

Regex ?????

????

- <https://likegeeks.com/regex-tutorial-linux/>
- <http://overapi.com/regex>
- <https://regexper.com/#%2F.echo%24%2F> (?????????)
- [The Greatest Regex Trick Ever](#)

?? Regex ?? (awk)

```
# echo "Testing regex using awk" | awk '/regex/{print $0}'
Testing regex using awk

# echo "\ is a special character" | awk '/\V/{print $0}'
\ is a special character

# echo "likegeeks website" | awk '/^likegeeks/{print $0}'
likegeeks website
```

????

```
# [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
echo "This is a test" | sed -n '/test/p'
echo "This is a test" | awk '/test/{print $0}'

# [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
echo "This is a test 2 again" | awk '/test 2/{print $0}'
```

Operators	Description
[abc]	Match any single character from from the listed characters
[a-z]	Match any single character from the range of characters
[^abc]	Match any single character not among listed characters
[^a-z]	Match any single character not among listed range of characters

.	Match any single character except a newline
\	Turn off (escape) the special meaning of a metacharacter
^	Match the beginning of a line.
\$	Match the end of a line.
*	Match zero or more instances of the preceding character or regex.
?	Match zero or one instance of the preceding character or regex.
+	Match one or more instances of the preceding character or regex.
{n,m}	Match a range of occurrences (at least n, no more than m) of preceding character or regex.
	Match the character or expression to the left or right of the vertical bar.

????

- `[:digit:]*` : any number of digits (zero or more)
- `[:digit:]+` : at least one digit
- `[:digit:]?` : zero or one digits
- `[:digit:]{1,3}` : at least one and no more than three digits
- `[:digit:]{2,}` : two or more digits
- `[:digit:]{3}` : three digits

????

???.*[^\$}\+?|()

```
$ cat myfile
There is 10$ on my pocket

awk '/\$/ {print $0}' myfile

echo "\ is a special character" | awk '/\V {print $0}'

echo "This ^ is a test" | sed -n '/s ^/p'
```

???????

```
# ████
$ echo "likegeeks website" | awk '/^likegeeks/ {print $0}'
```

```
# [ ] [ ] [ ] [ ]
$ echo "This is a test" | awk '/test$/{print $0}'
```

```
# [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
$ awk '/^this is a test$/{print $0}' myfile
```

```
# [ ] [ ] [ ] [ ] [ ] [ ]
$ awk '!/^$/{print $0}' myfile
```

????

? dot

```
# cat myfile
this is a test
This is another test
And this is one more
start with this
```

```
# awk '/.st/{print $0}' myfile
this is a test
This is another test
```

? []

```
// [ ] [ ] oth [ ] ith
$ awk '/[oi]th/{print $0}' myfile
This is another test
start with this
```

```
// [ ] [ ] [ ] oth [ ] ith
$ awk '/[^oi]th/{print $0}' myfile
And this is one more
start with this <== [ ] [ ] th
```

????

```
$ awk '/[e-p]st/{print $0}' myfile
this is a test
This is another test
```

```
$ echo "123" | awk '/[0-9][0-9][0-9]/'
```

```
$ awk '/[a-fm-z]st/{print $0}' myfile
```

this is a test

This is another test

Revision #12

Created 1 June 2020 09:42:01 by Admin

Updated 2 November 2023 14:27:17 by Admin