

# Базовый синтаксис письма и форматирования

Создавайте сложное форматирование для вашего текста и кода на GitHub с помощью простого синтаксиса.

## Кто может использовать эту функцию?

📁 Markdown можно использовать в веб-интерфейсе GitHub.

## В этой статье

Заголовки

Стилизация текста

Цитирование текста

Код цитирования

Поддерживаемые цветовые модели

Ссылки

Ссылки на разделы

Относительные ссылки

Индивидуальные якоря

Разрывы строк

Изображения

Списки

Списки задач

Упоминание людей и команд

Ссылки на проблемы и запросы на извлечение

Ссылки на внешние ресурсы

Загрузка активов



Использование эмодзи

Абзацы

Сноски

Оповещения

Скрытие контента с комментариями

Игнорирование форматирования Markdown

Отключение рендеринга Markdown

Дальнейшее чтение

---

## Заголовки

---

Чтобы создать заголовок, добавьте от одного до шести `#` символов перед его текстом. Количество `#` используемых символов определит уровень иерархии и размер шрифта заголовка.

```
# A first-level heading
## A second-level heading
### A third-level heading
```


# A first-level heading

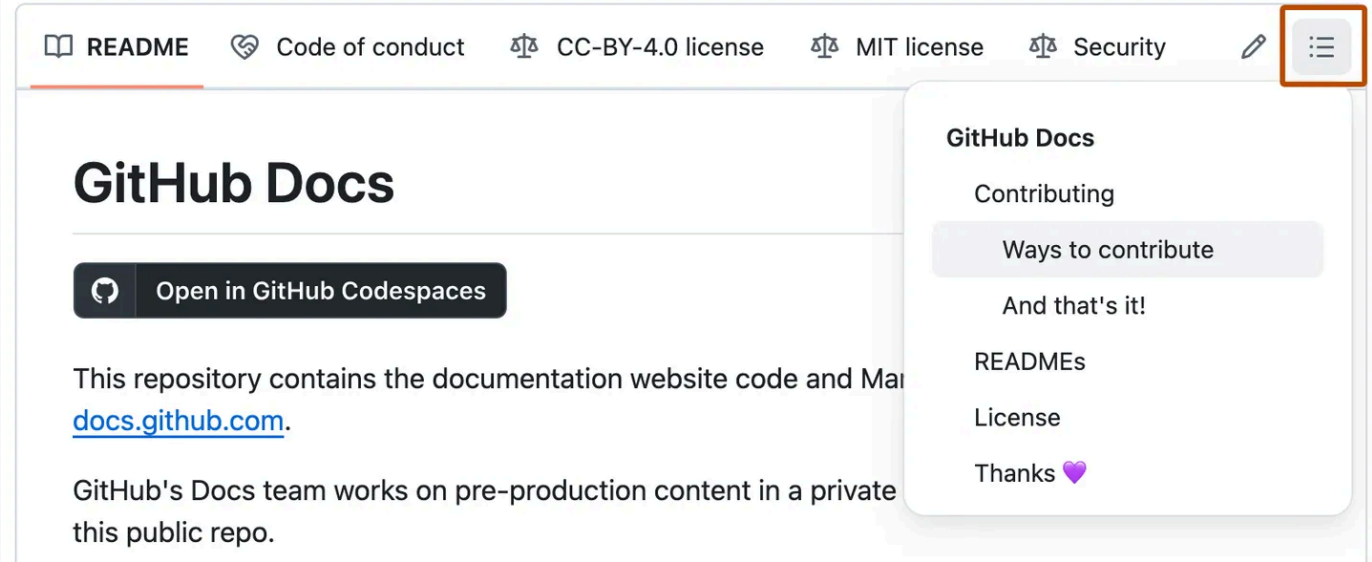
---

## A second-level heading

---

### A third-level heading

При использовании двух или более заголовков GitHub автоматически генерирует оглавление, к которому можно получить доступ, щёлкнув  по заголовку файла. Каждый заголовок указан в оглавлении, и вы можете щёлкнуть по нему, чтобы перейти к выбранному разделу.



