Stringr Explorer

Tweet-Driven Development for a Shiny App

Omayma Said

@OmaymaS

How do you find the **FUNCTION** you need?

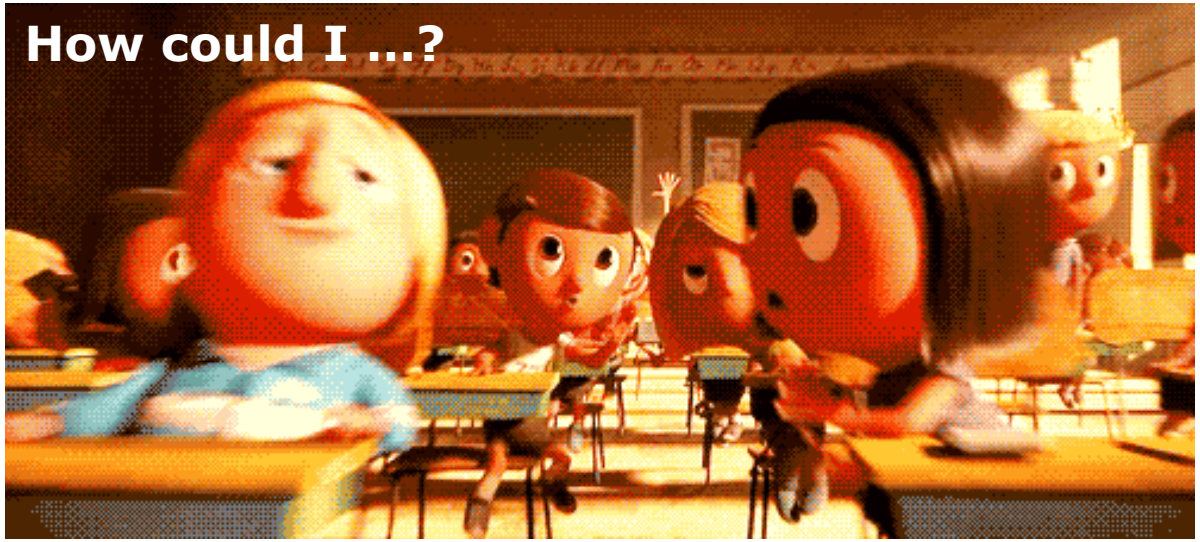
INSPIRATION

READ

The Documentation

ASK

A Question



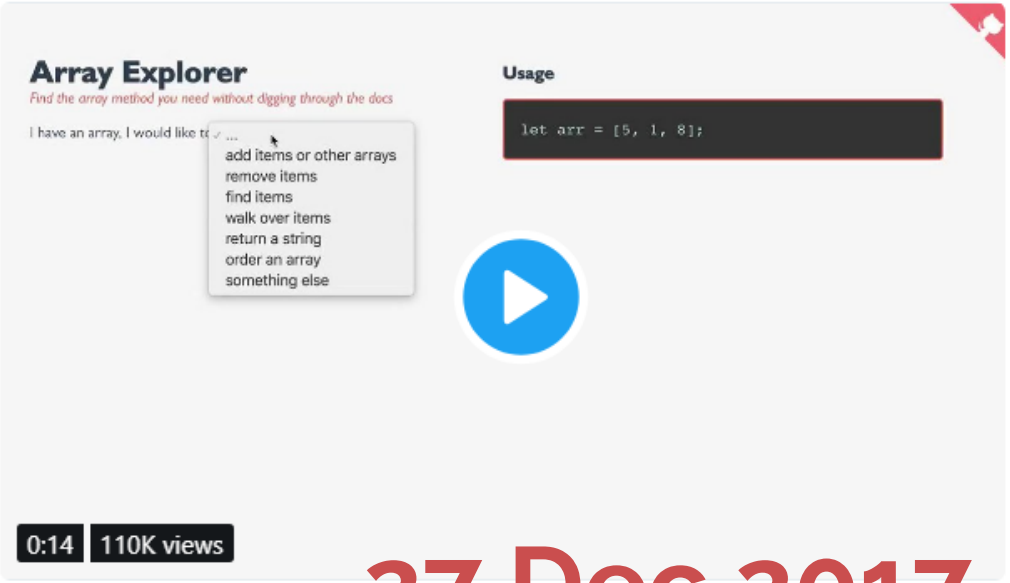
INSPIRATION

Sarah Drasner
@sarah_edo

Follow

🎁 A present for people learning JS! Finding the right array method can mean searching the docs one method at a time. I made this resource to help people quickly find what they need!

site: sdras.github.io/array-explorer/
codepen: codepen.io/sdras/full/gog...



