Unix Regular Expressions

Regular expressions can be used with text processing commands like vi, grep, sed, awk, and many programming languages, either natively or with imported libraries. Regular expressions process strings like mathematical expressions process numbers, thus the term regular.

1) ‘^’ – anchor character for start of line:

If the carat is the first character in an expression, it anchors the remainder of the expression to the start of the line.

Example1: Match all lines that start with ‘A’. e.g. “A plane” Pattern: ‘^A’ Example2: Match all lines that start with ‘hello’. e.g. “hello there”

```
$ grep “^hello” file1
```

2) ‘$’ – anchor character for end of line:

If the carat is the last character in an expression, it anchors the remainder of the expression to the end of the line.

Example1: Match all lines that end with ‘Z’. e.g. “The BUZZ” Pattern: ‘Z$’ Example2: Match all lines that end with ‘done’. e.g. “well done”

```
$ grep “done$” file1
```

3) ‘.’ – any single character:

The ‘.’ character matches any character except the end-of-line.

Example1: Match all lines that contain a single character. e.g. “a” Pattern: ‘^.$’ Example2: Match all lines that end with ‘done’. e.g. “well done”

```
$ grep “done$” file1
```

4) ‘[’ range ‘]’ – a range of characters:

This pattern matches the set of characters specified between the square brackets.

Example1: Match all lines that contain a single digit. e.g. “8” Pattern: ‘^[0-9]$’ Example2: Match all lines that contain any of the letters ‘a’, ‘b’, ‘c’, ‘d’ or ‘e’

```
$ grep “[abcde]”
```

Example3: Match all lines that contain any of the letters ‘a’, ‘b’, ‘c’, ‘d’ or ‘e’.

```
$ grep “[a-e]” file1
```

5) ‘[^’ range ‘]’ – a range of characters to be excluded:

This pattern matches any pattern except the set of characters specified between the square brackets.

Example1: Match all lines that do not contain a digit. e.g. “hello” Pattern: ‘[^0-9]’ Example2: Match all lines that do not contain a vowel

```
$ grep “[^aeiou]” file1
```

6) ‘*’ – ‘zero or more’ modifier:

This modifier matches with zero or more instances of the preceding character-set.

Example1: Match all lines that contain ‘ha’ followed by zero or more instances of ‘p’ and then followed by ‘y’. e.g. “happpy” or “hay” Pattern: ‘hap*y’ Example2: Match all lines that start with a digit following zero or more spaces e.g. “ ” or “2.”

```
$ grep “ *[0-9]” file1
```

7) ‘?’ – ‘zero or one’ modifier:

This modifier matches with zero or one instances of the preceding character-set.

Example1: Match all lines that contain ‘hap’ followed by zero or one instances of ‘p’ and then followed by ‘y’. g. “hapy” or “happy” Pattern: ‘happ?y’ Example2: Match all lines that start with a digit followed by zero or one ‘:’ characters E.g. “1” or “2:”

```
$ grep “^[0-9]:?” file1
```

Now let us go here to this online regex tester and practice together.