Package ‘filesstrings’

February 11, 2024

Type Package
Title Handy File and String Manipulation
Version 3.4.0
Maintainer Rory Nolan <rorynolan@gmail.com>
Description This started out as a package for file and string manipulation. Since then, the 'fs' and 'strex' packages emerged, offering functionality previously given by this package (but it's done better in these new ones). Those packages have hence almost pushed 'filesstrings' into extinction. However, it still has a small number of unique, handy file manipulation functions which can be seen in the vignette. One example is a function to remove spaces from all file names in a directory.
License GPL-3
BugReports https://github.com/rorynolan/filesstrings/issues
Depends R (>= 3.5), stringr (>= 1.5)
Imports checkmate (>= 1.9.3), magrittr (>= 1.5), purrr (>= 0.3.0), rlang (>= 0.3.3), strex (>= 1.6), stringi (>= 1.7.8), withr (>= 2.1.0)
Suggests covr, dplyr, knitr, rmarkdown, spelling, testthat (>= 2.1)
VignetteBuilder knitr
Encoding UTF-8
Language en-US
RoxygenNote 7.2.3
NeedsCompilation no
Author Rory Nolan [aut, cre, cph] (<https://orcid.org/0000-0002-5239-4043>), Sergi Padilla-Parra [ths] (<https://orcid.org/0000-0002-8010-9481>)
Repository CRAN
Date/Publication 2024-02-11 19:50:02 UTC
\textbf{R topics documented:}

\begin{verbatim}
all_equal ................................................................. 2
before_last_dot ........................................................... 4
can_be_numeric ............................................................ 4
create_dir ................................................................. 5
currency ................................................................. 5
extend_char_vec ............................................................ 6
extract_non_numerics ...................................................... 7
extract_numbers ........................................................... 7
filestrings .............................................................. 8
group_close ............................................................... 8
locate_braces .............................................................. 9
match_arg ................................................................. 9
move_files ............................................................... 10
nice_file_nums ............................................................ 11
nth_number_after_mth ................................................... 12
nth_number_before_mth .................................................. 13
put_in_pos ............................................................... 14
remove_dir ............................................................... 14
remove_filename_spaces ............................................... 15
rename_with_nums ........................................................ 16
str_after_nth ............................................................. 17
str_before_nth ............................................................ 17
str_elem ................................................................. 18
str elems ................................................................. 18
str_give_ext ............................................................. 19
str_locate_nth ........................................................... 19
str nice_nums ............................................................ 20
str_paste_elems .......................................................... 20
str_remove_quoted ........................................................ 21
str_singleize ............................................................. 21
str_split_by_nums ....................................................... 22
str_split_camel_case .................................................... 22
str_to_vec ............................................................... 23
str\_trim\_anything ........................................................ 23
unitize\_dirs ............................................................. 24
\end{verbatim}

\textbf{Index} ............................................................... 25

\texttt{all\_equal}  \hspace{1cm} \textit{An alternative version of \texttt{base::all.equal}().}
all_equal

Description
This function will return TRUE whenever base::all.equal() would return TRUE, however it will also return TRUE in some other cases:

- If a is given and b is not, TRUE will be returned if all of the elements of a are the same.
- If a is a scalar and b is a vector or array, TRUE will be returned if every element in b is equal to a.
- If a is a vector or array and b is a scalar, TRUE will be returned if every element in a is equal to b.

This function ignores names and attributes (except for dim).
When this function does not return TRUE, it returns FALSE (unless it errors). This is unlike base::all.equal().

Usage
all_equal(a, b = NULL)

Arguments
a A vector, array or list.
b Either NULL or a vector, array or list of length either 1 or length(a).

Value
TRUE if "equality of all" is satisfied (as detailed in 'Description' above) and FALSE otherwise.

Note
- This behaviour is totally different from base::all.equal().
- There's also dplyr::all_equal(), which is different again. To avoid confusion, always use the full filesstrings::all_equal() and never library(filesstrings) followed by just all_equal().

