

Instantly share code, notes, and snippets.



Jenkins / [Markdown-docs.md](#)

Last active yesterday

<> Code   Revisions 4   Stars 570   Forks 231



Download ZIP

Руководство по оформлению Markdown файлов

[Markdown-docs.md](#)

Настоящий документ предназначен для ознакомления пользователя с функциональными возможностями языка разметки Markdown. Markdown – это облегченный язык разметки, который является инструментом преобразования кода в HTML. Главной особенностью данного языка является максимально простой синтаксис, который служит для упрощения написания и чтения кода разметки, что, в свою очередь, позволяет легко его корректировать. Теперь рассмотрим более подробно функции языка разметки Markdown.

Markdown не является заменой HTML. Синтаксис Markdown достаточно ограничен, и соответствует лишь небольшому подмножеству элементов HTML. Он включает в себя следующие элементы:

#### 1. Блочные элементы

- [Параграфы и разрывы строк;](#)
- [Заголовки;](#)
- [Цитаты;](#)
- [Списки](#)
- [Блоки кода;](#)

- [Горизонтальные \(разделительные\) линии.](#)

## 2. Строчные элементы

- [Ссылки;](#)
- [Выделение текста;](#)
- [Кодовые фрагменты строк;](#)
- [Изображения.](#)

## 3. Дополнительные элементы

- [\[Обратный слеш\]\(#Backslash Escapes\);](#)
- [\[Автоматические ссылки\]\(#Automatic Links\);](#)
- [Специальные символы HTML.](#)

Более подробно с перечисленными функциями можно ознакомиться в разделе «Описание синтаксиса».

# ОПИСАНИЕ СИНТАКСИСА

---

## Блочные элементы

---

### Параграфы и разрывы строк

Для того, чтобы создать параграф с использованием синтаксиса языка Markdown, достаточно отделить строки текста одной (или более) пустой строкой (пустой считается всякая строка, которая не содержит в себе ничего, кроме пробелов и символов табуляции). Для того, чтобы вставить видимый перенос строки (элемент `<br/>`) необходимо окончить строку двумя пробелами и нажатием клавиши «Enter». Многие элементы синтаксиса Markdown выглядят и работают гораздо лучше в случае, когда их форматируют с помощью «жесткого перевода строк» (разделение строк, осуществленное самим пользователем, а не программой автоматически). К таким элементам относятся цитаты, списки и пр.

## Заголовки

Язык разметки Markdown поддерживает 2 стиля обозначения заголовков: подчеркивание и выделение символом («#»). Выделение заголовков с помощью подчеркивания производится знаками равенства («=») в случае, если заголовок первого уровня, и дефисами («-») в случае, если заголовок второго уровня. Количество знаков подчеркивания не ограничивается. При выделении заголовков с помощью символа («#») используется от одного до шести данных символов, которые устанавливаются в начале строки (перед заголовком). В данном случае количество символов соответствует уровню заголовка. Кроме того, заголовок возможно снабдить закрывающимися символами («#»), хотя это и не является обязательным. Количество закрывающихся символов не обязано соответствовать количеству начальных символов. Уровень заголовка определяется по количеству начальных символов.

Заголовки первого и второго уровней, выполненные с помощью подчеркивания, выглядят следующим образом:

```
Заголовок первого уровня
=====
Заголовок второго уровня
-----
```

Заголовки первого, третьего и шестого уровней, выполненные с помощью символа («#»), выглядят следующим образом:

```
# Заголовок первого уровня
### Заголовок третьего уровня
##### Заголовок шестого уровня
```

Приведенные выше заголовки, выполненные с помощью символа («#») тождественны следующим:

```
# Заголовок первого уровня #
### Заголовок третьего уровня ###
##### Заголовок шестого уровня #####
```

В результате на экран выводится следующее:

# Заголовок первого уровня

---

## Заголовок второго уровня

---

# Заголовок первого уровня

---

## Заголовок третьего уровня

### Заголовок шестого уровня

#### Цитаты

Для обозначения цитат в языке Markdown используется знак «больше» («>»). Его можно вставлять как перед каждой строкой цитаты, так и только перед первой строкой параграфа. Также синтаксис Markdown позволяет создавать вложенные цитаты (цитаты внутри цитат). Для их разметки используются дополнительные уровни знаков цитирования («>»). Цитаты в Markdown могут содержать всевозможные элементы разметки. Цитаты в языке Markdown выглядят следующим образом:

```
>Это пример цитаты,  
>в которой перед каждой строкой  
>ставится угловая скобка.
```

```
>Это пример цитаты,  
в которой угловая скобка  
ставится только перед началом нового параграфа.  
>Второй параграф.
```

