**re.split(pattern, string, maxsplit=0, flags=0)**

Split string by the occurrences of pattern. If capturing parentheses are used in pattern, then the text of all groups in the pattern are also returned as part of the resulting list. If maxsplit is nonzero, at most maxsplit splits occur, and the remainder of the string is returned as the final element of the list.

>>>>>> re.split('\W+', 'Words, words, words.')

['Words', 'words', 'words', '']

>>> re.split('(\W+)', 'Words, words, words.')

['Words', ', ', 'words', ', ', 'words', '.', '']

>>> re.split('\W+', 'Words, words, words.', 1)

['Words', 'words, words.']

>>> re.split('[a-f]+', '0a3B9', flags=re.IGNORECASE)

['0', '3', '9']

If there are capturing groups in the separator and it matches at the start of the string, the result will start with an empty string. The same holds for the end of the string:

>>>>>> re.split('(\W+)', '...words, words...')

['', '...', 'words', ', ', 'words', '...', '']

That way, separator components are always found at the same relative indices within the result list.

Note that split will never split a string on an empty pattern match. For example:

>>>>>> re.split('x\*', 'foo')

['foo']

>>> re.split("(?m)^$", "foo\n\nbar\n")

['foo\n\nbar\n']

Changed in version 3.1: Added the optional flags argument.

**6.2.5.4. Making a Phonebook**

**split()** splits a string into a list delimited by the passed pattern. The method is invaluable for converting textual data into data structures that can be easily read and modified by Python as demonstrated in the following example that creates a phonebook.

First, here is the input. Normally it may come from a file, here we are using triple-quoted string syntax:

>>>>>> text = """Ross McFluff: 834.345.1254 155 Elm Street

...

... Ronald Heathmore: 892.345.3428 436 Finley Avenue

... Frank Burger: 925.541.7625 662 South Dogwood Way

...

...

... Heather Albrecht: 548.326.4584 919 Park Place"""

The entries are separated by one or more newlines. Now we convert the string into a list with each nonempty line having its own entry:

>>>>>> entries = re.split("\n+", text)

>>> entries

['Ross McFluff: 834.345.1254 155 Elm Street',

'Ronald Heathmore: 892.345.3428 436 Finley Avenue',

'Frank Burger: 925.541.7625 662 South Dogwood Way',

'Heather Albrecht: 548.326.4584 919 Park Place']

Finally, split each entry into a list with first name, last name, telephone number, and address. We use the maxsplit parameter of split() because the address has spaces, our splitting pattern, in it:

>>>>>> [re.split(":? ", entry, 3) for entry in entries]

[['Ross', 'McFluff', '834.345.1254', '155 Elm Street'],

['Ronald', 'Heathmore', '892.345.3428', '436 Finley Avenue'],

['Frank', 'Burger', '925.541.7625', '662 South Dogwood Way'],

['Heather', 'Albrecht', '548.326.4584', '919 Park Place']]

The :? pattern matches the colon after the last name, so that it does not occur in the result list. With a maxsplit of 4, we could separate the house number from the street name:

>>>>>> [re.split(":? ", entry, 4) for entry in entries]

[['Ross', 'McFluff', '834.345.1254', '155', 'Elm Street'],

['Ronald', 'Heathmore', '892.345.3428', '436', 'Finley Avenue'],

['Frank', 'Burger', '925.541.7625', '662', 'South Dogwood Way'],

['Heather', 'Albrecht', '548.326.4584', '919', 'Park Place']]