Examples
all_equal(1, rep(1, 3))
all_equal(2, 1:3)
all_equal(1:4, 1:4)
all_equal(1:4, c(1, 2, 3, 3))
all_equal(rep(1, 10))
all_equal(c(1, 88))
all_equal(1:2)
all_equal(list(1:2))
all_equal(1:4, matrix(1:4, nrow = 2)) # note that this gives TRUE
before_last_dot  Get the part of a string before the last period.

Description

Copy of `strex::str_before_last_dot()`.

Usage

```c
before_last_dot(...)  
str_before_last_dot(...) 
```

Arguments

...  Pass-through to strex function.

can_be_numeric  Check if a string could be considered as numeric.

description

Copy of `strex::str_can_be_numeric()`.

Usage

```c
can_be_numeric(...)  
str_can_be_numeric(...) 
```

Arguments

...  Pass-through to strex function.
create_dir

Create directories if they don’t already exist

Description
Given the names of (potential) directories, create the ones that do not already exist.

Usage
create_dir(...)

Arguments
...

The names of the directories, specified via relative or absolute paths. Duplicates are ignored.

Value
Invisibly, a vector with a TRUE for each time a directory was actually created and a FALSE otherwise. This vector is named with the paths of the directories that were passed to the function.

Examples
## Not run:
create_dir(c("mydir", "yourdir"))
remove_dir(c("mydir", "yourdir"))

## End(Not run)

currency

Get the currencies of numbers within a string.

Description
See strex::str_extract_currencies().

Usage
str_extract_currencies(...)
extract_currencies(...)
str_nth_currency(...)
nth_currency(...)
extend_char_vec

   str_first_currency(...)  
first_currency(...)       
str_last_currency(...)   
last_currency(...)       

Arguments

...  Pass-through to strex function.

-------------

extend_char_vec Pad a character vector with empty strings.
-------------

Description

Extend a character vector by appending empty strings at the end.

Usage

   extend_char_vec(char_vec, extend_by = NA, length_out = NA)  
   str_extend_char_vec(char_vec, extend_by = NA, length_out = NA)

Arguments

char_vec A character vector. The thing you wish to expand.
extend_by A non-negative integer. By how much do you wish to extend the vector?
length_out A positive integer. How long do you want the output vector to be?

Value

A character vector.

Examples

   extend_char_vec(1:5, extend_by = 2)  
   extend_char_vec(c("a", "b"), length_out = 10)
extract_non_numerics

Extract non-numbers from a string.

Description
Copies of `strex::str_extract_non_numerics()` and friends.

Usage

```r
extract_non_numerics(...)  
str_extract_non_numerics(...)  
nth_non_numeric(...)  
str_nth_non_numeric(...)  
first_non_numeric(...)  
str_first_non_numeric(...)  
last_non_numeric(...)  
str_last_non_numeric(...)  
```

Arguments

```r
...
```
Pass-through to `strex` function.

extract_numbers

Extract numbers from a string.

Description
Copies of `strex::str_extract_numbers()` and friends.

Usage

```r
extract_numbers(...)  
str_extract_numbers(...)  
nth_number(...)  
str_nth_number(...)  
```
first_number(...)  
str_first_number(...)  
last_number(...)  
str_last_number(...)  

Arguments  
...  
Pass-through tostrexf function.

Description  
This started out as a package for file and string manipulation. Since then, the fs file manipulation package and the strex string manipulation package emerged, offering functionality previously given by this package (but slightly better). Those packages have hence almost pushed ‘filesstrings’ into extinction. However, it still has a small number of unique, handy file manipulation functions which can be seen in the vignette. One example is a function to remove spaces from all file names in a directory.

References  

Usage  
group_close(vec_ascending, max_gap = 1)
locate_braces

Arguments

vec_ascending  A strictly increasing numeric vector.
max_gap  The biggest allowable gap between adjacent elements for them to be considered part of the same group.

