Home

Pages Classes Methods

Search

Table of Contents

Show/hide navigation

Pages

Character Selectors

Character Selector

A *character selector* is a string argument accepted by certain Ruby methods. Each of these instance methods accepts one or more character selectors:

- `String#tr` (selector, replacements): returns a new string.
- `String#tr!` (selector, replacements): returns `self` or `nil`.
- `String#tr_s` (selector, replacements): returns a new string.
- `String#tr_s!` (selector, replacements): returns `self` or `nil`.
- `String#count(*selectors)` : returns the count of the specified characters.
- `String#delete(*selectors)` : returns a new string.
- `String#delete!(*selectors)` : returns `self` or `nil`.
- `String#squeeze(*selectors)` : returns a new string.
- `String#squeeze!(*selectors)` : returns `self` or `nil`.

A character selector identifies zero or more characters in `self` that are to be operands for the method.

In this section, we illustrate using method `String#delete(selector)`, which deletes the selected characters.

In the simplest case, the characters selected are exactly those contained in the selector itself:

```
'abracadabra'.delete('a')    # => "brcdbr"
'abracadabra'.delete('ab')   # => "rcdr"
'abracadabra'.delete('abc')  # => "rdr"
'0123456789'.delete('258')   # => "0134679"
'!@#$%&*()_+'.delete('+&#') # => "!@$%*()_"
'тест'.delete('т')           # => "ec"
'こんにちは'.delete('に')     # => "こんちは"
```

Note that order and repetitions do not matter:

```
'abracadabra'.delete('dcab') # => "rr"
'abracadabra'.delete('aaaa') # => "brcdbr"
```

In a character selector, these three characters get special treatment:

- A leading caret (`'^'`) functions as a “not” operator for the characters to its right:

  ```
  'abracadabra'.delete('^bc') # => "bcb"
  '0123456789'.delete('^852') # => "258"
  ```

- A hyphen (`'-'`) between two other characters defines a range of characters instead of a plain string of characters:

```
'abracadabra'.delete('a-d') # => "rr"
'0123456789'.delete('4-7')  # => "012389"
'!@#$%&*()_+'.delete(' -/') # => "@^_"

# May contain more than one range.
'abracadabra'.delete('a-cq-t') # => "d"

# Ranges may be mixed with plain characters.
'0123456789'.delete('67-950-23') # => "4"

# Ranges may be mixed with negations.
'abracadabra'.delete('^a-c') # => "abacaaba"
```

- A backslash (`'\'`) acts as an escape for a caret, a hyphen, or another backslash:

```
'abracadabra^'.delete('\^bc')   # => "araadara"
'abracadabra-'.delete('a\-d')   # => "brcbr"
"hello\r\nworld".delete("\r")   # => "hello\nworld"
"hello\r\nworld".delete("\\r")  # => "hello\r\nwold"
"hello\r\nworld".delete("\\\r") # => "hello\nworld"
```

Multiple Character Selectors

These instance methods accept multiple character selectors:

- `String#count(*selectors)` : returns the count of the specified characters.
- `String#delete(*selectors)` : returns a new string.
- `String#delete!(*selectors)` : returns `self` or `nil`.
- `String#squeeze(*selectors)` : returns a new string.
- `String#squeeze!(*selectors)` : returns `self` or `nil`.

In effect, the given selectors are formed into a single selector consisting of only those characters common to *all* of the given selectors.

All forms of selectors may be used, including negations, ranges, and escapes.

Each of these pairs of method calls is equivalent:

```
s.delete('abcde', 'dcbfg')
s.delete('bcd')

s.delete('^abc', '^def')
s.delete('^abcdef')

s.delete('a-e', 'c-g')
s.delete('cde')
```

Validate

Generated by RDoc 6.4.0.

Based on Darkfish by Michael Granger.

Ruby-doc.org is a service of James Britt and Neurogami, purveyors of fine dance noise