Вложение цитаты в цитату выглядит следующим образом:

> Первый уровень цитирования  
>> Второй уровень цитирования  
>>> Третий уровень цитирования  
>  
>Первый уровень цитирования

В результате на экран выводится следующее:

Это пример цитаты, в которой перед каждой строкой ставится угловая скобка.

Это пример цитаты, в которой угловая скобка ставится только перед началом нового параграфа.

Второй параграф.

Вложенная цитата:

Первый уровень цитирования

Второй уровень цитирования

Третий уровень цитирования

Первый уровень цитирования

Уровень цитирования не может превышать 15-й.

**Списки**

Markdown поддерживает упорядоченные (нумерованные) и неупорядоченные (ненумерованные) списки. Для формирования неупорядоченных списков используются такие маркеры, как звездочки, плюсы и дефисы. Все перечисленные маркеры могут использоваться взаимозаменяемо. Для формирования упорядоченных списков в качестве маркеров используются числа с точкой. Важной особенностью в данном случае является то, что сами номера, с помощью которых формируется список, не важны, так как они не оказывают влияния на выходной HTML код. Как бы ни нумеровал пользователь список, на выходе он в любом случае будет иметь упорядоченный список, начинающийся с единицы (1, 2, 3...). Эту особенность стоит учитывать в том случае, когда необходимо использовать порядковые номера элементов в списке, чтобы они соответствовали номерам, получающимся в HTML. Упорядоченные списки всегда следует начинать с единицы. Маркеры списков обычно начинаются с начала строки, однако они могут быть сдвинуты, но не более чем на 3 пробела. За маркером должен следовать пробел, либо символ табуляции. При необходимости в список можно вставить цитату. В этом случае обозначения цитирования ( «>» ) нужно писать с отступом. Упорядоченные списки выглядят следующим образом:

1. Проводник
2. Полупроводник
3. Диэлектрик

Неупорядоченные списки выглядят следующим образом:

- \* Проводник
- \* Полупроводник
- \* Диэлектрик

Или

- Проводник
- Полупроводник
- Диэлектрик

Или

- + Проводник
- + Полупроводник
- + Диэлектрик

На выходе всех трех перечисленных вариантов имеется один и тот же результат. В результате на экран выводится следующее:

1. Проводник
2. Полупроводник
3. Диэлектрик

и

- Проводник
- Полупроводник
- Диэлектрик

Цитата, вставленная в список, выглядит следующим образом:

1. Элемент списка с цитатой:

- > Это цитата
- > внутри элемента списка.

2. Второй элемент списка

В результате на экран выводится следующее:

1. Элемент списка с цитатой:

Это цитата внутри элемента списка.

## 2. Второй элемент списка

При вставке цитат в элементы списка важно учитывать, что элементы списка должны находиться на одном уровне, а цитаты должны указываться с отступом. В случае, если правило с единым уровнем списка не соблюдается, следующий после цитаты элемент списка будет автоматически нумероваться цифрой «1.». Кроме того, при необходимости в список можно вставить исходный код. В этом случае его нужно писать с двойным отступом – 8 пробелов или 2 символа табуляции.

- Элемент списка, содержащий исходный код

```
<исходный код >
```

## Блоки кода

Отформатированные блоки кода используются в случае необходимости процитировать исходный код программ или разметки. Для создания блока кода в языке Markdown необходимо каждую строку параграфа начинать с отступа, состоящего из четырех пробелов или одного символа табуляции. Блок кода продолжается до тех пор, пока не встретится строка без отступа (или конец текста). Внутри блока кода амперсанды («&») и угловые скобки («<» и «>») автоматически преобразуются в элементы HTML разметки. Кроме того, следует отметить, что внутри блоков кода обычный синтаксис Markdown не обрабатывается. Блок кода в Markdown выглядит следующим образом:

Это обычный параграф:

```
Это блок кода
```

## Горизонтальные линии (разделители)



Для того чтобы создать горизонтальную линию с использованием синтаксиса языка Markdown, необходимо поместить три (или более) дефиса или звездочки на отдельной строке текста. Между ними возможно располагать пробелы. Горизонтальные линии в Markdown выглядят следующим образом:

```
Первая часть текста, который необходимо разделить
***
Вторая часть текста, который необходимо разделить
```

Или

```
Первая часть текста, который необходимо разделить
---
Вторая часть текста, который необходимо разделить
```

В результате на экран выводится следующее:

Первая часть текста, который необходимо разделить

Вторая часть текста, который необходимо разделить

При использовании данного инструмента важно помнить, что после первой части текста и перед второй необходимо оставлять пустую строку. Данное правило необходимо соблюдать только при использовании дефисов. Если его не соблюдать, на экран будет выведен заголовок второго уровня и строка обычного текста. При использовании символа звездочки данным правилом можно пренебречь.

## Строчные элементы

---

## Ссылки

Markdown поддерживает два стиля оформления ссылок:

- Гиперссылка, с немедленным указанием адреса (внутритекстовая);
- Гиперссылка, подобная сноске.

Подразумевается, что помимо URL-адреса существует еще текст ссылки. Он заключается в квадратные скобки. Для создания внутритекстовой гиперссылки необходимо использовать круглые скобки сразу после закрывающей квадратной. Внутри них необходимо поместить URL-адрес. В них же возможно расположить название, заключенное в кавычки, которое будет отображаться при наведении, но этот пункт не является обязательным.

```
[пример](http://example.com/ "Необязательная подсказка")
```

В результате на экран выводится следующее: [пример](http://example.com/) При ссылке на локальную директорию возможно использование относительного пути (от текущей страницы, сайта и т.п.)

При создании сносной гиперссылки вместо целевого адреса используется вторая пара квадратных скобок, внутри которых помещается метка, идентификатор ссылки (id).

```
[пример][id]:
```

Также, можно использовать пробел, чтобы отделять 2 пары квадратных скобок:

```
[пример] [id]:
```

В этом случае возможно определить идентификатор в любом месте документа:

```
[id]: http://example.com/ "Необязательная подсказка"
```

В результате на экран выводится следующее: [пример] [id] [id]: <http://example.com/> "Необязательная подсказка" Иными словами, она состоит из следующих элементов:

- Идентификатор ссылки, окружённый квадратными скобками (которым может предшествовать необязательный отступ от одного до трёх пробелов);
- Двоеточие;
- Один или несколько пробелов (или символов табуляции);
- URL гиперссылки;
- Необязательный заголовок (подсказка к изображению, которая всплывает при наведении на него) гиперссылки, заключённый либо в двойные или одиночные кавычки, либо в скобки.

Идентификаторы ссылок могут состоять из букв, цифр, пробелов и знаков пунктуации, однако они не чувствительны к регистру. То есть эти два варианта эквивалентны:

```
[ текст ссылки ][a]
[ текст ссылки ][A]
```

Markdown позволяет также использовать неявно выраженный идентификатор (сокращённый). В этом случае метка не приводится, вместо неё текст гиперссылки используется и в качестве её имени, а вторая пара квадратных скобок остаётся пустой. Например, чтобы сделать слово «Example» гиперссылкой, ведущей на сайт <http://example.com/>, достаточно написать:

```
[Example][ ]
```

и затем определить гиперссылку:

```
[Example]: http://example.com/
```

В результате на экран выводится следующее: [Example][] [Example]: <http://example.com/>

### Выделение текста

Markdown воспринимает звёздочки «\*» и символы подчёркивания «\_» как признаки смыслового выделения текста:

- Текст, окружённый одиночными «\*» или «\_», будет заключен в HTML-тэг `<em>` .
- Текст, окружённый двойными «\*» или «\_», будет заключен в HTML-тэг `<strong>` .

