# Student Rec Guide – Regex

November 16th, 2011
CS 1332 – Georgia Institute of Technology

Symbols:

| | |
|---|---|
| `.` | Can match any character. Does not match new line "\n" use [\d\D] instead. |
| `\|` | Matches the character before or after |
| `^` | Not, negates the pattern when inside of square brackets "[^a]" |
| `*` | Matches 0 or more of the characters immediately before |
| `+` | Matches 1 or more of the characters immediately before |
| `?` | Matches 0 or 1 of the characters immediately before |
| `*? or +?` | Reluctant matching, tries to find the smallest match |
| `\` | Regex Escape Character |
| `[]` | Matches any character inside the brackets |
| `{#}` | Defines how many times the character immediately before is repeated |
| `{#0,#1}` | Defines a range of how many times the character immediately before is repeated. |
| `( )` | Groups matches |
| `( ) \1` | Back Reference: References a previous match |

Examples

| | | |
|---|---|---|
| `.*` | "kasjdfk;ljasdfn…" | True |
| `.*` | "" | True |
| `.+` | "" | False |
| `[abc]` | "a" | True |
| `[abc]` | "aa" | False |
| `[abc]*` | "aa" | True |
| `a?b` | "b" | True |
| `a?b` | "ab" | True |
| `a{4}b` | "aaaab" | True |
| `a{5,10}` | "aaaaaaab" | True |
| `(foo\|bar\|zzz)` | "bar" | True |
| `(foo\|bar\|zzz)-\1` | "foo-foo" | True |

Special Characters:

| | |
|---|---|
| \d | Any digit, same as [0-9] |
| \D | Any non-digit, same as [^0-9] |
| \s | A whitespace character, short for [ \t\n\x0b\r\f] |
| \S | A non-whitespace character, for short for [^\s] |
| \w | A word character, short for [a-zA-Z_0-9] |
| \W | A non-word character [^\w] |

Java Methods:

| | |
|---|---|
| s.matches(regexPattern); | Tests the string to see if the regex pattern matches the ENTIRE string. You must account for any leading or trailing characters. |
| s.split(regexPattern, (limit) ); | Splits the string around the regex pattern into an array of strings. Limit defines how many times to split the string. |
| s.replace(regexPattern, replacement); | Replaces any matches of the regex pattern with the provided string. |

***** Important: if you escape in Java, you must escape twice! However, do not double escape “\n” and “\t”. ex. “[\\d\n]” This will match any digit or newline character.**