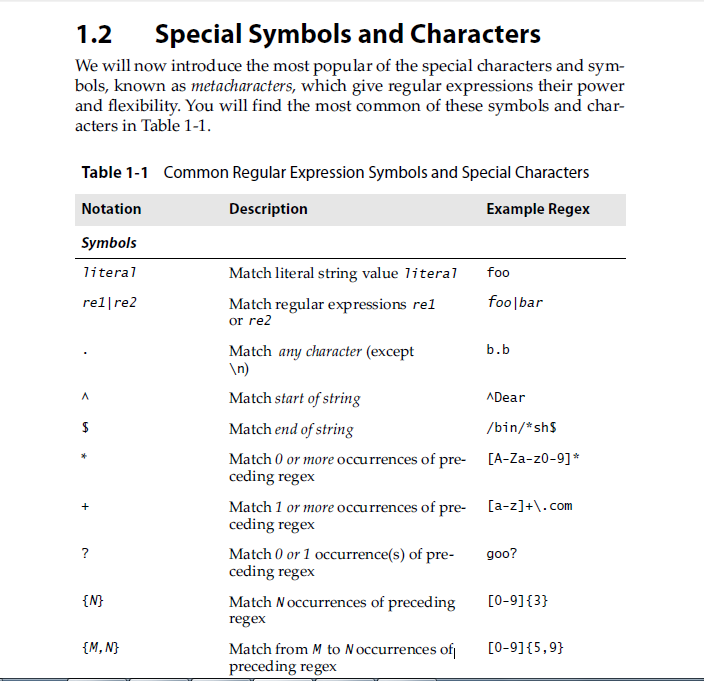
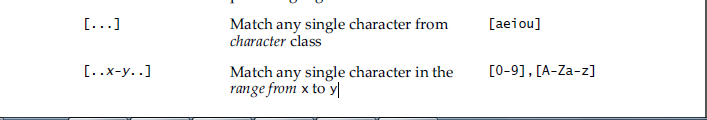
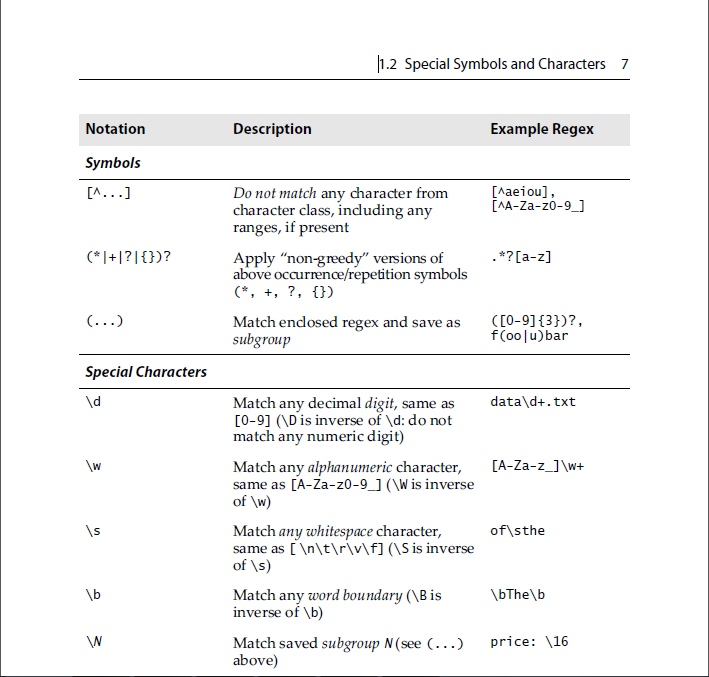
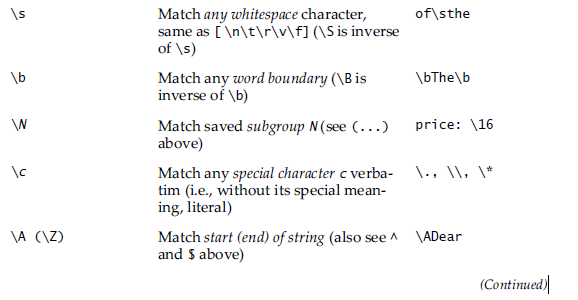
Regular expressions:

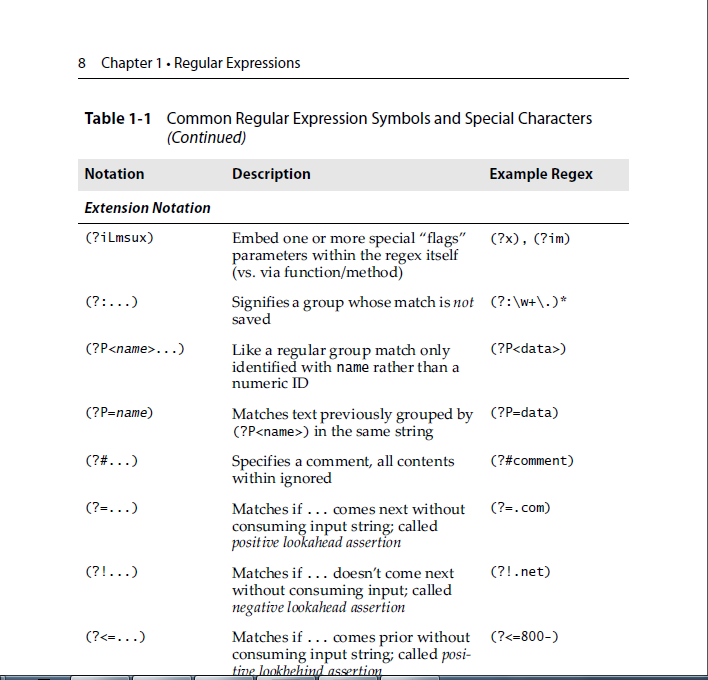
Simple email address: **\w+@\w+\.com** Simple e-mail addresses of the form [*XXX@YYY.com*](mailto:XXX@YYY.com)

**

**

**

**

**

(?=...)

Matches if ... matches next, but doesn’t consume any of the string. This is called a lookahead assertion. For example, Isaac (?=Asimov)will match 'Isaac ' only if it’s followed by 'Asimov'.

(?!...)

Matches if ... doesn’t match next. This is a negative lookahead assertion. For example, Isaac (?!Asimov) will match 'Isaac ' only if it’s *not* followed by 'Asimov'.

(?<=...)

Matches if the current position in the string is preceded by a match for ... that ends at the current position. This is called a *positive lookbehind assertion*. (?<=abc)def will find a match in abcdef, since the lookbehind will back up 3 characters and check if the contained pattern matches. The contained pattern must only match strings of some fixed length, meaning that abc or a|b are allowed, but a\* and a{3,4} are not. Group references are not supported even if they match strings of some fixed length. Note that patterns which start with positive lookbehind assertions will not match at the beginning of the string being searched; you will most likely want to use the [**search()**](https://docs.python.org/2/library/re.html#re.search)function rather than the [**match()**](https://docs.python.org/2/library/re.html#re.match) function:

>>>

**>>> import** **re**

**>>>** m = re.search('(?<=abc)def', 'abcdef')

**>>>** m.group(0)

'def'

This example looks for a word following a hyphen:

>>>

**>>>** m = re.search('(?<=-)\w+', 'spam-egg')

**>>>** m.group(0)

'egg'

(?<!...)

Matches if the current position in the string is not preceded by a match for .... This is called a *negative lookbehind assertion*. Similar to positive lookbehind assertions, the contained pattern must only match strings of some fixed length and shouldn’t contain group references. Patterns which start with negative lookbehind assertions may match at the beginning of the string being searched.