## Стилизация текста

Вы можете выделить текст в полях комментариев и `.md` файлах с помощью жирного шрифта, курсива, зачеркивания, подстрочного или надстрочного шрифта.

| Стиль                         | Синтаксис                                 | Сочетание клавиш   | Пример  | Выход                                |
|-------------------------------|---|--|---|--------------------------------------|
| Смелый                        | <code>** **</code> или <code>__ __</code> | <code>Command + B</code> (Mac) или <code>Ctrl + B</code> (Windows/Linux) | <code>**This is bold text**</code>                  | <b>Это жирный текст</b>              |
| Курсив                        | <code>* *</code> или <code>_ _</code>     | <code>Command + I</code> (Mac) или <code>Ctrl + I</code> (Windows/Linux) | <code>_This text is italicized_</code>              | <i>Этот текст выделен курсивом.</i>  |
| Зачеркивание                  | <code>~~ ~~</code> или <code>~ ~</code>   | Никто  | <code>~~This was mistaken text~~</code>             | <del>Это был ошибочный текст.</del>  |
| Жирный и вложенный курсив     | <code>** **</code> и <code>_ _</code>     | Никто  | <code>**This text is _extremely_ important**</code> | <b>Этот текст чрезвычайно важен</b>  |
| Все жирным шрифтом и курсивом | <code>*** ***</code>                      | Никто  | <code>***All this text is important***</code>       | <b><i>Весь этот текст важен.</i></b> |

| Стиль              | Синтаксис                             | Сочетание клавиш | Пример   | Выход                              |
|--------------------|---------------------------------------|------------------|--|------------------------------------|
| Подстрочный индекс | <code>&lt;sub&gt; &lt;/sub&gt;</code> | Никто            | This is a <code>&lt;sub&gt;subscript&lt;/sub&gt;</code> text   | Это <sub>подстрочный</sub> текст   |
| Надстрочный индекс | <code>&lt;sup&gt; &lt;/sup&gt;</code> | Никто            | This is a <code>&lt;sup&gt;superscript&lt;/sup&gt;</code> text | Это <sup>надстрочный</sup> текст . |
| Подчеркнуть        | <code>&lt;ins&gt; &lt;/ins&gt;</code> | Никто            | This is an <code>&lt;ins&gt;underlined&lt;/ins&gt;</code> text | Это <u>подчеркнуты</u> й текст     |

## Цитирование текста

Вы можете цитировать текст с помощью `>` .

Text that is not a quote

> Text that is a quote

Цитируемый текст отступает вертикальной чертой слева и отображается серым шрифтом.

Text that is not a quote

Text that is a quote

📘 Примечание

При просмотре беседы вы можете автоматически цитировать текст комментария, выделив его и нажав `R` . Вы можете процитировать весь комментарий, нажав `...` , а затем **«Цитировать ответ»** . Подробнее о сочетаниях клавиш см. в [разделе Сочетания клавиш](#) .

## Код цитирования

Вы можете выделить код или команду внутри предложения, используя одинарные обратные кавычки. Текст внутри обратных кавычек не будет отформатирован. Вы также можете нажать сочетание клавиш `Command + E` (Mac) или `Ctrl + E` (Windows/Linux), чтобы вставить обратные кавычки для блока кода в строку Markdown.

```
Use `git status` to list all new or modified files that haven't yet been committed.
```

Use `git status` to list all new or modified files that haven't yet been committed.

Чтобы выделить код или текст в отдельный блок, используйте тройные обратные кавычки.

```
Some basic Git commands are:
```
git status
git add
git commit
```
```

Some basic Git commands are:

```
git status
git add
git commit
```

Дополнительные сведения см. в разделе [Создание и выделение блоков кода](#).

Если вы часто редактируете фрагменты кода и таблицы, вам может быть полезно включить моноширинный шрифт во всех полях комментариев на GitHub. Подробнее см. в статье «[О написании и форматировании на GitHub](#)».

## Поддерживаемые цветовые модели

В задачах, запросах на включение изменений и обсуждениях вы можете указывать цвета внутри предложения, используя обратные кавычки. Поддерживаемая цветовая модель внутри обратных кавычек будет отображать визуализацию цвета.

```
The background color is `#ffffff` for light mode and `#000000` for dark mode.
```

The background color is `#ffffff` for light mode and `#000000` for dark mode.

