

Poznaj

Markdown

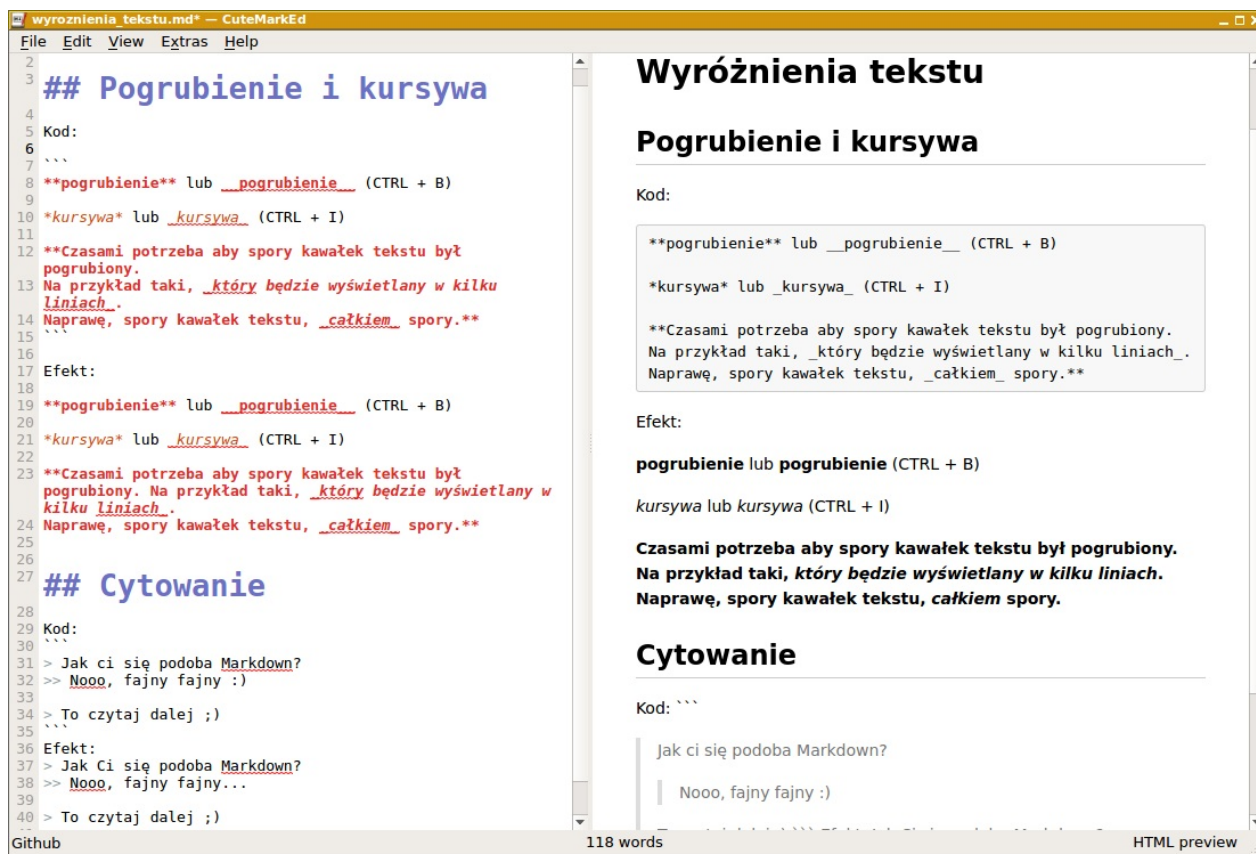
Spis treści

1. [Wstęp](#)
2. [O Markdown](#)
3. [Wyróżnienia tekstu](#)
4. [Nagłówki](#)
5. [Linki](#)
6. [Obrazki](#)
7. [Blok kodu](#)
8. [Tabele](#)

Poznaj Markdown

Markdown jest to język znaczników przeznaczony do formatowania tekstu. Powstał w celu maksymalnego uproszczenia tworzenia i formatowania tekstu. Markdown to format czysto tekstowy, można go łatwo konwertować do **HTML**. A także **pdf**, **epub** czy **mobi**.

Dzięki temu opracowaniu nauczysz się jak pisać dokumenty wykorzystując składnię markdown.



