DMIT 1530- Web Fundamentals 2

Regular Expressions

Introduction To Regular Expressions

- Regular Expressions (or Regex) is a sequence of characters that defines a pattern (generally a search pattern or a matched pattern for an input).
- A Regular Expression is a generalized way to match a pattern with a sequence of characters.
- Regular expressions are used extensively in URL matching and in supporting Search and Replace in most popular editors

How to write Regular Expressions?

- There are certain elements that are required for writing regular expressions, namely:
  1. Repeaters (*, + and { })
  2. Wildcard (.)
  3. Optional Character (?)
  4. The caret (^) symbol (Setting Position for the match)
  5. The ($) symbol
  6. The ([ ]) pair for sets of characters
  7. A range of characters ([first-last])
  8. The negation ([^])
  9. Character classes
  10. The escape symbol (\)
  11. Grouping of characters (( ))
  12. Choosing one of the patterns (|)
  13. Backreference (\number e.g. \1, \2 etc.)

Repeaters

- The repeater symbols tells the computer to match the preceding character in a repeated way:
  - The '*' means that the preceding character can be matched 0 or more times (up to infinite)– Thus ab*c will match ac, abc, abbc, abbbc etc.
  - The '+' tells the computer to match the preceding character at least once. Thus ab+c will match abc, abbc, abbbc etc.
  - The {..} tells the computer to match the preceding character for the number of times specified in the braces. ab{2}c will match abbc.
  - The { } can also specify a minimum and a maximum number of times. Eg. ab{2,4}c will match abbc, abbbc, abbbbc

Other single letter symbols

- The wildcard character (.) can take the place of any other symbol. Thus a.c will match aac, abc, acc, adc, aec, a1c etc.
- The optional character (?) tells the computer that the preceding character may or may not be present in the pattern. Thus docx? Will match both doc and docx
- The caret (^) symbol tells the computer that the match must start at the beginning of a string or line. Thus ^ai will match patterns like ai and airs but not lairs
- The dollar ($) tells the computer that the match must be at the end of the string or line. Thus ay$ will match day, say, bay, essay but not aye or bayes

Sets, Range of characters and negation

- The [] symbol allows us to specify a set of characters. Thus b[aei]d matched bad, bed and bid
- Using the [ ], we can also specify a range of characters e.g b[a-e]k will match bak, bbk, bck, bdk, bek.
- If we include a ^ inside the [ ], it means negation. [^abc]d, will match any combination of character with d except ad, bd and cd

Character classes

- A character class matches any one of a set of characters. It is used to match the very basic elements of a language like a letter, a digit, a space and a tab.
  - \s : matches any whitespace character like space and tab
  - \S: matches any non-whitespace character
  - \d: matches any digit character
  - \D: matches any non-digit character
  - \w: matches any word character (essentially alpha-numeric)
  - \W: matches any non-word character
  - \b: matches any word boundary- spaces, commas, dashes, semi-colons etc.

Other combinations

- While the \ is used to define character classes, it's also used as escape character. For example if we want to match a +, we use \+.
- Grouping characters (). A set of different symbols of a regular expression can be grouped together as a single unit and behave as a block. Eg. ([A-Z]\w) matches an uppercase letter followed by any character.
- The vertical bar (|) matches any element of a separated by the symbol. Eg. th(e|is|at) matches the, this and that
- \number stands for backreference. ([a-z])\1 will match "ee" in Geek because the character at the second position matches the character at the first position.