7:53 PM - 27 Dec 2017

27 Dec 2017

INSPIRATION

Array Explorer

A PEN BY Sarah Drasner PRO

Fork | Change View | Log In | Sign Up

JavaScript Array Explorer

Find the array method you need without digging through the docs

I have an array, I would like to order an array

I need to reverse the order of the array

Array.reverse()

Reverses the order of the elements of an array in place — the first becomes the last, and the last becomes the first.

see the docs →

Usage

```
let arr = [5, 1, 8];
arr.reverse();
console.log(arr);
```

Output

```
[8, 1, 5]
```

Array Explorer

SUGGESTION (TWEET)

Emily Robinson
@robinson_es

Following

What an awesome idea. #rstats shiny app anyone? Maybe one for working with strings?

Sarah Drasner @sarah_edo
🎁 A present for people learning JS! Finding the right array method can mean searching the docs one method at a time. I made this resource to help people quickly find what they need!...

Show this thread

5:23 PM - 28 Dec 2017

28 Dec 2017

CONVERSATION

Omayma @OmaymaS_ · 28 Dec 2017

Replying to @robinson_es

You mean sth for stringr functions for instance, and regex?

1

28 Dec 2017

CONVERSATION

Omayma @OmaymaS_ · 28 Dec 2017

Replying to @robinson_es

You mean sth for stringr functions for instance, and regex?

1

Emily Robinson @robinson_es · 29 Dec 2017

Not sure about Regex, but yes thinking about the stringr functions replicating a lot what's in the cheat sheet

1

28 Dec 2017

29 Dec 2017

IMPLEMENTATION

Omayma @OmaymaS_ · 28 Dec 2017

Replying to @robinson_es

You mean sth for stringr functions for instance, and regex?

1

Emily Robinson @robinson_es · 29 Dec 2017

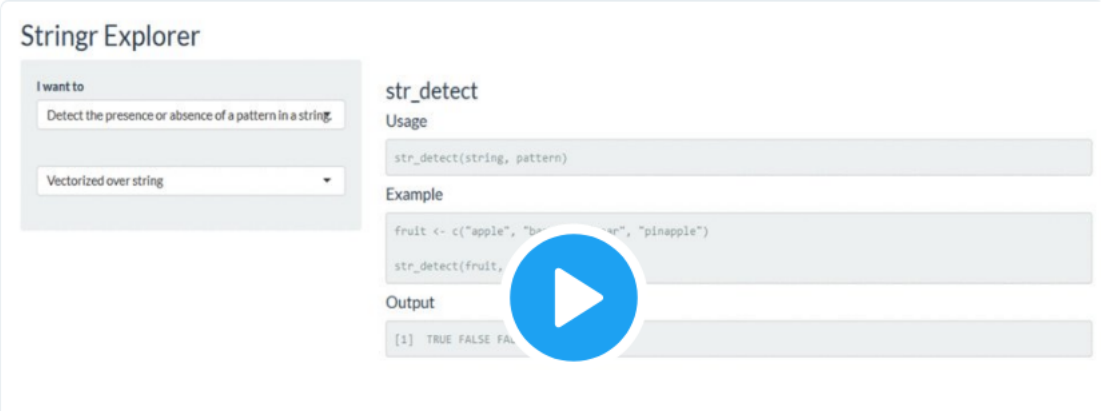
Not sure about Regex, but yes thinking about the stringr functions replicating a lot what's in the cheat sheet

1

Omayma @OmaymaS_ · 30 Dec 2017

I liked the idea, and thought about trying this simple Shiny App!

omaymas.shinyapps.io/stringr_help/



28 Dec 2017

29 Dec 2017

30 Dec 2017

IMPLEMENTATION

Stringr Explorer

Secure https://omaymas.shinyapps.io/stringr_help/

Stringr Explorer

I want to

Extract matching patterns from a string.