Вот поддерживаемые в настоящее время цветовые модели.

| Цвет                   | Синтаксис                 | Пример                            | Выход                           |
|------------------------|---------------------------|-----------------------------------|---------------------------------|
| ШЕСТИНАДЦАТЕРИЧНЫ<br>Й | <code>`#RRGGBB`</code>    | <code>`#0969DA`</code>            | <code>#0969DA</code>            |
| РГБ                    | <code>`rgb(R,G,B)`</code> | <code>`rgb(9, 105, 218)`</code>   | <code>rgb(9, 105, 218)</code>   |
| HSL                    | <code>`hsl(H,S,L)`</code> | <code>`hsl(212, 92%, 45%)`</code> | <code>hsl(212, 92%, 45%)</code> |

Примечание

- Поддерживаемая цветовая модель не может иметь начальных или конечных пробелов внутри обратных кавычек.
- Визуализация цвета поддерживается только в задачах, запросах на извлечение и обсуждениях.

## Ссылки

Вы можете создать встроенную ссылку, заключив текст ссылки в квадратные скобки `[ ]`, а затем URL-адрес в круглые скобки `( )`. Вы также можете использовать сочетание клавиш `Command + K` для создания ссылки. Выделив текст, вы можете вставить URL-адрес из буфера обмена, чтобы автоматически создать ссылку на основе выделенного текста.

Вы также можете создать гиперссылку Markdown, выделив текст и нажав сочетание клавиш `Command + V`. Если вы хотите заменить текст ссылкой, используйте сочетание клавиш `Command + Shift + V`.


This site was built using [GitHub Pages](https://pages.github.com/).

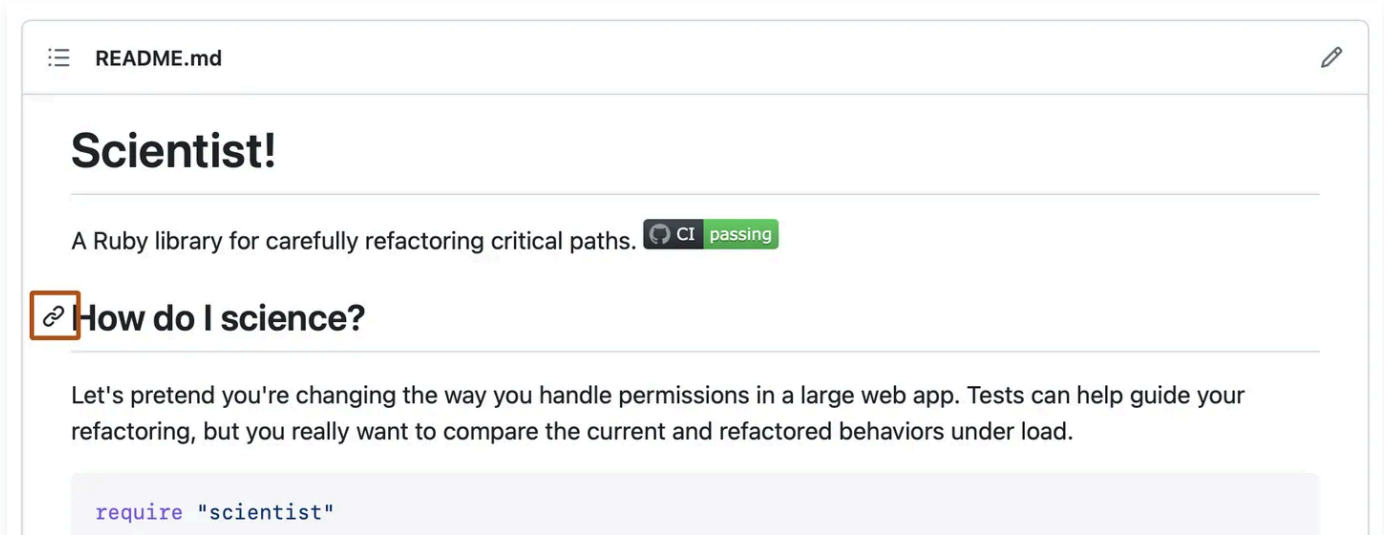
This site was built using [GitHub Pages](#).

