Table 13.4 *The Regular Expression Module's Functions*

Syntax	Description
`re.compile( r, f)`	Returns compiled regex `r` with its flags set to `f` if specified. (The flags are described in Table 13.5.)
`re.escape(s)`	Returns string `s` with all nonalphanumeric characters backslash-escaped—therefore, the returned string has no special regex characters
`re.findall( r, s, f)`	Returns all nonoverlapping matches of regex `r` in string `s` (influenced by the flags `f` if given). If the regex has captures, each match is returned as a tuple of captures.
`re.finditer( r, s, f)`	Returns a match object for each nonoverlapping match of regex `r` in string `s` (influenced by the flags `f` if given)
`re.match( r, s, f)`	Returns a match object if the regex `r` matches at the start of string `s` (influenced by the flags `f` if given); otherwise, returns `None`
`re.search( r, s, f)`	Returns a match object if the regex `r` matches anywhere in string `s` (influenced by the flags `f` if given); otherwise, returns `None`
`re.split( r, s, m, f)`	Returns the list of strings that results from splitting string `s` on every occurrence of regex `r` doing up to `m` splits (or as many as possible if no `m` is given, and for Python 3.1 influenced by flags `f` if given). If the regex has captures, these are included in the list between the parts they split.
`re.sub( r, x, s, m, f)`	Returns a copy of string `s` with every (or up to `m` if given, and for Python 3.1 influenced by flags `f` if given) match of regex `r` replaced with `x`—this can be a string or a function; see text
`re.subn( r, x, s m, f)`	The same as `re.sub()` except that it returns a 2-tuple of the resultant string and the number of substitutions that were made

Table 13.5 *The Regular Expression Module's Flags*

Flag	Meaning
`re.A` or `re.ASCII`	Makes `\b`, `\B`, `\s`, `\S`, `\w`, and `\W` assume that strings are ASCII; the default is for these character class shorthands to depend on the Unicode specification
`re.I` or `re.IGNORECASE`	Makes the regex match case-insensitively
`re.M` or `re.MULTILINE`	Makes `^` match at the start and after each newline and `$` match before each newline and at the end
`re.S` or `re.DOTALL`	Makes `.` match every character including newlines
`re.X` or `re.VERBOSE`	Allows whitespace and comments to be included