regex	Description	Example
`()`	This treats the terms enclosed as one entity	`ma(tri)?x` matches `max` or `matrix`.
`{n}`	This means that the preceding item must match n times.	`[0-9]{3}` matches any three-digit number. `[0-9]{3}` can be expanded as `[0-9][0-9][0-9]`.
`{n,}`	This specifies the minimum number of times the preceding item should match.	`[0-9]{2,}` matches any number that is two digits or longer.
`{n, m}`	This specifies the minimum and maximum number of times the preceding item should match.	`[0-9]{2,5}` matches any number that has two digits to five digits.
`|`	This specifies the alternation—one of the items on either of side of `|` should match.	`Oct (1st | 2nd)` matches `Oct 1st` or `Oct 2nd`.
`\`	This is the escape character for escaping any of the special characters mentioned previously.	`a\.b` matches `a.b`, but not `ajb`. It ignores the special meaning of `.` because of `\`.

For more details on the regular expression components available, you can refer to the following URL:

`http://www.linuxforu.com/2011/04/sed-explained-part-1/`

There's more...

Let's see how the special meanings of certain characters are specified in the regular expressions.

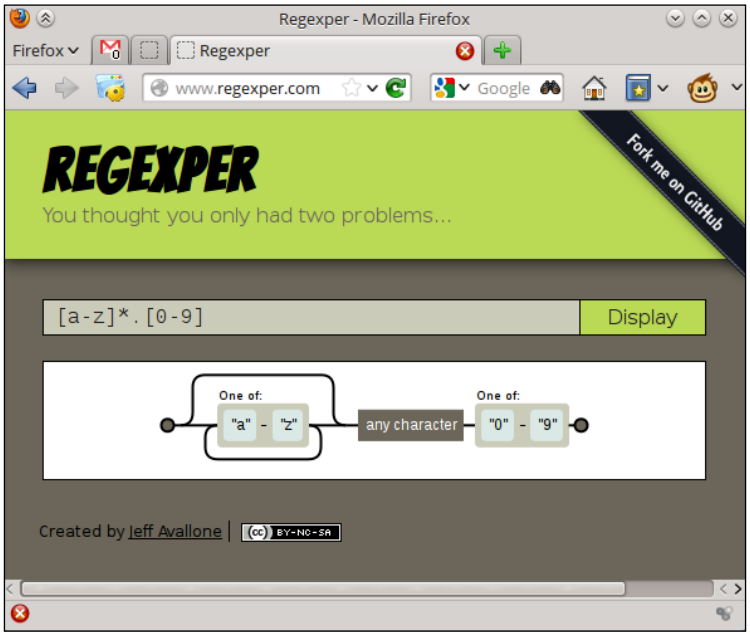
Treatment of special characters

Regular expressions use some characters, such as `$`, `^`, `.`, `*`, `+`, `{`, and `}`, as special characters. But, what if we want to use these characters as normal text characters? Let's see an example of a regex, `a.txt`.

This will match the character `a`, followed by any character (due to the '`.`' character), which is then followed by the string `txt` . However, we want '`.`' to match a literal '`.`' instead of any character. In order to achieve this, we precede the character with a backward slash `\` (doing this is called escaping the character). This indicates that the regex wants to match the literal character rather than its special meaning. Hence, the final regex becomes `a\.txt`.

Visualizing regular expressions

Regular expressions can be tough to understand at times, but for people who are good at understanding things with diagrams, there are utilities available to help in visualizing regex. Here is one such tool that you can use by browsing to `http://www.regexper.com`; it basically lets you enter a regular expression and creates a nice graph to help understand it. Here is a screenshot showing the regular expression we saw in the previous section:



Searching and mining a text inside a file with grep

Searching inside a file is an important use case in text processing. We may need to search through thousands of lines in a file to find out some required data, by using certain specifications. This recipe will help you learn how to locate data items of a given specification from a pool of data.