Примечание

GitHub автоматически создаёт ссылки, когда в комментарии указаны корректные URL-адреса. Подробнее см. в разделе [«Автоссылки и URL-адреса»](#).

# Ссылки на разделы

Вы можете напрямую ссылаться на любой раздел с заголовком. Чтобы увидеть автоматически сгенерированный якорь в отрендеренном файле, наведите курсор на заголовок раздела, чтобы увидеть  значок, и щёлкните по нему, чтобы отобразить якорь в браузере.



Если вам необходимо определить якорь для заголовка в редактируемом файле, вы можете воспользоваться следующими основными правилами:

- Буквы преобразуются в строчные.
- Пробелы заменяются дефисами ( - ). Все остальные пробелы и знаки препинания удаляются.
- Начальные и конечные пробелы удаляются.
- Форматирование разметки удаляется, остается только содержимое (например, `_italics_` становится `italics` ).
- Если автоматически сгенерированный якорь для заголовка идентичен более раннему якорю в том же документе, уникальный идентификатор генерируется путем добавления дефиса и автоматически увеличивающегося целого числа.

Более подробную информацию о требованиях к фрагментам URI см. в [RFC 3986: Uniform Resource Identifier \(URI\): Generic Syntax, Section 3.5](#) .

В приведенном ниже блоке кода показаны основные правила, используемые для генерации якорей из заголовков в отображаемом контенте.

```
# Example headings
```

```
## Sample Section
```

```
## This'll be a Helpful Section About the Greek Letter Θ!
```

A heading containing characters not allowed in fragments, UTF-8 characters, two consecutive spaces between the first and second words, and formatting.

**## This heading is not unique in the file**

TEXT 1

**## This heading is not unique in the file**

TEXT 2

**# Links to the example headings above**

Link to the sample section: [Link Text](#sample-section).

Link to the helpful section: [Link Text](#thisll-be-a-helpful-section-about-the-greek-letter-θ).

Link to the first non-unique section: [Link Text](#this-heading-is-not-unique-in-the-file).

Link to the second non-unique section: [Link Text](#this-heading-is-not-unique-in-the-file-1).

### 📘 Примечание

Если вы редактируете заголовок или меняете порядок заголовков с «одинаковыми» анкерами, вам также потребуется обновить все ссылки на эти заголовки, поскольку анкоры изменятся.

## Относительные ссылки

Вы можете определить относительные ссылки и пути к изображениям в визуализированных файлах, чтобы помочь читателям перейти к другим файлам в вашем репозитории.

Относительная ссылка — это ссылка, которая указывается относительно текущего файла. Например, если в корне репозитория есть файл README, а в *docs/CONTRIBUTING.md* есть другой файл, относительная ссылка на *CONTRIBUTING.md* в файле README может выглядеть так:

```
[Contribution guidelines for this project](docs/CONTRIBUTING.md)
```

GitHub автоматически преобразует относительную ссылку или путь к изображению в зависимости от текущей ветки, чтобы ссылка или путь всегда были работоспособны. Путь к ссылке будет указан относительно текущего файла. Ссылки, начинающиеся с ,



/ будут указаны относительно корня репозитория. Вы можете использовать все операнды относительной ссылки, такие как `./` и `../`.

Текст ссылки должен быть в одну строку. Пример ниже не будет работать.

```
[Contribution  
guidelines for this project](docs/CONTRIBUTING.md)
```

Относительные ссылки удобнее для пользователей, клонирующих ваш репозиторий. Абсолютные ссылки могут не работать в клонах вашего репозитория. Мы рекомендуем использовать относительные ссылки для ссылки на другие файлы в вашем репозитории.

## Индивидуальные якоря

Вы можете использовать стандартные теги привязки HTML ( `<a name="unique-anchor-name"></a>` ) для создания точек привязки навигации в любом месте документа. Чтобы избежать неоднозначных ссылок, используйте уникальную схему именования тегов привязки, например, добавляя префикс к `name` значению атрибута.

### 📌 Примечание

Пользовательские якоря не будут включены в структуру документа/содержание.

