**REGEX METACHARACTERS**

*Metacharacters are ordinary characters that cause the compiled regex expression to be interpreted in a special way.  
The following list of characters are all metacharacters with a brief description of what they commonly do :*

|  |  |
| --- | --- |
| **Subexpression** | **Matches** |
| ^ | Matches the beginning of the line. |
| $ | Matches the end of the line. |
| . | Matches any single character except newline. Using **m** option allows it to match the newline as well. |
| [...] | Matches any single character in brackets. |
| [^...] | Matches any single character not in brackets. |
| \A | Beginning of the entire string. |
| \z | End of the entire string. |
| \Z | End of the entire string except allowable final line terminator. |
| re\* | Matches 0 or more occurrences of the preceding expression. |
| re+ | Matches 1 or more of the previous thing. |
| re? | Matches 0 or 1 occurrence of the preceding expression. |
| re{ n} | Matches exactly n number of occurrences of the preceding expression. |
| re{ n,} | Matches n or more occurrences of the preceding expression. |
| re{ n, m} | Matches at least n and at most m occurrences of the preceding expression. |
| a| b | Matches either a or b. |
| (re) | Groups regular expressions and remembers the matched text. |
| (?: re) | Groups regular expressions without remembering the matched text. |
| (?> re) | Matches the independent pattern without backtracking. |
| \w | Matches the word characters. |
| \W | Matches the nonword characters. |
| \s | Matches the whitespace. Equivalent to [\t\n\r\f]. |
| \S | Matches the nonwhitespace. |
| \d | Matches the digits. Equivalent to [0-9]. |
| \D | Matches the nondigits. |
| \A | Matches the beginning of the string. |
| \Z | Matches the end of the string. If a newline exists, it matches just before newline. |
| \z | Matches the end of the string. |
| \G | Matches the point where the last match finished. |
| \n | Back-reference to capture group number "n". |
| \b | Matches the word boundaries when outside the brackets. Matches the backspace (0x08) when inside the brackets. |
| \B | Matches the nonword boundaries. |
| \n, \t, etc. | Matches newlines, carriage returns, tabs, etc. |
| \Q | Escape (quote) all characters up to \E. |
| \E | Ends quoting begun with \Q. |

***Problems:***

|  |  |
| --- | --- |
| **package** com.Soham;  **import** java.util.regex.\*;  **public class** Main {   **public static void** main(String[] args) {  *regix*(**"\n"**, **"are you a god damn \n bitch"**);  }   **public static void** regix(String regex, String find\_me){  **boolean** getmatch = **false**;  Pattern p = Pattern.*compile*(regex);  Matcher m = p.matcher(find\_me);   **while** (m.find()){  System.***out***.println(**"Matcher found "** + m.group() + **" at index "** + m.start() + **" for regex "** + regex + **" with "** + find\_me);  getmatch = **true**;  }   **if** (!getmatch){  System.***out***.printf(**"no match for "** + regex + **" on "** + find\_me);  }  System.***out***.println();  } } | Matcher found  at index 19 for regex  with are you a god damn  bitch |

**How can we force one of these characters to become a regular character? There are two ways to make that happen?**  
(1) put a backslash **\** in front of the character.   
(2) put the metacharacter(s) inside of a **\Q** and a **\E**. An example of a regular period would be \Q.\E