```
>    Regular Expressions Cheat Sheet
>    A regular expression specifies a set of strings that matches it.
>    This cheat sheet is based off Python 3's Regular Expressions (http://docs.python.org/3/library/re.html)
        but is designed for searches within Sublime Text.


>    Special Characters

    \            Escapes special characters or signals a special sequence.
    .            Matches any single character except a newline.
    ^            Matches the start of the string.
    $            Matches the end of the string.
    *            Greedily matches 0 or more repetitions of the preceding RE.
    *?            Matches 0 or more repetitions of the preceding RE.
    +            Greedily matches 1 or more repetitions of the preceding RE.
    +?            Matches 1 or more repetitions of the preceding RE.
    ?            Greedily matches 0 or 1 repetitions of the preceding RE.
    ??            Matches 0 or 1 repetitions of the preceding RE.
    A|B            Matches A, if A is unmatched then matches B, where A and B are arbitrary REs.
    {m}            Matches exactly m many repetitions of the previous RE.
    {m,n}            Greedily matches from m many to n many repetitions of the previous RE.
    {m,n}?            Matches m many to n many repetitions of the previous RE.


[...]    Indicates a set of characters to match.

    [amk]            Matches 'a', 'm', or 'k'.
    [a-z]            Matches 'a' through 'z'.
    [a-f0-7]        Matches 'a' through 'f' or '0' through '7'.
    [a\-z]            Matches 'a', '-', or 'z'.
    [a-]            Matches 'a' or '-'.
    [-a]            Matches 'a' or '-'.
    [(+*)]            Matches '(', '+', '*', or ')'. [] matches special characters literally.
    [\w]            Matches the character class for '\w'. See character classes.
    [^5]            Matches anything other than '5'. '^' forms the complementary set only as the first character in a set.
    []()]            Matches ']', '(', and ')'. ']' is taken literally only as the first character in a set.
    [()\]]            Matches ']', '(', and ')'.


(...)    Matches the RE inside the parenthesis and assigns a new group.

    (?P<name>...)        The RE matched is accessible by the group indicated by name.
    (?...)            Extension notation which changes a RE's behavior. These do not assign a new group.
    (?aiLmsux)        Sets the corresponding flag to each letter. Does not work within Sublime Text.
    (?:...)            A non-capturing version of parenthesis. The matched substring cannot be retrieved later.
    (?P=name)        Matches the substring matched by the group named name.
    (?#...)            A comment, the contents are ignored.
    (?=...)            Lookahead assertion, the preceding RE only matches if this matches.
    (?!...)            Negative lookahead assertion, the preceding RE only matches if this doesn't match.
    (?<=...)        Positive lookbehind assertion, the following RE will only match if preceded with this fixed length RE.
    (?<!...)        Negative lookbehind assertion, the following RE will only match if not preceded with this fixed length RE.
    (?(id)true|false)    If group id exists then uses the true RE, else use the false RE.


>    Character classes

    \1            Matches the contents of the group labelled by the same number. Acceptable numbers are 1-99.
    \A            Matches at the start of the current string.
    \b            Matches the empty string at the beginning or end of a word. \b matches the boundary between \w and \W.
    \B            Matches the empty string not at the beginning or end of a word.
    \d            Matches any Unicode decimal digit, including 0-9.
    \D            Matches any Unicode non-decimal digit.
    \s            Matches any Unicode whitespace character, including ' ', \t, \n, \r, \f and \v.
    \S            Matches any Unicode non-whitespace character.
    \w            Matches any Unicode word character, including a-z, A-Z, and 0-9.
    \W            Matches any Unicode non-word character.
    \Z            Matches at the end of the string.

    \a            Matches the ASCII Bell (□).
    \f            Matches the ASCII Formfeed ().
    \n            Matches the ASCII Linefeed.
    \r            Matches the ASCII Carriage Return ().
    \t            Matches the ASCII Horizontal Tab.
    \v            Matches the ASCII Vertical Tab (□).
```