Вы можете создать ссылку на пользовательский якорь, используя значение атрибута, `name` присвоенного якорю. Синтаксис точно такой же, как при ссылке на якорь, автоматически сгенерированный для заголовка.

Например:

```
# Section Heading
```

```
Some body text of this section.
```

```
<a name="my-custom-anchor-point"></a>
```

```
Some text I want to provide a direct link to, but which doesn't have its own  
heading.
```

```
(... more content...)
```

```
[A link to that custom anchor](#my-custom-anchor-point)
```

### 💡 Кончик

Пользовательские якоря не учитываются при автоматическом именовании и нумерации ссылок автоматических заголовков.

## Разрывы строк

Если вы пишете о проблемах, запросах на извлечение или обсуждениях в репозитории, GitHub автоматически отобразит перенос строки:

```
This example  
will span two lines
```

Однако, если вы пишете в MD-файле, приведенный выше пример будет отображаться в одну строку без переноса. Чтобы создать перенос строки в MD-файле, необходимо включить один из следующих параметров:

- В конце первой строки поставьте два пробела.

```
Этот пример  
Будет охватывать две строки
```

- В конце первой строки поставьте обратную косую черту.

```
This example\  
will span two lines
```

- Добавьте HTML-тег разрыва строки в конце первой строки.

```
This example<br/>  
will span two lines
```

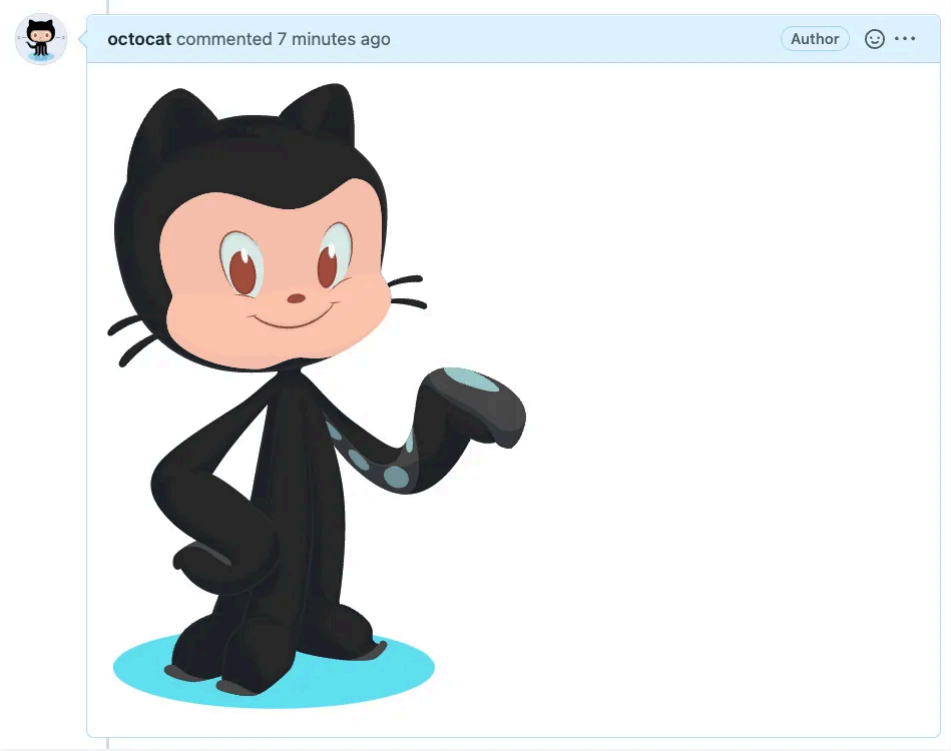
Если вы оставите пустую строку между двумя строками, как файлы .md, так и Markdown в задачах, запросах на извлечение и обсуждениях отобразят две строки, разделенные пустой строкой:

```
This example  
  
will have a blank line separating both lines
```

## Изображения

Вы можете отобразить изображение, добавив `!` и заключив альтернативный текст в скобки `[ ]`. Альтернативный текст — это краткий текстовый эквивалент информации на изображении. Затем заключите ссылку на изображение в скобки `( )`.

```
![Screenshot of a comment on a GitHub issue showing an image, added in the
Markdown, of an Octocat smiling and raising a tentacle.]
(https://myoctocat.com/assets/images/base-octocat.svg)
```



GitHub поддерживает встраивание изображений в ваши задачи, запросы на включение изменений, обсуждения, комментарии и `.md` файлы. Вы можете отобразить изображение из своего репозитория, добавить ссылку на изображение в интернете или загрузить изображение. Подробнее см. в [разделе Загрузка ресурсов](#).