Value

A where each element is one group, as a numeric vector.

Examples

```r
group_close(1:10, 1)
group_close(1:10, 0.5)
group_close(c(1, 2, 4, 10, 11, 14, 20, 25, 27), 3)
```

locate_braces  *Locate the braces in a string.*

Description

Copy of `strex::str_locate_braces()`.

Usage

```r
locate_braces(...)
str_locate_braces(...)
```

Arguments

`...`  Pass-through to `strex` function.

match_arg  *Argument Matching*

Description

Copy of `strex::match_arg()`.

Usage

```r
match_arg(...)
str_match_arg(...)
```

Arguments

`...`  Pass-through to `strex` function.
move_files

Move files around.

Description

Move specified files into specified directories

Usage

move_files(files, destinations, overwrite = FALSE)

file.move(files, destinations, overwrite = FALSE)

Arguments

files A character vector of files to move (relative or absolute paths).
destinations A character vector of the destination directories into which to move the files.
overwrite Allow overwriting of files? Default no.

Details

If there are \( n \) files, there must be either 1 or \( n \) directories. If there is one directory, then all \( n \) files are moved there. If there are \( n \) directories, then each file is put into its respective directory. This function also works to move directories.

If you try to move files to a directory that doesn’t exist, the directory is first created and then the files are put inside.

Value

Invisibly, a logical vector with a TRUE for each time the operation succeeded and a FALSE for every fail.

Examples

```r
## Not run:
dir.create("dir")
files <- c("1litres_1.txt", "1litres_30.txt", "3litres_5.txt")
file.create(files)
file.move(files, "dir")

## End(Not run)
```
nice_file_nums

Make file numbers comply with alphabetical order

Description

If files are numbered, their numbers may not comply with alphabetical order, i.e. "file2.ext" comes after "file10.ext" in alphabetical order. This function renames the files in the specified directory such that they comply with alphabetical order, so here "file2.ext" would be renamed to "file02.ext".

Usage

nice_file_nums(dir = ".", pattern = NA)

Arguments

- **dir**: Path (relative or absolute) to the directory in which to do the renaming (default is current working directory).
- **pattern**: A regular expression. If specified, files to be renamed are restricted to ones matching this pattern (in their name).

Details

It works on file names with more than one number in them e.g. "file01part3.ext" (a file with 2 numbers). All the file names that it works on must have the same number of numbers, and the non-number bits must be the same. One can limit the renaming to files matching a certain pattern. This function wraps nice_nums(), which does the string operations, but not the renaming. To see examples of how this function works, see the examples in that function’s documentation.

Value

A logical vector with a TRUE for each successful rename (should be all TRUEs) and a FALSE otherwise.

Examples

```r
## Not run:
dir.create("NiceFileNums_test")
setwd("NiceFileNums_test")
files <- c("1litres_1.txt", "1litres_30.txt", "3litres_5.txt")
file.create(files)
nice_file_nums()
nice_file_nums(pattern = "\.txt$")
setwd("..")
dir.remove("NiceFileNums_test")

## End(Not run)
```
nth_number_after_mth  
Find the \( n \)th number after the \( m \)th occurrence of a pattern.

Description
Copy of \texttt{strex::str\_nth\_number\_after\_mth()}.  

Usage
\begin{verbatim}
nth_number_after_mth(...) 
str_nth_number_after_mth(...) 
nth_number_after_first(...) 
nth_number_after_last(...) 
first_number_after_mth(...) 
last_number_after_mth(...) 
first_number_after_first(...) 
first_number_after_last(...) 
last_number_after_first(...) 
last_number_after_last(...) 
str_nth_number_after_first(...) 
str_nth_number_after_last(...) 
str_first_number_after_mth(...) 
str_last_number_after_mth(...) 
str_first_number_after_first(...) 
str_first_number_after_last(...) 
str_last_number_after_first(...) 
str_last_number_after_last(...) 
\end{verbatim}

Arguments
\begin{verbatim}
...  
\end{verbatim}
Pass-through to \texttt{strex} function.
Find the \( n \)th number before the \( m \)th occurrence of a pattern.