- Join multiple strings into a single string.
- Specify the encoding of a string.
- Count the number of matches in a string.
- Detect the presence or absence of a pattern in a string.
- Duplicate and concatenate strings within a character vector.
- Extract matching patterns from a string.
- The length of a string.

str_extract

Usage

```
str_extract(string, pattern)
str_extract_all(string, pattern, simplify = FALSE)
```

Example

```
shopping_list <- c("apples x4", "bag of flour",
                   "bag of sugar", "milk x2")
str_extract(shopping_list, "[a-z]+")
```

Output

```
[1] "apples" "bag"    "bag"    "milk"
```

R_Documentation

IMPLEMENTATION

Stringr Explorer

Secure | https://omaymas.shinyapps.io/stringr_help/

Stringr Explorer

I want to

Extract matching patterns from a string. 1

Extract all matches 2

str_extract_all 3

Usage

```
str_extract(string, pattern)
str_extract_all(string, pattern, simplify = FALSE)
```

4

Example

```
shopping_list <- c("apples x4", "bag of flour","bag of sugar", "milk x2")

str_extract_all(shopping_list, "[a-z]+")
```

5

Output

```
[[1]]
[1] "apples" "x"

[[2]]
[1] "bag"   "of"    "flour"

[[3]]
[1] "bag"   "of"    "sugar"

[[4]]
[1] "milk" "x"
```

R_Documentation 6

IMPLEMENTATION

Underlying Date

```
# A tibble: 30 x 6
   str_fn_names    str_fn_help str_fn_title          str_fn_usage example_title   example
   <chr>           <list>      <chr>                 <list>       <chr>           <chr>
 1 str_c           <S3: Rd>    Join multiple string~ <S3: glue>   ...             "str_c(\"Letter: \", letters)"
 2 str_conv        <S3: Rd>    Specify the encoding~ <S3: glue>   ...             "as.raw(177) %>% rawToChar %>% ~
 3 str_count       <S3: Rd>    Count the number of ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 4 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 5 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "fruit <- c(\"apple\", \"banana~
 6 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "str_detect(\"aecfg\", letters)"
 7 str_dup         <S3: Rd>    Duplicate and concat~ <S3: glue>   ...             "fruit <- c(\"apple\", \"pear\"~
 8 str_extract     <S3: Rd>    Extract matching pat~ <S3: glue>   Extract first ~ "shopping_list <- c(\"apples x4~
 9 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all ma~ "shopping_list <- c(\"apples x4~
10 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all wo~ "str_extract_all(\"This is, sup~
# ... with 20 more rows
```

Underlying Date

IMPLEMENTATION

Package Function Names

```
ls("package:stringr")
```

```
# A tibble: 30 x 6
   str_fn_names    str_fn_help str_fn_title          str_fn_usage example_title   example
   <chr>           <list>      <chr>                 <list>       <chr>           <chr>
 1 str_c           <S3: Rd>    Join multiple string~ <S3: glue>   ...             "str_c(\"Letter: \", letters)"
 2 str_conv        <S3: Rd>    Specify the encoding~ <S3: glue>   ...             "as.raw(177) %>% rawToChar %>% ~
 3 str_count       <S3: Rd>    Count the number of ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 4 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 5 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "fruit <- c(\"apple\", \"banana~
 6 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "str_detect(\"aecfg\", letters)"
 7 str_dup         <S3: Rd>    Duplicate and concat~ <S3: glue>   ...             "fruit <- c(\"apple\", \"pear\"~
 8 str_extract     <S3: Rd>    Extract matching pat~ <S3: glue>   Extract first ~ "shopping_list <- c(\"apples x4~
 9 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all ma~ "shopping_list <- c(\"apples x4~
10 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all wo~ "str_extract_all(\"This is, sup~
# ... with 20 more rows
```

More Details in the blog post

Underlying Date

IMPLEMENTATION

Functions Help Text
(Documentation)