**Примечание**

Если вы хотите отобразить изображение, находящееся в вашем репозитории, используйте относительные ссылки вместо абсолютных.

Вот несколько примеров использования относительных ссылок для отображения изображения.

| Контекст                                 | Относительная ссылка                                 |
|--|--|
| В <code>.md</code> файле на той же ветке | <code>/assets/images/electrocat.png</code>           |
| В <code>.md</code> файле на другой ветке | <code>../../main/assets/images/electrocat.png</code> |

| Контекст  | Относительная ссылка   |
|---|--|
| В проблемах, запросах на включение изменений и комментариях репозитория | <code>../blob/main/assets/images/electrocat.png?raw=true</code>                      |
| В <code>.md</code> файле в другом репозитории                           | <code>/../../../../github/docs/blob/main/assets/images/electrocat.png</code>         |
| В проблемах, запросах на извлечение и комментариях другого репозитория  | <code>../../../../github/docs/blob/main/assets/images/electrocat.png?raw=true</code> |

### 📘 Примечание

Последние две относительные ссылки в таблице выше будут работать для изображений в частном репозитории только в том случае, если у зрителя есть как минимум права на чтение к частному репозиторию, содержащему эти изображения.

Более подробную информацию см. в разделе [Относительные ссылки](#).

## Элемент «Изображение»

HTML `<picture>` -элемент поддерживается.

## Списки

Вы можете создать неупорядоченный список, поставив перед одной или несколькими строками текста `-`, `*`, или `+`.

```
- George Washington
* John Adams
+ Thomas Jefferson
```

- George Washington
- John Adams
- Thomas Jefferson

Чтобы упорядочить список, поставьте перед каждой строкой номер.

1. James Madison
2. James Monroe
3. John Quincy Adams

1. James Madison
2. James Monroe
3. John Quincy Adams

## Вложенные списки

Вы можете создать вложенный список, разместив один или несколько элементов списка под другим элементом.

Чтобы создать вложенный список с помощью веб-редактора на GitHub или текстового редактора с моноширинным шрифтом, например, [Visual Studio Code](#), вы можете визуально выравнивать список. Вводите пробелы перед элементом вложенного списка, пока символ маркера списка ( `-` или `*` ) не окажется непосредственно под первым символом текста в элементе, расположенном выше.

- ```
1. First list item
  - First nested list item
  - Second nested list item
```

### 📘 Примечание

В веб-редакторе вы можете увеличить или уменьшить отступ одной или нескольких строк текста, сначала выделив нужные строки, а затем используя `Tab` или `Shift + Tab` соответственно.

- ```
1. First list item
  - First nested list item
  - Second nested list item
```

1. First list item
  - First nested list item
    - Second nested list item

Чтобы создать вложенный список в редакторе комментариев на GitHub, где не используется моноширинный шрифт, вы можете посмотреть на элемент списка, расположенный непосредственно над вложенным списком, и подсчитать количество

символов перед его содержимым. Затем введите это количество пробелов перед элементом вложенного списка.

В этом примере вы можете добавить вложенный элемент списка под элементом списка, 100. First list item сделав отступ вложенного элемента списка минимум в пять пробелов, поскольку 100. перед находится пять символов ( ) First list item .

```
100. First list item
    - First nested list item
```

## 100. First list item

- First nested list item

Используя один и тот же метод, можно создавать несколько уровней вложенности списков. Например, поскольку первый элемент вложенного списка содержит семь символов ( 100. ) перед содержимым вложенного списка First nested list item , вам потребуется добавить отступ для второго элемента вложенного списка как минимум на два символа (минимум девять пробелов).

```
100. First list item
    - First nested list item
      - Second nested list item
```

## 100. First list item

- First nested list item
  - Second nested list item

Дополнительные примеры см. в [спецификации GitHub Flavored Markdown Spec](#) .

