che darà come risultato questo listato html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-
8" />
```

```
<meta http-equiv="Content-Style-Type" content="text/css" />
<meta name="generator" content="pandoc" />
<title></title>
<style type="text/css">code{white-space: pre;}</style>
<script src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?
config=TeX-AMS-MML_HTMLorMML" type="text/javascript"></script>
</head>
<body>
<p><span class="math">\[ x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]</span></p>
</body>
</html>
```

da notare che il documento così generato per funzionare ha bisogno di internet, perché il rendering o (interpretazione) delle formule è svolta attraverso la CDN di mathjax:

```
"http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML\_HTMLorMML"
```

che per quanto strano sembri non si limita solo a scaricare qualche javascript e font matematici in locale, ma a livello server fa la conversione di formato da un linguaggio all'altro, e molto probabilmente restituisce le formule trovate in MathML o in formato immagine se MathML non è supportato dal browser in uso. In alcuni casi le formule convertite sono trasformate in Jason e interpretate dal javascript di mathjax. Questo dipende come si settano le invocazioni di mathjax. Per il momento non ci dilungheremo oltre sul funzionamento di mathjax, poiché questi approfondimenti esulano dallo scopo della guida, però per chi volesse trovare tutto nella documentazione ufficiale!

Sintassi Markdown come caratteri letterali

Il linguaggio Markdown ti permette di usare il carattere backslash (\) per generare caratteri letterali che avrebbero altrimenti un significato speciale nella sintassi di formattazione del Markdown.

Il linguaggio Markdown usa il carattere backslash (\) per generare i seguenti caratteri letterali:

```
\ backslash
` apice inverso
* asterisco
_ il segno di sottolineatura
{} parentesi graffe
[] parentesi quadre
() parentesi tonde
# cancelletto
+ il segno dell'addizione
- la linetta, o segno della sottrazione
! il punto esclamativo
```

Quindi il codice:

```
\**frase in grassetto\**
```

Invece della frase in grassetto produce la scritta:

****frase in grassetto****

oppure

```
\[prova\]
```

[prova]

oppure:

```
\# questo carattere è un cancelletto
```

genera una scritta che non è un intestazione h.

questo carattere è un cancelletto

Inserire codice in HTML nel documento markdown

I documenti markdown permettono l'inserimento del codice HTML. Questo permette di ampliare le funzionalità del documento in quanto attraverso il codice HTML possiamo inserire funzioni che non sono previste dal codice markdown. Questo ci permette anche di creare delle pagine web in cui usiamo normalmente il codice markdown ma secondo gli elementi che dobbiamo inserire possiamo semplicemente ampliare il nostro lavoro integrando codice in HTML.

Il codice HTML inserito in un documento markdown viene trattato dal codice markdown com se fosse del semplice testo. Quindi per esempio possiamo scrivere del testo colorato in html .

digitando il seguente codice HTML:

```
<font color="red">Testo in rosso</font>
```

Otteniamo:

testo in rosso

Aggiungendo poi i due asterischi del codice markdown ai lati del codice HTML digitando il codice:

```
**<font color="red">Testo in rosso</font>**
```

Otteniamo lo stesso testo in grassetto:

Testo in rosso

Quindi possiamo usare i due linguaggi insieme senza difficoltà.

Quindi spesso per redigere i nostri documenti e le nostre pagine web ci troveremo ad usare il codice markdown insieme con alcuni comandi di codice HTML.

Questo ci porterà ad un risultato migliore, più completo e con la massima comodità nel redigere il nostro documento.

Evidenziare il testo usando il codice HTML

Per evidenziare del testo digitare il seguente codice:

```
<span style="background-color:colore_testo">Testo evidenziato</span>
```

Quindi il seguente codice:

```
<span style="background-color:yellow">Testo evidenziato</span>
```

Produce testo evidenziato in giallo:

Testo evidenziato

Scrivere pedici e apici usando il codice HTML

Per scrivere un pedice in piccolo sotto il testo X scrivere il seguente codice:

```
X<sub>pedice</sub>
```

E produrremo:

X_{pedice}

Per scrivere una apice in piccolo sopra il testo X scrivere il seguente codice:

```
X<sup>apice</sup>
```

E produrremo:

X^{apice}

Immagine a sinistra con testo al lato

Per posizionare una immagine a sinistra con il testo di lato digitare il seguente codice HTML:

```
 testo al lato
```

Si avrà un risultato come questo:



Markdown è una sintassi per la formattazione di testo puro, progettata in modo che possa essere convertito in HTML utilizzando un semplice script. Attualmente è comunemente usato come formato per file readme, per la formattazione di messaggi nei forum o in editor di testo per la creazione rapida di documenti di testo formattati e in molti CMS di uso

comune. La sua semplicità lo rende facile da usare.

Nel codice c'è il codice HTML per definire spazi e bordi intorno all'immagine:

```
Hspace="15"    determina lo spazio orizzontale intorno all'immagine  
Vspace="0"    determina lo spazio verticale intorno all'immagine  
Border="0"    determina lo spessore del bordo intorno all'immagine
```

Immagine a destra con testo al lato

Per posizionare una immagine a sinistra con il testo di lato digitare il seguente codice HTML:

```
 testo al lato
```

Si avrà un risultato come questo:

Markdown è una sintassi per la formattazione di testo puro, progettata in modo che possa essere convertito in HTML utilizzando un semplice script. Attualmente è comunemente usato come formato per file readme, per la formattazione di messaggi nei forum o in editor di testo per la creazione rapida di documenti di testo formattati e in molti CMS di uso comune. La sua semplicità lo rende facile da usare.

