

1 -e Шаблон

Grep позволяет указать несколько шаблонов с помощью -e:

```
$ grep -e [pattern1] -e [pattern2] -e [pattern3]...[file]
```

Пример:

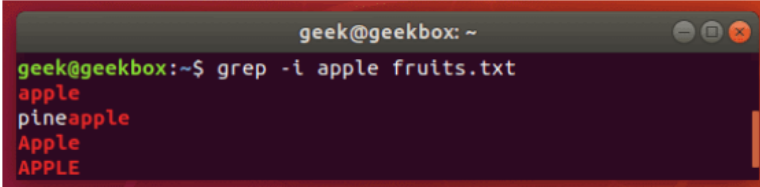
```
$ grep -e is -e It -e to text_file.txt
This is a sample text file. It contains
several lines to be used as part of testing grep
It supports numbers like 1, 2, 3 etc. as well as
This is a sample text file. It's repeated two times.
$
```

2 -i Игнорирует различия регистра.

ИГНОРИРОВАТЬ РЕГИСТР

По умолчанию, grep чувствительна к регистру, что немного мешает поиску, так как мы должны точно указать регистр букв при поиске. Эта проблема решается путем добавления ключа i к команде поиска.

```
$ grep -i string filename
```



```
geek@geekbox: ~
geek@geekbox:~$ grep -i apple fruits.txt
apple
pineapple
Apple
APPLE
```

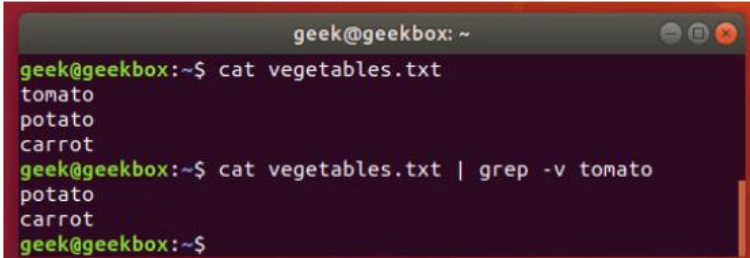
Как видно из скриншота, мы вывели слова независимо от того какими буквами они набраны.

3 -v Инвертирует смысл поиска соответствий.

ИСКЛЮЧИТЬ ШАБЛОН

Чтобы просмотреть содержание файла с исключением конкретной строки к запросу нужно добавить ключ v.

```
$ grep -v string-to-exclude filename
```



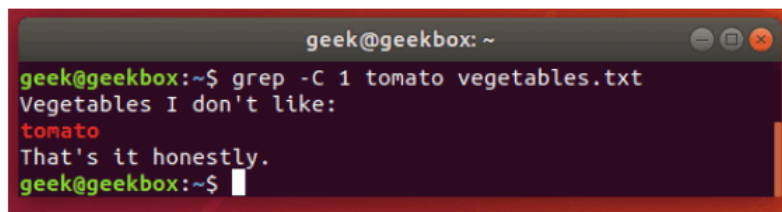
```
geek@geekbox: ~
geek@geekbox:~$ cat vegetables.txt
tomato
potato
carrot
geek@geekbox:~$ cat vegetables.txt | grep -v tomato
potato
carrot
geek@geekbox:~$
```

4 -c Выводит только количество совпадающих строк.

ПОКАЗАТЬ СТРОКИ ДО И ПОСЛЕ

Если нужно узнать в каком контексте употреблялась искомая фраза, можно вывести предложения предшествующее и следующее за указанной строкой. Для этого нужно запустить команду `grep` с ключом `-C` и передать число строк для отображения.

```
$ grep -c 1 string filename
```



```
geek@geekbox: ~  
geek@geekbox:~$ grep -C 1 tomato vegetables.txt  
Vegetables I don't like:  
tomato  
That's it honestly.  
geek@geekbox:~$
```

5 -l Выводит только совпадающие файлы.

Чтобы напечатать только имя файлов, в которых найден шаблон, используйте флаг `-l`:

```
$ grep -l [pattern] [file]
```

Вывод:

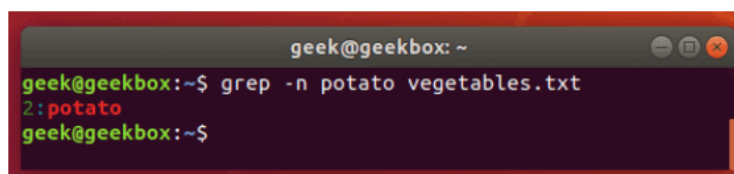
```
$ grep -l su *.txt  
file.txt  
text_file.txt  
$
```

6 -n Предваряет каждую строку вывода номером строки из файла ввода.

ВЫВОД НОМЕРА СТРОКИ

Чтобы показать номер строки, где была найдена искомая строка, команду `grep` следует запускать с ключом `-n`.

```
$ grep -n string filename
```



```
geek@geekbox: ~  
geek@geekbox:~$ grep -n potato vegetables.txt  
2:potato  
geek@geekbox:~$
```

- 7 -h Выводит совпадающие строки, не предваряя их именами файлов.

```
[deadline@fedora test_files]$ grep end *.txt
1.txt: end... 123
1.txt: end.
2.txt: end.
3.txt: end.
4.txt: end.
[deadline@fedora test_files]$ grep end *.txt -h
end... 123
end.
end.
end.
end.
```

- 8 -s Подавляет сообщения об ошибках о несуществующих или нечитаемых файлах.

```
[deadline@fedora test_files]$ grep -e 1 0.txt
grep: 0.txt: No such file or directory
[deadline@fedora test_files]$ grep -e 1 0.txt -s
[deadline@fedora test_files]$
```

- 9 -f file Получает регулярные выражения из файла.

Если у вас есть некий список часто используемых шаблонов, укажите его в файле и используйте флаг -f. Файл должен содержать по одному шаблону на строку.

```
$ grep -f [pattern_file] [file_to_match]
```

В примере мы создали файл шаблонов pattern.txt с таким содержанием:

```
$ cat pattern.txt
This
It
$
```

Чтобы это использовать, применяйте ключ -f:

```
$ grep -f pattern.txt text_file.txt
This is a sample text file. It contains
It supports numbers like 1, 2, 3 etc. as well as
This is a sample text file. It's repeated two times.
$
```

10 -o Печатает только совпадающие (непустые) части совпавшей строки.

```
[deadline@fedora test_files]$ grep end *.txt
1.txt: end... 123
1.txt: end.
2.txt: end.
3.txt: end.
4.txt: end.
[deadline@fedora test_files]$ grep end *.txt -o
1.txt: end
1.txt: end
2.txt: end
3.txt: end
4.txt: end
```