## Списки задач

Чтобы создать список задач, добавьте к каждому пункту списка дефис и пробел, а затем [ ] . Чтобы отметить задачу как выполненную, используйте [x] .

```
- [x] #739
- [ ] https://github.com/octo-org/octo-repo/issues/740
- [ ] Add delight to the experience when all tasks are complete :tada:
```

- ☒ ☒ **Convert text into issues** #739
- ☐ ☒ **Keep issue state and checkboxes in sync** #740
- ☐ Add delight to the experience when all tasks are complete 🎉

Если описание элемента списка задач начинается со скобок, вам необходимо экранировать их с помощью `\`:

- [ ] \ (Optional) Open a followup issue

Более подробную информацию см. в [разделе О списках задач](#).

## Упоминание людей и команд

Вы можете упомянуть человека или [команду](#) на GitHub, добавив `@` к нему имя пользователя или название команды. Это вызовет уведомление и привлечёт внимание участников обсуждения. Пользователи также получают уведомление, если вы отредактируете комментарий, упомянув имя пользователя или название команды. Подробнее об уведомлениях см. в разделе «[Об уведомлениях](#)».

### 📘 Примечание

Человек будет уведомлен об упоминании только в том случае, если у него есть доступ на чтение к репозиторию и, если репозиторий принадлежит организации, человек является членом этой организации.

@github/support What do you think about these updates?

**@github/support** What do you think about these updates?

Когда вы упоминаете родительскую команду, участники её дочерних команд также получают уведомления, что упрощает взаимодействие с несколькими группами людей. Подробнее см. в разделе «[О командах](#)».

Ввод `@` символа откроет список людей или команд, работающих над проектом. Список фильтруется по мере ввода, поэтому, найдя имя нужного человека или команды, вы можете использовать клавиши со стрелками, чтобы выбрать его, и нажать клавишу Tab или Enter, чтобы завершить ввод имени. Для команд введите @organization/team-name, и все участники этой команды будут подписаны на обсуждение.

Результаты автозаполнения доступны только соавторам репозитория и любым другим участникам обсуждения.

## Ссылки на проблемы и запросы на извлечение

Вы можете открыть список предложенных проблем и запросов на включение изменений в репозитории, нажав `[]` `#`. Введите номер или название проблемы или запроса на включение изменений, чтобы отфильтровать список, а затем нажмите клавишу **Tab** или **Enter**, чтобы завершить выбор выделенного результата.

Более подробную информацию см. в разделе [Автоматически связанные ссылки и URL-адреса](#).

## Ссылки на внешние ресурсы

Если для репозитория настроены пользовательские автоссылки, то ссылки на внешние ресурсы, такие как задачи JIRA или тикеты Zendesk, преобразуются в сокращённые ссылки. Чтобы узнать, какие автоссылки доступны в вашем репозитории, обратитесь к администратору репозитория. Подробнее см. в разделе «[Настройка автоссылок для ссылок на внешние ресурсы](#)».

## Загрузка активов

Вы можете загружать ресурсы, например изображения, перетаскивая их, выбирая в браузере файлов или вставляя. Вы можете загружать ресурсы в задачи, запросы на извлечение, комментарии и `.md` файлы в вашем репозитории.

## Использование эмодзи

Вы можете добавить эмодзи в свой текст, набрав `:EMOJICODE:`, двоеточие и название эмодзи.

@octocat :+1: This PR looks great - it's ready to merge! :shipit:

@octocat 🍏 This PR looks great - it's ready to merge! 🧑🏻

При вводе текста `:` появится список предлагаемых эмодзи. Список фильтруется по мере ввода, поэтому, как только вы найдёте нужный эмодзи, нажмите **Tab** или **Enter**, чтобы завершить подсвеченный результат.

Полный список доступных эмодзи и кодов смотрите [в Emoji-Cheat-Sheet](#).



## Абзацы

Вы можете создать новый абзац, оставив пустую строку между строками текста.