**Description**

Copy of `strex::str_nth_number_before_mth()`.

**Usage**

```c
nth_number_before_mth(...)
str_nth_number_before_mth(...)
nth_number_before_first(...)
nth_number_before_last(...)
first_number_before_mth(...)
last_number_before_mth(...)
first_number_before_first(...)
first_number_before_last(...)
last_number_before_first(...)
last_number_before_last(...)
str_nth_number_before_first(...)
str_nth_number_before_last(...)
str_first_number_before_mth(...)
str_last_number_before_mth(...)
str_first_number_before_first(...)
str_first_number_before_last(...)
str_last_number_before_first(...)
str_last_number_before_last(...)
```

**Arguments**

`...`  
Pass-through to strex function.
Put specified strings in specified positions in an otherwise empty character vector.

Description
Create a character vector with a set of strings at specified positions in that character vector, with the rest of it taken up by empty strings.

Usage

\[
\text{put\_in\_pos(strings, positions)}
\]

\[
\text{str\_put\_in\_pos(strings, positions)}
\]

Arguments

- **strings**: A character vector of the strings to put in positions (coerced by `as.character` if not character already).
- **positions**: The indices of the character vector to be occupied by the elements of strings. Must be the same length as strings or of length 1.

Value
A character vector.

Examples

\[
\text{put\_in\_pos(1:3, c(1, 8, 9))}
\]

\[
\text{put\_in\_pos(c("Apple", "Orange", "County"), c(5, 7, 8))}
\]

\[
\text{put\_in\_pos(1:2, 5)}
\]

Remove directories

Description
Delete directories and all of their contents.

Usage

\[
\text{remove\_dir(...)}
\]

\[
\text{dir.remove(...)}
\]
Arguments

... The names of the directories, specified via relative or absolute paths.

Value

Invisibly, a logical vector with TRUE for each success and FALSE for failures.

Examples

```r
## Not run:
sapply(c("mydir1", "mydir2"), dir.create)
remove_dir(c("mydir1", "mydir2"))
## End(Not run)
```

---

**remove_filename_spaces**

*Remove spaces in file names*

Description

Remove spaces in file names in a specified directory, replacing them with whatever you want, default nothing.

Usage

```r
remove_filename_spaces(dir = ".", pattern = "", replacement = "")
```

Arguments

- **dir**: The directory in which to perform the operation.
- **pattern**: A regular expression. If specified, only files matching this pattern will be treated.
- **replacement**: What do you want to replace the spaces with? This defaults to nothing, another sensible choice would be an underscore.

Value

A logical vector indicating which operation succeeded for each of the files attempted. Using a missing value for a file or path name will always be regarded as a failure.
rename_with_nums

Example

## Not run:

dir.create("RemoveFileNameSpaces_test")
setwd("RemoveFileNameSpaces_test")
files <- c("1litres 1.txt", "1litres 30.txt", "3litres 5.txt")
file.create(files)
remove_filename_spaces()
list.files()
setwd("..")
dir.remove("RemoveFileNameSpaces_test")

## End(Not run)

rename_with_nums Replace file names with numbers

Description

Rename the files in the directory, replacing file names with numbers only.

Usage

rename_with_nums(dir = ".", pattern = NULL)

Arguments

dir The directory in which to rename the files (relative or absolute path). Defaults to current working directory.

pattern A regular expression. If specified, only files with names matching this pattern will be treated.

Value

A logical vector with a TRUE for each successful renaming and a FALSE otherwise.