`utils:::.getHelpFile()`

```
# A tibble: 30 x 6
   str_fn_names    str_fn_help str_fn_title          str_fn_usage example_title   example
   <chr>           <list>      <chr>                 <list>       <chr>           <chr>
 1 str_c           <S3: Rd>    Join multiple string~ <S3: glue>   ...             "str_c(\"Letter: \", letters)"
 2 str_conv        <S3: Rd>    Specify the encoding~ <S3: glue>   ...             "as.raw(177) %>% rawToChar %>% ~
 3 str_count       <S3: Rd>    Count the number of ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 4 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 5 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "fruit <- c(\"apple\", \"banana~
 6 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "str_detect(\"aecfg\", letters)"
 7 str_dup         <S3: Rd>    Duplicate and concat~ <S3: glue>   ...             "fruit <- c(\"apple\", \"pear\"~
 8 str_extract     <S3: Rd>    Extract matching pat~ <S3: glue>   Extract first ~ "shopping_list <- c(\"apples x4~
 9 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all ma~ "shopping_list <- c(\"apples x4~
10 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all wo~ "str_extract_all(\"This is, sup~
# ... with 20 more rows
```

More Details in the blog post

Underlying Date

IMPLEMENTATION

Functions Titles and Usage Text

`tools:::.Rd_get_metadata()`

```
# A tibble: 30 x 6
   str_fn_names    str_fn_help str_fn_title          str_fn_usage example_title   example
   <chr>           <list>      <chr>                 <list>       <chr>           <chr>
 1 str_c           <S3: Rd>    Join multiple string~ <S3: glue>   ...             "str_c(\"Letter: \", letters)"
 2 str_conv        <S3: Rd>    Specify the encoding~ <S3: glue>   ...             "as.raw(177) %>% rawToChar %>% ~
 3 str_count       <S3: Rd>    Count the number of ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 4 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   ...             "fruit <- c(\"apple\", \"banana~
 5 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "fruit <- c(\"apple\", \"banana~
 6 str_detect      <S3: Rd>    Detect the presence ~ <S3: glue>   Vectorized ove~ "str_detect(\"aecfg\", letters)"
 7 str_dup         <S3: Rd>    Duplicate and concat~ <S3: glue>   ...             "fruit <- c(\"apple\", \"pear\"~
 8 str_extract     <S3: Rd>    Extract matching pat~ <S3: glue>   Extract first ~ "shopping_list <- c(\"apples x4~
 9 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all ma~ "shopping_list <- c(\"apples x4~
10 str_extract_all <S3: Rd>    Extract matching pat~ <S3: glue>   Extract all wo~ "str_extract_all(\"This is, sup~
# ... with 20 more rows
```

More Details in the blog post

Underlying Date

IMPLEMENTATION

Package Function Names	`ls("package:stringr")`
Functions Help Text (Documentation)	`utils:::.getHelpFile()`
Functions Titles and Usage Text	`tools:::.Rd_get_metadata`
Examples	`copied`

More Details in the blog post

IMPLEMENTATION

Stringr Explorer

Secure | https://omaymas.shinyapps.io/stringr_help/

Stringr Explorer

I want to

Extract matching patterns from a string. 1

Extract all matches 2

str_extract_all 3

Usage

```
str_extract(string, pattern)
str_extract_all(string, pattern, simplify = FALSE)
```

4

Example

```
shopping_list <- c("apples x4", "bag of flour","bag of sugar", "milk x2")

str_extract_all(shopping_list, "[a-z]+")
```

5

Output

```
[[1]]
[1] "apples" "x"

[[2]]
[1] "bag"   "of"    "flour"

[[3]]
[1] "bag"   "of"    "sugar"

[[4]]
[1] "milk" "x"
```

R_Documentation 6

rud.is

"In God we trust. All others must bring data"

SODD — StackOverflow Driven-Development

posted in R, Stack Overflow on 2017-09-28 by hrbrmstr

@hrbrmstr

rud.is

"In God we trust. All others must bring data"

SODD — StackOverflow Driven-Development

posted in R, Stack Overflow on 2017-09-28 by hrbrmstr

I occasionally hang out on StackOverflow and often use an answer as an opportunity to fill a package void for a particular need. `docxtractr` and `qrencoder` are two (of many) packages that were birthed from SO answers. I usually try to answer with inline code first then expand the functionality into a package (if warranted). Some make it to CRAN (like those two), others stay on GitHub.

@hrbrmstr

Maëlle Salmon
@ma_salmon

"What about calling this **Tweet-Driven Development**"?

Stringr Explorer

Tweet-Driven Development for a Shiny App

Omayma Said

@OmaymaS

Blog Post

Shiny App