## Сноски

Вы можете добавлять сноски к своему контенту, используя следующий синтаксис скобок:

```
Here is a simple footnote[1]
```



Документы GitHub

Версии: бесплатная, профессиональная и командная ▾



Начать / Написание на GitHub / Начните писать на GitHub /

Базовый синтаксис форматирования

Сноска будет выглядеть так:

Here is a simple footnote<sup>[1]</sup>.

A footnote can also have multiple lines<sup>[2]</sup>.

1. My reference. ↩

2. To add line breaks within a footnote, prefix new lines with 2 spaces.

This is a second line. ↩

### 📘 Примечание

Положение сноски в Markdown не влияет на её отображение. Вы можете написать сноску сразу после ссылки на неё, и она всё равно будет отображаться внизу Markdown. Сноски не поддерживаются в вики.

## Оповещения

Оповещения — это расширение Markdown, основанное на синтаксисе цитирования, которое можно использовать для выделения важной информации. На GitHub они отображаются с помощью особых цветов и значков, указывающих на важность содержимого.

Используйте оповещения только тогда, когда они критически важны для успеха пользователя, и ограничьте их одним-двумя на статью, чтобы не перегружать читателя. Кроме того, следует избегать размещения оповещений подряд. Оповещения нельзя вкладывать в другие элементы.

Чтобы добавить оповещение, используйте специальную строку цитаты, указав тип оповещения, а затем добавьте информацию о нём в стандартной цитате. Доступно пять типов оповещений:

```
> [!NOTE]
> Useful information that users should know, even when skimming content.

> [!TIP]
> Helpful advice for doing things better or more easily.

> [!IMPORTANT]
> Key information users need to know to achieve their goal.

> [!WARNING]
> Urgent info that needs immediate user attention to avoid problems.

> [!CAUTION]
> Advises about risks or negative outcomes of certain actions.
```

Вот отображаемые оповещения:

#### **Note**

Useful information that users should know, even when skimming content.

#### **Tip**

Helpful advice for doing things better or more easily.

#### **Important**

Key information users need to know to achieve their goal.

#### **Warning**

Urgent info that needs immediate user attention to avoid problems.

#### **Caution**

Advises about risks or negative outcomes of certain actions.

## Скрытие контента с комментариями

Вы можете указать GitHub скрыть содержимое при отображении Markdown, поместив его в HTML-комментарий.

```
<!-- This content will not appear in the rendered Markdown -->
```

## Игнорирование форматирования Markdown

Вы можете указать GitHub игнорировать (или избегать) форматирование Markdown, используя `\` перед символом Markdown.

```
Let's rename \*our-new-project\* to \*our-old-project\*.
```

```
Let's rename *our-new-project* to *our-old-project*.
```

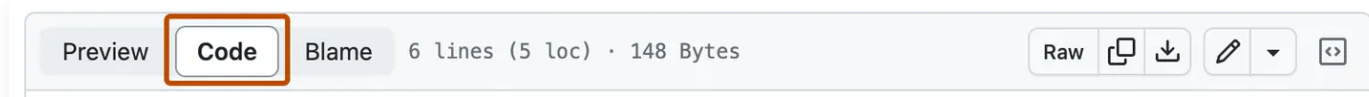
[Дополнительную информацию об обратных косых чертах см. в разделе Синтаксис разметки](#) Daring Fireball .

### 📘 Примечание

Форматирование Markdown не будет игнорироваться в заголовке проблемы или запросе на извлечение.

## Отключение рендеринга Markdown

При просмотре файла Markdown вы можете нажать **«Код»** в верхней части файла, чтобы отключить рендеринг Markdown и вместо этого просмотреть исходный код файла.



Отключение рендеринга Markdown позволяет использовать функции просмотра исходного кода, такие как связывание строк, что невозможно при просмотре обработанных файлов Markdown.

## Дальнейшее чтение

- [Спецификация Markdown с GitHub Flavored](#)
- [О написании и форматировании на GitHub](#)
- [Работа с расширенным форматированием](#)
- [Быстрый старт для написания на GitHub](#)

## Юридический

© 2025 GitHub, Inc. [Условия](#) [Конфиденциальность](#) [Статус](#) [Цены](#) [Экспертные услуги](#)  
[Блог](#)