Examples

## Not run:
dir.create("RenameWithNums_test")
setwd("RenameWithNums_test")
files <- c("1litres 1.txt", "1litres 30.txt", "3litres 5.txt")
file.create(files)
rename_with_nums()
list.files()
setwd("..")
dir.remove("RenameWithNums_test")

## End(Not run)
**str_after_nth**  
*Text after the nth occurrence of pattern.*

**Description**
Copies of 

```cpp
strex::str_after_nth()
```

**Usage**

```cpp
str_after_nth(...)  
after_nth(...)  
str_after_first(...)  
after_first(...)  
str_after_last(...)  
after_last(...)  
```

**Arguments**

...  
Pass-through to strex function.

---

**str_before_nth**  
*Text before the nth occurrence of pattern.*

**Description**
Copies of 

```cpp
strex::str_before_nth()
```

**Usage**

```cpp
str_before_nth(...)  
before_nth(...)  
str_before_first(...)  
before_first(...)  
str_before_last(...)  
before_last(...)  
```
str_elem

Extract a single character from a string, using its index.

Description

Copy of `strex::str_elem()`.

Usage

```cpp
str_elem(...)  
elem(...)  
```

Arguments

... Pass-through to `strex` function.

str elems

Extract several single elements from a string.

Description

Copy of `strex::str elems()`.

Usage

```cpp
str elems(...)  
elems(...)  
```

Arguments

... Pass-through to `strex` function.
**str_give_ext**  
Ensure a file name has the intended extension.

**Description**  
Copy of `strex::str_give_ext()`.

**Usage**  
```cpp
str_give_ext(...)  
give_ext(...)  
```

**Arguments**  
... Pass-through to strex function.

---

**str_locate_nth**  
Get the indices of the nth instance of a pattern.

**Description**  
Copy of `strex::str_locate_nth()`.

**Usage**  
```cpp
str_locate_nth(...)  
locate_nth(...)  
str_locate_first(...)  
locate_first(...)  
str_locate_last(...)  
locate_last(...)  
```

**Arguments**  
... Pass-through to strex function.
str_nice_nums  Make string numbers comply with alphabetical order.

Description
Copy of `strex::str_alphord_nums()`.

Usage
str_nice_nums(...)
nice_nums(...)
str_alphord_nums(...)
alphord_nums(...)

Arguments
...
    Pass-through to strex function.

str_pasteelems  Extract bits of a string and paste them together.

Description
Copy of `strex::str_paste elems()`.

Usage
str_paste elems(...)
paste elems(...)

Arguments
...
    Pass-through to strex function.
**str_remove_quoted**

Remove the quoted parts of a string.

**Description**

Copy of `strex::str_remove_quoted()`.

**Usage**

```
str_remove_quoted(...)
```

```
remove_quoted(...)
```

**Arguments**

```
...
```

Pass-through to `strex` function.

**str_singleize**

Remove back-to-back duplicates of a pattern in a string.

**Description**

Copy of `strex::str_singleize()`.

**Usage**

```
str_singleize(...)
```

```
singleize(...)
```

**Arguments**

```
...
```

Pass-through to `strex` function.
\begin{verbatim}

str_split_by_nums  Split a string by its numeric characters.

Description
Copy of \texttt{strex::str_split_by_numbers()}.  

Usage
\begin{verbatim}
str_split_by_nums(...)  
split_by_nums(...)  
split_by_numbers(...)  
str_split_by_numbers(...)  
\end{verbatim}

Arguments
\begin{verbatim}
...  Pass-through to \texttt{strex} function.  
\end{verbatim}

str_split_camel_case  Split a string based on CamelCase

Description
See \texttt{strex::str_split_camel_case()}.  

Usage
\begin{verbatim}
str_split_camel_case(string, lower = FALSE)  
split_camel_case(string, lower = FALSE)  
\end{verbatim}

Arguments
\begin{verbatim}
string  A character vector.  
lower  Do you want the output to be all lower case (or as is)?  
\end{verbatim}

\end{verbatim}
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>str_to_vec</td>
<td>Convert a string to a vector of characters</td>
</tr>
</tbody>
</table>

**Description**

Copy of `strex::str_to_vec()`.

