

# Linux Regular Expression Tutorial: Grep Regex Example

By Mary Brent ⌚ Updated October 7, 2021

## What are Linux Regular Expressions?

**Linux Regular Expressions** are special characters which help search data and matching complex patterns. Regular expressions are shortened as 'regexp' or 'regex'. They are used in many Linux programs like grep, bash, rename, sed, etc.

## Types of Regular expressions

For ease of understanding let us learn the different types of Regex one by one.

- [Basic Regular expressions](#)
- [Interval Regular expressions](#)
- [Extended regular expressions](#)
- [Summary](#)

Regular Expressions - Linux Tutorial 10



## Basic Regular expressions

Some of the commonly used commands with Regular expressions are tr, sed, vi and grep. Listed below are some of the basic Regex.

Symbol	Descriptions
.	replaces any character
^	matches start of string
\$	matches end of string
*	matches up zero or more times the preceding character
\	Represent special characters
()	Groups regular expressions
?	Matches up exactly one character

Let's see an example.

Execute cat sample to see contents of an existing file

```
guru99@guru99-VirtualBox:~$ cat sample
apple
bat
ball
ant
eat
pant
people
taste
guru99@guru99-VirtualBox:~$
```

Search for content containing letter 'a'.

```
guru99@guru99-VirtualBox:~$ cat sample | grep a
apple
bat
ball
ant
eat
pant
taste
guru99@guru99-VirtualBox:~$
```

‘^’ matches the start of a string. Let’s search for content that STARTS with a

```
guru99@guru99-VirtualBox:~$ cat sample | grep ^a
apple
ant
guru99@guru99-VirtualBox:~$
```

Only lines that start with character are filtered. Lines which do not contain the character ‘a’ at the start are ignored.

Let’s look into another example –

```
guru99@guru99-VirtualBox:~$ cat sample | grep t
bat
ant
eat
pant
taste
```

Select only those lines that end with t using \$

```
guru99@guru99-VirtualBox:~$ cat sample | grep t$
bat
ant
eat
pant
guru99@guru99-VirtualBox:~$
```

## Interval Regular expressions

are

Expression	Description
{n}	Matches the preceding character appearing 'n' times exactly
{n,m}	Matches the preceding character appearing 'n' times but not more than m
{n, }	Matches the preceding character only when it appears 'n' times or more

Example:

Filter out all lines that contain character 'p'

```
guru99@guru99-VirtualBox:~$ cat sample|grep p
apple
pant
people
```

We want to check that the character 'p' appears exactly 2 times in a string one after the other. For this the syntax would be:

```
cat sample | grep -E p\{2}
```

```
guru99@guru99-VirtualBox:~$ cat sample|grep -E p\{2}
apple
guru99@guru99-VirtualBox:~$
```

Note: You need to add -E with these regular expressions.

## Extended regular expressions

These regular expressions contain combinations of more than one expression. Some of them are:

Expression	Description
------------	-------------

`\?` Matches zero or one occurrence of the previous character

Example:

Searching for all characters 't'

```
guru99@guru99-VirtualBox:~$ cat sample|grep t
bat
ant
eat
pant
taste
```

Suppose we want to filter out lines where character 'a' precedes character 't'

We can use command like

```
cat sample|grep "a\+t"
```

```
guru99@guru99-VirtualBox:~$ cat sample|grep "a\+t"
bat
eat
guru99@guru99-VirtualBox:~$
```

## Brace expansion

The syntax for brace expansion is either a sequence or a comma separated list of items inside curly braces "{}". The starting and ending items in a sequence are separated by two periods "...".

Some examples:

```
guru99@VirtualBox:~$ echo {aa,bb,cc,dd}
aa bb cc dd
guru99@VirtualBox:~$ echo {0..11}
0 1 2 3 4 5 6 7 8 9 10 11
guru99@VirtualBox:~$ echo {a..z}
a b c d e f g h i j k l m n o p q r s t u v w x y z
guru99@VirtualBox:~$ echo a{0..9}b
a0b a1b a2b a3b a4b a5b a6b a7b a8b a9b
```

---

In the above examples, the echo command creates strings using the brace expansion.

## Summary:

- Regular expressions are a set of characters used to check patterns in strings
- They are also called 'regexp' and 'regex'
- It is important to learn regular expressions for writing scripts
- Some basic regular expressions are:

Symbol	Descriptions
.	replaces any character
^	matches start of string
\$	matches end of string

- Some extended regular expressions are:

Expression	Description
\+	Matches one or more occurrence of the previous character
\?	Matches zero or one occurrence of the previous character

- Some interval regular expressions are:

Expression	Description
{n}	Matches the preceding character appearing 'n' times exactly
{n,m}	Matches the preceding character appearing 'n' times but not more than m

- The brace expansion is used to generate strings. It helps in creating multiple strings out of one.

### Guru99 is Sponsored by Acunetix

Acunetix, the developers of dead-accurate web application security scanners have sponsored the Guru99 project to help scan for over 4500 web vulnerabilities accurately and at top speed.

VISIT THE ACUNETIX WEBSITE

← Prev

[Report a Bug](#)

Next →



About

[About Us](#)

[Advertise with Us](#)

## Career Suggestion

[SAP Career Suggestion Tool](#)

[Software Testing as a Career](#)

## Interesting

[eBook](#)

[Blog](#)

[Quiz](#)

[SAP eBook](#)

## Execute online

[Execute Java Online](#)

[Execute Javascript](#)

[Execute HTML](#)

[Execute Python](#)

---

© Copyright - Guru99 2021     [Privacy](#).

[Policy](#). | [Affiliate Disclaimer](#) | [ToS](#)