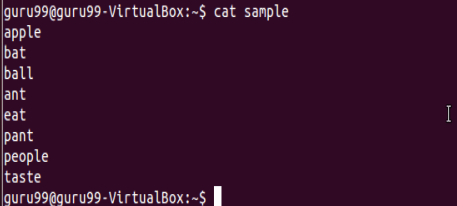
**Basic Regular expressions**

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

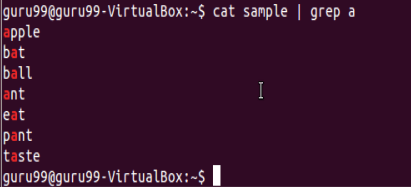
|  |  |
| --- | --- |
| **Symbol** | **Descriptions** |
| . | replaces any character |
| ^ | matches start of string |
| $ | matches end of string |
| \* | matches up zero or more times the preceding character |
| \ | Represent special characters |
| () | Groups regular expressions |
| ? | Matches up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

Let's see an example.

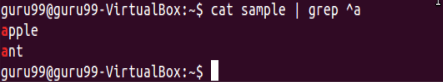
Execute cat sample to see contents of an existing file

[](https://cdn.guru99.com/images/regex1.png)

Search for content containing letter 'a'.

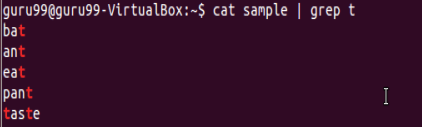
[](https://cdn.guru99.com/images/regex2.png)

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

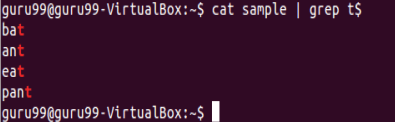
[](https://cdn.guru99.com/images/regex3.png)

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 -

[](https://cdn.guru99.com/images/regex4.png)

Select only those lines that end with t using**$**

[](https://cdn.guru99.com/images/regex5.png)

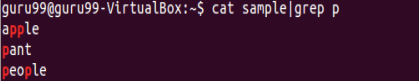
**Interval Regular expressions**

These expressions tell us about the number of occurrences of a character in a string. They 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'

[](https://cdn.guru99.com/images/regex6.png)

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}

[Linux - Regular Expressions](https://cdn.guru99.com/images/regex7.png)

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

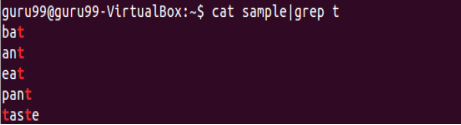
**Extended regular expressions**

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

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

Example:

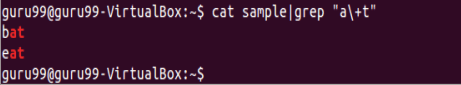
Searching for all characters 't'

[](https://cdn.guru99.com/images/regex8.png)

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

We can use command like

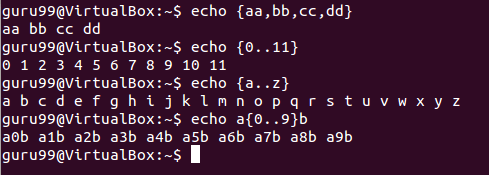
cat sample|grep "a\+t"

[](https://cdn.guru99.com/images/regex9.png)

**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:

[](https://cdn.guru99.com/images/brace_expansion.png)

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 |
| {n, } | Matches the preceding character only when it appears 'n' times or more |

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