Иными словами, текст, окруженный одинарными символами, выделяется курсивным шрифтом, а текст, окруженный двойными символами, выделяется полужирным шрифтом. Также, выделенный фрагмент может находиться в любой части слова. Текст, выделенный курсивом с использованием синтаксиса языка Markdown, выглядит следующим образом:

*\*Пример\**

### *Пример*

Текст, выделенный полужирным шрифтом с использованием синтаксиса языка Markdown, выглядит следующим образом:

**\*\*Пример\*\***

### **Пример**

Текст, выделенный курсивным полужирным шрифтом с использованием синтаксиса языка Markdown выглядит следующим образом:

***\*\*\*Пример\*\*\****

### ***Пример***

Все приведенные выше примеры аналогичны следующим:

`_Пример_`

`__Пример__`

`Пере__распред__деление`

`___Пример___`

### Кодовые фрагменты строк

Чтобы отметить фрагмент строки, содержащий код, необходимо окружить его обратными апострофами «`». При использовании кодовых фрагментов строк текст будет отображаться в виде моноширинного шрифта. В отличие от блоков кода, кодовый фрагмент позволяет поместить код внутри обычного абзаца текста. Кодовый фрагмент строки в языке Markdown выглядит следующим образом:

Используйте оператор `if`

### Изображения

В Markdown существует 2 способа вставки изображений в документ:

а. С помощью непосредственного указания URL-адреса изображения. Синтаксис данной команды выглядит следующим образом:

```
![Альтернативный текст](/путь/к/изображению.jpg)
```

или

```
![Альтернативный текст](/путь/к/изображению.jpg "Подсказка")
```

Иными словами, он состоит из следующих элементов:

- восклицательный знак;
- квадратные скобки, в которых указывается альтернативный изображению текст (он станет содержимым атрибута в элементе `img`);
- круглые скобки, содержащие URL-адрес или относительный путь изображения, а также (необязательно) всплывающую подсказку, заключённую в двойные или одиночные кавычки.

b. С помощью метки-идентификатора. Синтаксис данной команды записывается следующим образом:

```
![Альтернативный текст][id]
```

где «`id`» — имя определённой метки изображения. Метки изображений определяются при помощи синтаксиса, совершенно идентичного меткам гиперссылок:

```
[id]: путь/к/изображению "Необязательная подсказка"
```

Важной особенностью является то, что Markdown не позволяет задать размеры изображения (ширину, высоту).

## Дополнительные элементы

---

### Обратный слеш

Может употребляться в Markdown перед специальными символами для того, чтобы они воспринимались в их буквальном (а не служебном) значении. Полный список данных символов приводится ниже:

«\» - слеш;

«`» - обратный апостроф;

«\*» - звездочка;

«\_» - символ подчеркивания;

«{}» - фигурные скобки;

«[]» - квадратные скобки;

«()» - круглые скобки;

«#» - символ решетки;

«+» - плюс;

«-» - минус (дефис);

«.» — точка;

«!» - восклицательный знак.

#### Автоматические ссылки

Markdown поддерживает упрощённый порядок автоматического создания ссылок для URL-адресов и адресов электронной почты. Для этого достаточно поместить URL-адрес или почтовый адрес в угловые скобки, и Markdown сделает его гиперссылкой. В отличие от вышеописанных стилей, в данном случае сам же URL-адрес или почтовый адрес становится и текстом гиперссылки. Автоматические ссылки на адреса электронной почты работают аналогично. Автоматические ссылки в языке Markdown выглядят следующим образом

```
<http://example.com/>
```

В результате на экран выводится следующее: <http://example.com/>

Автоматическая ссылка на адрес электронной почты в Markdown выглядит следующим образом

```
<address@example.com>
```

В результате на экран выводится следующее: [address@example.com](mailto:address@example.com)

### Специальные символы HTML

В языке HTML существует два символа, требующих специального рассмотрения: это символы («<» и «&»). Левая угловая скобка используется как начало тэга; амперсанды применяются для обозначения специального символа HTML. Для того чтобы использовать эти символы в их буквальном смысле, необходимо заменить их элементами HTML, а именно `&lt;` и `&amp;` соответственно. При использовании Markdown подобных действий совершать не нужно. Он позволяет использовать эти символы в исходном виде. В случае если амперсанд используется как часть спецсимвола HTML, он останется неизменным. В противном случае Markdown преобразует его в `&amp;` .

-----

copyright: <https://github.com/OlgaVlasova/markdown-doc/edit/master/README.md>



**S-inTheory** commented on 12 May 2020

у вас ошибка в названии: «Руководство по оформлениe Markdown файлов», а нужно «Руководство по оформлению Markdown файлов»



**Jenkins** commented on 12 May 2020

Author

у вас ошибка в названии: «Руководство по оформлениe Markdown файлов», а нужно «Руководство по оформлению Markdown файлов»

Спасибо, исправил