

str.module

`str.module`

String functions

str.ltrim

`str.ltrim :str [ws str.whitespace]`

Trims string's left end spaces

Parameters

str: string

ws: whitespace characters

Returns

A left trimmed string.

Examples

```
show str.ltrim "| abc de fgh |"
```

str.rtrim

`str.rtrim :str [ws str.whitespace]`

Trims string's right end spaces

Parameters

str: string

ws: whitespace characters

Returns

A right trimmed string.

Examples

```
show str.rtrim "| abc de fgh |"
```

str.trim

`str.trim :str [ws str.whitespace]`

Trims all spaces from a string

Parameters

str: string

ws: whitespace characters

Returns

A trimmed string.

Examples

```
show str.trim "| 1,1 ,3 |"
```

str.substr

```
str.substr :string :begin :end [range iseq :begin :end]
```

Returns a substring of the original string

Parameters

begin: starting position

end: ending position

range: list, integer sequence between begin and end

Returns

a string, from begin to end.

Examples

```
show str.substr "AABAACAADAABAABA" 2 4
```

str.overlapping.search

```
str.overlapping.search :pattern :text [begin 1] [end count :pattern] [out []] [s str.substr :text :begin :end]
```

Searches a (overlapping) pattern in a text

Parameters

pattern: pattern

text: text

Returns

a list of patterns starting positions.

Examples

```
show str.overlapping.search "AABA" "AABAACAADAABAABA"
```

str.search

```
str.search :pattern :text [n count :pattern]
```

Searches a (non overlapping) pattern in a text

Parameters

pattern: pattern

text: text

Returns

a list of patterns starting positions.

Examples

```
show str.search "AABA" "AABAACAADAABAABA"
```

str.replace

```
str.replace :pattern :text :string
```

Replaces a pattern in a text

Parameters

pattern: pattern

text: text

string: string that replaces text

Returns

a string.

Examples

```
show str.replace ", " "12,3 "
```

```
show map [str.replace ", ? ".] [12,3 45,5]
```

str.remove

```
str.remove :pattern :text
```

Removes a pattern in a text

Parameters

pattern: pattern

text: text

Returns

a string.

Examples

```
show str.remove ", "12,3
```

str.split

```
str.split :str :by [:acc []] [:w "||"]
```

Splits a string into substrings

Parameters

str: string

by: separating character

Returns

a list.

Examples

```
show str.split "12,3 ",
```

```
show map [str.split ? "="] str.split "|uno=1,due=2,tre=3 | ",
```