Obrazek 1: Przykład tekstu markdown wraz z wyrenderowanym dokumentem po prawej.

Krótkie wprowadzenie do Markdown

Język Markdown został stworzony w 2004 roku przez Johna Grubera z istotnym wkładem Aarona Swartza. Głównym celem było umożliwienie "tworzenia dokumentów za pomocą łatwego-w-czytaniu, łatwego-w-pisaniu czystego formatu tekstowego, z możliwością jego konwersji do strukturalnie poprawnego XHTML(lub HTML)".

Zapożyczając techniki z istniejących konwencji oznaczania czystego tekstu w wiadomościach e-mail, język został zaprojektowany aby być czytelnym takim jakim jest, aby nie wyglądał jakby został udekorowany znacznikami oraz instrukcjami formatującymi jak to się dzieje w HTML. Markdown posiada składnię formatującą, która może być z łatwością odczytywana przez ludzi, a także konwertowana do HTML.

Gruber napisał skrypt w Perlu - Markdown.pl, który konwertował oznaczony tekst do poprawnego i dobrze sformatowanego XHTML lub HTML. Skrypt może być używany samodzielnie, jako wtyczka dla Blosxom lub Movable Type, lub jako filtr tekstowy w BBEEdit.

Od tego czasu powstały napisane przez innych autorów reimplementacje Markdown, jako moduł Perla dostępny w CPAN (Text::Markdown), oraz w wielu innych językach programowania. Markdown jest rozprowadzany na licencji BSD i został dołączony lub jest dostępny jako wtyczka dla wielu systemów CMS.

Gdzie jest używany?

Markdown jest powszechnie stosowany w serwisach **GitHub**, **GitBook**, **Reddit**, **Diaspora**, **Stack Overflow**, **OpenStreetMap**, oraz wielu innych.

Nawet ta książka jest napisana przy użyciu Markdown: [Źródło tej strony](#).

Pliki

Dokument markdown jest plikiem tekstowym z rozszerzeniem `.md`. Plik markdown można otworzyć za pomocą każdego prostego edytora tekstowego.

Wyróżnienia tekstu

Pogrubienie i kursywa

Kod:

```
**pogrubienie** lub __pogrubienie__ (CTRL + B)

*kursywa* lub _kursywa_ (CTRL + I)

**Czasami potrzeba aby spory kawałek tekstu był pogrubiony.
Na przykład taki, _który będzie wyświetlany w kilku liniach_.
Naprawę, spory kawałek tekstu, całkiem spory.**
```

Efekt:

pogrubienie lub **pogrubienie** (CTRL + B)

kursywa lub *kursywa* (CTRL + I)

Czasami potrzeba aby spory kawałek tekstu był pogrubiony. Na przykład taki, *który będzie wyświetlany w kilku liniach*. Naprawę, spory kawałek tekstu, *całkiem* spory.

Cytowanie

Kod:

```
> Jak ci się podoba Markdown?
>> Nooo, fajny fajny :)

> To czytaj dalej ;)
```

Efekt:

Jak Ci się podoba Markdown?

Nooo, fajny fajny...

To czytaj dalej ;)

Nagłówki

Gdy piszemy dokument w markdown musimy mieć możliwość dodawania tytułów oraz nagłówków.

Markdown udostępnia dwa sposoby tworzenia nagłówków.

Pierwszy z nich to podkreślanie za pomocą znaków = (równa się) - dla nagłówków pierwszego poziomu oraz znaków - (myślnik) - dla nagłówków drugiego poziomu.

Kod:

```
To jest nagłówek H1
=====

To jest nagłówek H2
-----
```

Jakakolwiek ilość = lub - pod treścią nagłówka zadziała.

Wynik:

To jest nagłówek H1

To jest nagłówek H2

Drugi sposób to użycie od 1 do 6 znaków # (hash) na początku linii, odpowiadających nagłówkom poziomów 1 - 6.

Kod:

```
# To jest nagłówek H1

## To jest nagłówek H2

##### To jest nagłówek H6
```

Wynik:

To jest nagłówek H1

To jest nagłówek H2

To jest nagłówek H6

Opcjonalnie możesz zamykać znaczniki nagłówków. To tylko kosmetyka - możesz tak robić jeśli myślisz że będzie to wyglądało lepiej. Ilość zamykających znaków # nie musi odpowiadać ilości znaków użytych do *otwarcia* nagłówka.

```
# To jest nagłówek H1 #

## To jest nagłówek H2 ##
```

```
### To jest nagłówek H3 #####
```

Linki

Markdown posiada dwa style linków: w linii oraz referencyjne.

