Regular expression cheat sheet:

A regular expression tries to match its characters with any sequence of characters somewhere in another string. There are a lot of special commands used inside of regular expressions, for which this sheet can be a reference.

Note: When you see "C" or "A", assume that "C" and "A" can be replaced with any character. When you see "foo" or "bar", assume that these can be replaced with arbitrary strings of characters.

Matching:

"C" matches with the character "C"
"." matches with any other character
"^" matches the invisible character at the beginning of a string
"$" matches the invisible character at the end of a string.

Examples:
"^Phil$" matches with "Phil" but not with "PPhil"
"Phil" matches with "aaaaaaaPhilaaa"
"Carrie" does not match with "Alex"
"^CA.GA.$" matches with "CATGAG"
"ca.t" does not match with "CAT"

Counting:

"C*" matches with any number of occurrences of the character "C" in a row.
"C+" matches with at least one occurrence of the character "C" in a row.
"C{n,m}" matches between n and m occurrences of "C" in a row, inclusive.
"C{m}" matches at most m occurrences of the character "C" in a row.
"C?" mean that "C" is optional. It is equivalent to the regular expression "C{0,1}"

Examples:
"R*i*c*h*$" matches with "qqqRRRRRRiihhhhhhh"
"^K*e*s*d*e*n+" matches with "nP"
"^151{2}00?$" matches with "15110"
"^151{3,7}0$" does not match with "15110"

Grouping:

"[bar]" creates an equivalence class of characters "b" "a" and "r" It will match with any of them.
"(foo)" creates a group of characters. It will only match with the string "foo"
"[A-C]" matches everything between "A" and "C".
"[^bar]" matches with everything except for the letters "b" "a" and "r".
"(foo|bar)" matches with either "foo" or "bar".
All the signs associated with counting characters can also be placed after equivalence classes and groups.

Examples:
"[abc][a-c][stu]" matches with "cat".

"(Alex|Carrie|Phil|Rich)" matches with "Phil".
"[Rr]{8}iI(ch|SH)?(ch|SH)?" matches with "RrrRiIch".
"[^a-z]*" will never match with any word made up of only lower case letters.

Escaping:

If you want to use a character that is otherwise reserved for some special regular expression command, you can put a backslash ("\") in front of it. This is called an escape character. Since python and the shell both will have confusing behavior when dealing with escape characters, if you want to use them put the character r before the string.

Example:
r"R*ich" matches with "R*ich" but not with "RRRRRRich".