# **Игнорирование файлов в Git**

Часто бывает так, что в папке-репозитории есть файлы, для которых не нужно хранить историю изменений. Например:

* macOS иногда создаёт скрытый файл .DS\_Store для хранения настроек папки. К вашему проекту он, скорее всего, никакого отношения не имеет.
* В Git не принято коммитить результаты компиляции исходного кода, то есть получившиеся исполняемые файлы.
* Среды разработки (вроде IntelliJ IDEA) могут создавать папку с вашими личными настройками проекта. Если добавить её в репозиторий, то среда разработки других участников проекта может загрузить ваши настройки и начать вести себя странно.

Чтобы Git игнорировал такие файлы и не пытался добавить их в репозиторий, нужно создать файл .gitignore (от англ. *ignore* — «игнорировать») и записать в него названия игнорируемых файлов. В этом уроке разберём, как это сделать.

## **Как заполнить .gitignore**

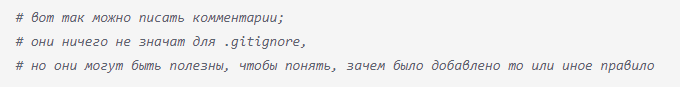
С точки зрения Git .gitignore — это обычный текстовый файл. Его добавляют в корень репозитория и тоже коммитят.

В простейшем случае в .gitignore указывают все файлы, которые нужно игнорировать (по одному имени на строку). Но часто удобнее использовать **шаблоны**. Шаблон, или правило, — это способ указать сразу на несколько файлов с однотипными названиями.

Разберём подробнее формат файла .gitignore, какие в нём могут встречаться строки и как выглядят шаблоны.

### **Комментарий**

Если строка начинается с #, то это комментарий, и .gitignore не будет его учитывать.



### **Просто название файла**

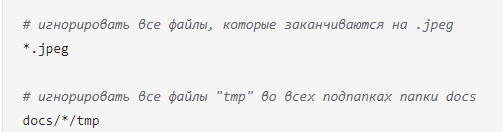
Допустим, нужно, чтобы Git игнорировал все файлы .DS\_Store. Для этого достаточно добавить в .gitignore строку с названием файла.



В таком случае Git будет игнорировать файлы с именем .DS\_Store, причём не только в корне репозитория, но и во всех вложенных папках.

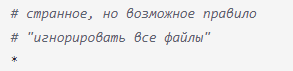
### **Звёздочка (\*)**

Символ звёздочки (\*) соответствует любой строке, включая пустую. Если такой символ используется в шаблоне в .gitignore, значит, файл будет проигнорирован вне зависимости от того, что будет на месте звёздочки.



Теперь Git будет игнорировать все файлы, которые заканчиваются на .jpeg — пригодится тем, кто не любит картинки. А также все временные файлы tmp (от англ. ***t****e****mp****orary* — «временный») в подпапках папки docs. Например, Git проигнорирует файл docs/current/tmp.

Если задать правило, которое состоит только из звёздочки, Git будет игнорировать все файлы. Это происходит потому, что под звёздочку подходит любое имя файла.



### **Вопросительный знак (?)**

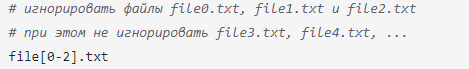
Вопросительный знак ? соответствует одному любому символу.



Если сохранить такую запись в .gitignore, то будут проигнорированы, например, файлы fileA.txt и file1.txt. А вот файл file12.txt не будет проигнорирован, потому что в его названии два символа после file, а не один.

### **Квадратные скобки ([…])**

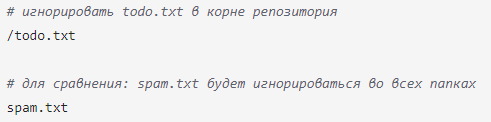
Квадратные скобки, как и вопросительный знак, соответствуют одному символу. При этом символ не любой, а только из списка, который указан в скобках.



В скобках можно либо перечислить символы ([abc]), либо задать диапазон ([a-z]).

### **Слеш (/)**

Косая черта, или слеш (/), указывает на каталоги. Если шаблон в .gitignore начинается со слеша, то Git проигнорирует файлы или каталоги только в корневой директории.



Теперь файл todo.txt в корневом каталоге будет проигнорирован. При этом, например, файл subdir/todo.txt по-прежнему отслеживается.

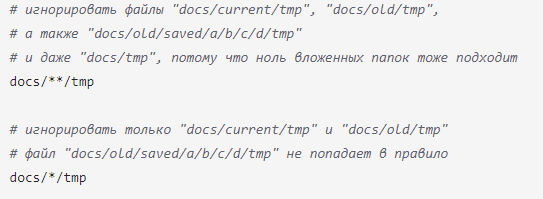
Если шаблон заканчивается слешем, то правило применится только к папке.



Обратите внимание: если build — это папка, то она будет проигнорирована. Если build — обычный файл, то он не подпадёт под правило и не будет игнорироваться.

### **Парные звёздочки (\*\*)**

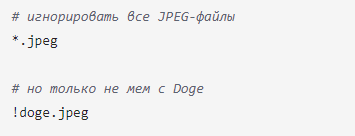
Функция парных звёздочек (\*\*) похожа на функцию одинарной (\*). Отличие в том, как они работают с вложенными папками. Двойная звёздочка может соответствовать любому количеству таких папок (в том числе нулю). Одинарная может соответствовать только одной.



Для двойной звёздочки верно то же самое, что и для одной: если задать правило \*\*, то будут проигнорированы все файлы.

### **Восклицательный знак (!)**

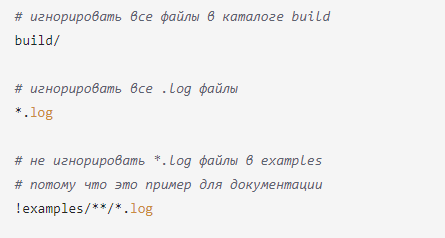
Любое правило в файле .gitignore можно инвертировать с помощью восклицательного знака (!).



Теперь файл doge.jpeg будет отслеживаться, хотя остальные jpeg-файлы будут проигнорированы. Такие правила удобны для добавления исключений из других правил .gitignore.c

### **Пример файла .gitignore**

Содержание .gitignore может быть таким.



## **.gitignore и git status**

Игнорируемые файлы не отображаются в выводе команды git status, иначе они бы засоряли вывод.

Если всё же нужно отобразить все игнорируемые файлы, то это можно сделать с помощью ключа --ignored: git status --ignored. В таком случае в выводе git status появится раздел Ignored files.

На скриншотах ниже видно, как меняется вывод команды git status без ключа и с ним.

Обратите внимание: сам файл .gitignore не отображается в выводе только потому, что мы его предварительно закоммитили.

