

Regular Expression Metacharacters

Anchor Metacharacters

The first group of *metacharacter* we will discuss are `^` and `$`. `^` and `$` matches the start and end of a line (or a pattern) respectively and are called anchor metacharacters.

To print lines starting with *I*:

```
$ egrep '^I' country.txt
India ,1014003817 ,New Delhi ,Asia ,Hindi
Italy ,57634327 ,Rome ,Europe ,Italian
Ireland ,3797257 ,Dublin ,Europe ,English
```

To print all lines ending with *ish* :

```
$ egrep 'ish$' country.txt
United Kingdom ,57533000 ,London ,Europe ,English
Argentina ,36955182 ,BuenosA ,S America ,Spanish
Fiji ,832494 ,Suva ,Oceania ,English
United States ,252177000 ,Washington ,N America ,English
Honduras ,6249598 ,Tegucigalpa ,N America ,Spanish
Canada ,31281092 ,Ottawa ,N America ,English
Ireland ,3797257 ,Dublin ,Europe ,English
```

The Dot “.”

To match all the lines in which the country name is exactly 5 characters long:

```
$ egrep '^.....,' country.txt
China ,1261832482 ,Beijing ,Asia ,Chinese
India ,1014003817 ,New Delhi ,Asia ,Hindi
Italy ,57634327 ,Rome ,Europe ,Italian
Japan ,126549976 ,Tokio ,Asia ,Japanese
```

The Character Class

Character class matches any one character mentioned in it. We can give a range using a *hyphen (-)*. The character class can be negated using a \wedge as the first character inside the character class.

```
# Match all lines starting  
# with H,I or J  
$ egrep '[HIJ]' country.txt
```

```
# Same as the last one  
$ egrep '[H-J]' country.txt
```

```
# Match all lines, starting with  
# any character, but not H, I or J  
$ egrep '[^H-J]' country.txt
```

The Alternator and the Grouping Metacharacters

To find all lines containing *Spanish* or *English*, we use:

```
$ egrep 'Spanish|English' country.txt
```

On closer look, we find that *ish* is in both the words, so we can:

```
$ egrep '(Span|Engl)ish' country.txt
```

The Quantifiers

The quantifiers decide how many times the character before them should occur. In the example given below, we are try-

ing to match all lines in which the country name is exactly 5 characters long.

The alternate way of writing,

```
$ egrep '^.....,' country.txt
```

is

```
$ egrep '^.{5},' country.txt
```

To match all countries whose names are either 5 or 6 or 7 characters long is:

```
$ egrep '^.{5,7},' country.txt
```

There are some shortcuts available for the quantifiers. For example,

```
# the two statements are similar
```

```
$ egrep '^ab{0,1}c$' filename
```

```
$ egrep '^ab?c' filename
```

```
# and these two are similar
```

```
$ egrep '^ab{0,}c$' filename
```

```
$ egrep '^ab*c' filename
```

```
# and these two also
```

```
$ egrep '^ab{1,}c$' filename
```

```
$ egrep '^ab+c' filename
```

Let us see some examples involving the expressions we have seen so far. Here instead of searching from a file, we search from standard input. The trick we use is that we know `grep` (or `egrep`) searches for a pattern, and if a pattern is found, then the entire line containing the pattern is shown.

Lets start with searching the string 'gray color'. Note that grey is spelled *grey or gray* and color is spelled as *color or colour*.

So the expression now becomes:

```
$ egrep 'gr[ea]y colou?r'  
grey color  
grey color  
gray color  
gray color  
grey colour  
grey colour  
gray colour  
gray colour  
^d
```

Here \hat{d} (typed as control-d) indicates the end of input.