W obu stylach tekst linku jest oznaczony [nawiasami kwadratowymi].

Aby utworzyć link w linii, używamy zwykłych nawiasów natychmiast po nawiasie kwadratowym zawierającym tekst linku. W nawiasie umieszczamy adres URL na który ma wstazywać link oraz opcjonalnie tytuł linku w cudzysłowie. Na przykład:

```
[Jestem linkiem w linii](https://www.google.com)

[Jestem linkiem w linii z tytułem](https://www.google.com "Strona Google")
```

Linki referencyjne używają drugiej pary nawiasów kwadratowych, pomiędzy którymi umieszczamy etykietę służącą do identyfikacji linku:

```
To jest [przykładowy][id] link referencyjny.
```

Opcjonalnie można użyć spacji do rozdzielania par nawiasów kwadratowych:

```
To jest [przykładowy] [id] link referencyjny.
```

Następnie, gdziekolwiek w dokumencie definiujemy link referencyjny w nowej linii, jak poniżej:

```
[id]: http://example.com/ "Opcjonalny tytuł linku"
```

GitHub oraz **GitBook** obsługują autolinkowanie adresów URL. Jeśli chcesz wstawić link (zamiast ustawiać tekst linku) możesz po prostu wpisać URL i zostanie on automatycznie przetworzony w działający link.

Obrazki

```
# W linii
![Tekst zastępczy](/path/to/img.jpg "Opcjonalny tytuł")

# Referencyjne
![Tekst zastępczy][id]
[id]: url/to/image "Opcjonalny tytuł"
```

Jak widać, obrazki w markdown są bardzo podobne do linków. Różnice są takie:

- nawiasy kwadratowe muszą być poprzedzone wykrzyknikiem
- pomiędzy nawiasami kwadratowymi można umieścić tekst zastępczy. Opis obrazka, który zostanie wyświetlony jeśli obrazek nie będzie mógł być załadowany.

Bloki kodu

Preformatowane bloki kodu są używane do pisania o programowaniu lub oznaczania fragmentów kodu źródłowego. Zamiast tworzyć normalne akapity, linie w bloku kodu są interpretowane dosłownie.

Przykład:

```
To jest blok kodu
```

Aby utworzyć blok kodu, można po prostu użyć przynajmniej 4 spacji lub 1 tabulatora na początku linii.

Dla przykładu:

```
To będzie normalny akapit:
```

```
    A to blok kodu.
```

Można także utworzyć blok kodu za pomocą:

```
```
```

## Kod w linii

Kod w linii można utworzyć za pomocą: `

Przykład:

```
To jest `kod w linii`
```

## Podświetlanie składni

Można zdefiniować język dla podświetlania składni dodając jego nazwę na początku tagu otwierającego blok. Przykład:

```
```js
var a = {};
```
```

## Tabele

Tabele nie są częścią standardowej specyfikacji Markdown, ale należą do GFM (GitHub Markdown), więc można z nich korzystać na GitHub.

Przykładowa tabela - kod oraz wynik:

```
Tabele	są	fajne
kol 3 jest	wyrównana do prawej	$1600
kol 2 jest	wycentrowana	$12
wierzeż zebra	są fajne	$1
```

| Tabele        | są                  | fajne  |
|---------------|---------------------|--------|
| kol 3 jest    | wyrównana do prawej | \$1600 |
| kol 2 jest    | wycentrowana        | \$12   |
| wierzeż zebra | są fajne            | \$1    |

Dwukropki są używane do określenia wyrównania kolumn.

Zewnętrzne kreski (|) są opcjonalne, nie musisz także starać się wyrównywać wszystkich kolumn w wierszu. Możesz także używać formatowania markdown w komórkach tabeli.

Przykład:

```
Markdown	niezbyt	staranny
Nadał | `renderuje się` | **ładnie**
1 | 2 | 3
```

| Markdown     | niezbyt       | staranny |
|--------------|---------------|----------|
| <i>Nadał</i> | renderuje się | ładnie   |
| 1            | 2             | 3        |