**Usage**

```r
str_to_vec(...)  
to_vec(...)  
```

**Arguments**

```r
...  
```

Pass-through to `strex` function.

---

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>str_trim_anything</td>
<td>Trim something other than whitespace.</td>
</tr>
</tbody>
</table>

**Description**

Copy of `strex::str_trim_anything()`.

**Usage**

```r
str_trim_anything(...)  
trim_anything(...)  
```

**Arguments**

```r
...  
```

Pass-through to `strex` function.
unitize_dirs

Put files with the same unit measurements into directories

Description

Say you have a number of files with "5min" in their names, number with "10min" in the names, a number with "15min" in their names and so on, and you’d like to put them into directories named "5min", "10min", "15min" and so on. This function does this, but not just for the unit "min", for any unit.

Usage

unitize_dirs(unit, pattern = NULL, dir = ".")

Arguments

  unit          The unit upon which to base the categorizing.
  pattern       If set, only files with names matching this pattern will be treated.
  dir           In which directory do you want to perform this action (defaults to current)?

Details

This function takes the number to be the last number (as defined in nth_number()) before the first occurrence of the unit name. There is the option to only treat files matching a certain pattern.

Value

Invisibly TRUE if the operation is successful, if not there will be an error.

Examples

```r
## Not run:
dir.create("UnitDirs_test")
setwd("UnitDirs_test")
files <- c("1litres_1.txt", "1litres_3.txt", "3litres.txt", "5litres_1.txt")
file.create(files)
unitize_dirs("litres", ".\".txt")
setwd("..")
dir.remove("UnitDirs_test")

## End(Not run)
```
after_first (str_after_nth), 17
after_last (str_after_nth), 17
after_nth (str_after_nth), 17
all_equal, 2
alphord_nums / str_nice_nums, 20
as.character, 14
base::all.equal(), 2, 3
before_first (str_before_nth), 17
before_last (str_before_nth), 17
before_last_dot, 4
before_nth (str_before_nth), 17
can_be_numeric, 4
create_dir, 5
currency, 5
dir.remove (remove_dir), 14
dplyr::all.equal(), 3
elem (str_elem), 18
elems (str_elements), 18
extend_char_vec, 6
extract_currencies (currency), 5
extract_non_numerics, 7
extract_numbers, 7
file.move (move_files), 10
filesstrings, 8
filesstrings-package (filesstrings), 8
first_currency (currency), 5
first_number (extract_numbers), 7
first_number_after_first
(nth_number_after_mth), 12
first_number_after_last
(nth_number_after_mth), 13
first_number_after_mth
(nth_number_after_mth), 13
give_ext (str_give_ext), 19
group_close, 8
last_currency (currency), 5
last_non_numeric
(extract_non_numerics), 7
last_number (extract_numbers), 7
last_number_after_first
(nth_number_after_mth), 12
last_number_after_last
(nth_number_after_mth), 12
last_number_after_mth
(nth_number_after_mth), 12
last_number_before_first
(nth_number_after_mth), 13
last_number_before_last
(nth_number_after_mth), 13
last_number_before_mth
(nth_number_after_mth), 13
locate_braces, 9
locate_first (str_locate_nth), 19
locate_last (str_locate_nth), 19
locate_nth (str_locate_nth), 19
match_arg, 9
move_files, 10
nice_file_nums, 11
nice_nums (str_nice_nums), 20
nice_nums(), 11
nth_currency (currency), 5
nth_non_numeric (extract_non_numerics), 7
nth_number(extract_numbers), 7
nth_number(), 24
nth_number_after_first
  (nth_number_after_mth), 12
nth_number_after_last
  (nth_number_after_mth), 12
nth_number_after_mth, 12
nth_number_before_first
  (nth_number_before_mth), 13
nth_number_before_last
  (nth_number_before_mth), 13
nth_number_before_mth, 13
paste_elems(str_paste_elems), 20
put_in_pos, 14
remove_dir, 14
remove_filename_spaces, 15
remove_quoted(str_remove_quoted), 21
rename_with_nums, 16
singleize(str_singleize), 21
split_by_numbers(str_split_by_nums), 22
split_by_nums(str_split_by_nums), 22
split_camel_case
  (str_split_camel_case), 22
str_after_first(str_after_nth), 17
str_after_last(str_after_nth), 17
str_after_nth, 17
str_alphord_nums(str_nice_nums), 20
str_before_first(str_before_nth), 17
str_before_last(str_before_nth), 17
str_before_last_dot(before_last_dot), 4
str_before_nth, 17
str_can_be_numeric(can_be_numeric), 4
str_elem, 18
str_elems, 18
str_extract_currencies(currency_vec), 6
str_extract_currencies(currency), 5
str_extract_non_numerics
  (extract_non_numerics), 7
str_extract_numbers(extract_numbers), 7
str_first_currency(currency), 5
str_first_non_numeric
  (extract_non_numerics), 7
str_first_number(extract_numbers), 7
str_first_number_after_first
  (nth_number_after_mth), 12
str_first_number_after_last
  (nth_number_after_mth), 12
str_first_number_after_mth
  (nth_number_after_mth), 12
str_first_number_before_first
  (nth_number_before_mth), 12
str_first_number_before_last
  (nth_number_before_mth), 12
str_first_number_before_mth
  (nth_number_before_mth), 13
str_give_ext, 19
str_last_currency(currency), 5
str_last_non_numeric
  (extract_non_numerics), 7
str_last_number(extract_numbers), 7
str_last_number_after_first
  (nth_number_after_mth), 12
str_last_number_after_last
  (nth_number_after_mth), 12
str_last_number_after_mth
  (nth_number_after_mth), 12
str_last_number_before_first
  (nth_number_before_mth), 13
str_last_number_before_last
  (nth_number_before_mth), 13
str_last_number_before_mth
  (nth_number_before_mth), 13
str_locate_braces(locate_braces), 9
str_locate_first(str_locate_nth), 19
str_locate_last(str_locate_nth), 19
str_locate_nth, 19
str_match_arg(match_arg), 9
str_nice_nums, 20
str_nth_currency(currency), 5
str_nth_non_numeric
  (extract_non_numerics), 7
str_nth_number(extract_numbers), 7
str_nth_number_after_first
  (nth_number_after_mth), 12
str_nth_number_after_last
  (nth_number_after_mth), 12
str_nth_number_after_mth
  (nth_number_after_mth), 12
str_nth_number_before_first
  (nth_number_before_mth), 13
str_nth_number_before_last
  (nth_number_before_mth), 13
str_nth_number_before_mth
(nth_number_before_mth), 13
str_paste elems, 20
str_put_in_pos (put_in_pos), 14
str_remove_quoted, 21
str_singleize, 21
str_split_by_numbers
  (str_split_by_nums), 22
str_split_by_nums, 22
str_split_camel_case, 22
str_to_vec, 23
str_trim_anything, 23
strex::match_arg(), 9
strex::str_after_nth(), 17
strex::str_alphord_nums(), 20
strex::str_before_last_dot(), 4
strex::str_before_nth(), 17
strex::str_can_be_numeric(), 4
strex::str_elem(), 18
strex::str elems(), 18
strex::str_extract_currencies(), 5
strex::str_extract_non_numerics(), 7
strex::str_extract_numbers(), 7
strex::str_give_ext(), 19
strex::str_locate_braces(), 9
strex::str_locate_nth(), 19
strex::str_nth_number_after_mth(), 12
strex::str_nth_number_before_mth(), 13
strex::str_paste_elems(), 20
strex::str_remove_quoted(), 21
strex::str_singleize(), 21
strex::str_split_by_numbers(), 22
strex::str_split_camel_case(), 22
strex::str_to_vec(), 23
strex::str trim anything(), 23

to_vec (str_to_vec), 23
trim_anything (str trim anything), 23

unitize_dirs, 24