

# Package ‘tokenbrowser’

October 14, 2022

**Type** Package

**Title** Create Full Text Browsers from Annotated Token Lists

**Version** 0.1.5

**Author** Kasper Welbers and Wouter van Atteveldt

**Maintainer** Kasper Welbers <kasperwelbers@gmail.com>

**License** GPL-3

**Depends** R (>= 2.10)

**Imports** methods, Rcpp, stringi

**Suggests** testthat

**LinkingTo** Rcpp

**LazyData** true

**Description** Create browsers for reading full texts from a token list format.  
Information obtained from text analyses (e.g., topic modeling, word scaling)  
can be used to annotate the texts.

**SystemRequirements** C++11

**Encoding** UTF-8

**RoxygenNote** 7.1.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2020-10-16 06:50:02 UTC

## R topics documented:

add_tag . . . . .	2
attr_style . . . . .	3
categorical_browser . . . . .	4
category_highlight_tokens . . . . .	5
colourscaled_browser . . . . .	6
colourscale_tokens . . . . .	8
create_browser . . . . .	9

create_meta_tables . . . . .	11
highlighted_browser . . . . .	11
highlight_col . . . . .	13
highlight_tokens . . . . .	13
html_template . . . . .	15
rescale_var . . . . .	15
save_html . . . . .	16
scale_col . . . . .	16
set_col . . . . .	17
sotu_data . . . . .	18
sotu_lda . . . . .	18
tag_attr . . . . .	19
tag_tokens . . . . .	19
view_browser . . . . .	21
wrap_documents . . . . .	21

<b>Index</b>	<b>23</b>
--------------	-----------

---

add_tag	<i>Wrap values in an HTML tag</i>
---------	-----------------------------------

---

## Description

Wrap values in an HTML tag

## Usage

```
add_tag(
  x,
  tag,
  attr_str = NULL,
  ignore_na = F,
  span_adjacent = F,
  doc_id = NULL
)
```

## Arguments

x	a vector of values to be wrapped in a tag
tag	A character vector of length 1, specifying the html tag (e.g., "div", "h1", "span")
attr_str	A character string of the same length as x (or of length 1).
ignore_na	If TRUE, do not add tag if value is NA
span_adjacent	If TRUE, include adjacent tokens with identical attr_str within the same tag
doc_id	If span_adjacent is TRUE, The document ids are required to ensure that tags do not span from one document to another.

**Value**

a character vector

**Examples**

```
x = c("Obama", "Bush")
add_tag(x, 'span')

## add attributes with the tag_attr function
add_tag(x, 'span',
        tag_attr(class = "president"))

## add style attributes with the attr_style function within tag_attr
add_tag(x, 'span',
        tag_attr(class = "president",
                 style = attr_style(`background-color` = 'rgba(255, 255, 0, 1)')))
```

---

attr\_style

*Create the content of the html style attribute*

---

**Description**

Designed to be used together with the tag\_attr function.

**Usage**

```
attr_style(...)
```

**Arguments**

...            named arguments are used as settings in the html style attribute, with the name being the name of the setting (e.g., background-color). All arguments must be vectors of the same length. NA values can be used to ignore a setting, and if all settings are NA then NA is returned (instead of an empty string for style settings).

**Value**

a character vector with the content of the html style attribute

**Examples**

```
tag_attr(class = c('x', 'y'),
        style = attr_style(`background-color` = 'rgba(255, 255, 0, 1)'))
```

---

categorical_browser	<i>Convert tokens into full texts in an HTML file with category highlighting</i>
---------------------	--

---

### Description

Convert tokens into full texts in an HTML file with category highlighting

### Usage

```
categorical_browser(
  tokens,
  category,
  alpha = 0.3,
  labels = NULL,
  meta = NULL,
  colors = NULL,
  doc_col = "doc_id",
  token_col = "token",
  filename = NULL,
  unfold = NULL,
  span_adjacent = T,
  ...
)
```

### Arguments

tokens	A data.frame with a column for document ids (doc_col) and a column for tokens (token_col)
category	Either a numeric vector with values representing categories, or a factor vector, in which case the values are used as labels. If a numeric vector is used, the labels can also be specified in the labels argument
alpha	Optionally, the alpha (transparency) can be specified, with 0 being fully transparent and 1 being fully colored. This can be a vector to specify a different alpha for each value.
labels	A character vector giving names to the unique category values. If category is a factor vector, the factor levels are used.
meta	A data.frame with a column for document_ids (doc_col). All other columns are added to the browser as document meta.
colors	A character vector with color names for unique values of the category argument. Has to be the same length as unique(na.omit(category))
doc_col	The name of the document id column
token_col	The name of the token column
filename	Name of the output file. Default is temp file

unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. list(doc_id = 1, sentence = 1) will be [doc_id = 1, sentence = 2].
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
...	Additional formatting arguments passed to create_browser()

**Value**

The name of the file where the browser is saved. Can be opened conveniently from within R using `browseUrl()`

**Examples**

```
## as an example, use simple grep to code tokens
code = rep(NA, nrow(sotu_data$tokens))
code[grep('war', sotu_data$tokens$token)] = 'War'
code[grep('mother|father|child', sotu_data$tokens$token)] = 'Family'
code = as.factor(code)
url = categorical_browser(sotu_data$tokens, category=code, meta=sotu_data$meta)

view_browser(url) ## view browser in the Viewer

if (interactive()) {
  browseURL(url) ## view in default webbrowser
}
```

---

```
category_highlight_tokens
  Highlight tokens per category
```

---

**Description**

This is a convenience wrapper for `tag_tokens()` that can be used if tokens need to be colored per category

**Usage**

```
category_highlight_tokens(
  tokens,
  category,
  labels = NULL,
  alpha = 0.4,
  class = NULL,
  colors = NULL,
  unfold = NULL,
  span_adjacent = F,
```

```

    doc_id = NULL
  )

```

### Arguments

tokens	A character vector of tokens
category	Either a factor, or a numeric vector with values representing category indices. If a numeric vector is used, labels must also be given
labels	A character vector with labels for the categories
alpha	Optionally, the alpha (transparency) can be specified, with 0 being fully transparent and 1 being fully colored. This can be a vector to specify a different alpha for each value.
class	Optionally, a character vector of the class to add to the span tags. If NA no class is added
colors	A character vector with color names for unique values of the value argument. Has to be the same length as <code>unique(na.omit(category))</code>
unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. <code>list(doc_id = 1, sentence = 1)</code> will be <code>[doc_id = 1, sentence = 2]</code> . This only works if the tagged tokens are used in the html browser created with the <a href="#">create_browser</a> function (as it relies on javascript).
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
doc_id	If span_adjacent is TRUE, The document ids are required to ensure that tags do not span from one document to another.

### Value

a character vector of color-tagged tokens

### Examples

```

tokens = c('token_1', 'token_2', 'token_3', 'token_4')
category = c('a', 'a', NA, 'b')
category_highlight_tokens(tokens, category)

```

---

colourscaled_browser	<i>Convert tokens into full texts in an HTML file with color ramp highlighting</i>
----------------------	--

---

### Description

Convert tokens into full texts in an HTML file with color ramp highlighting

**Usage**

```
colourscaled_browser(
  tokens,
  value,
  alpha = 0.4,
  meta = NULL,
  col_range = c("red", "blue"),
  doc_col = "doc_id",
  token_col = "token",
  doc_nav = NULL,
  token_nav = NULL,
  filename = NULL,
  unfold = NULL,
  span_adjacent = T,
  ...
)
```

**Arguments**

tokens	A data.frame with a column for document ids (doc_col) and a column for tokens (token_col)
value	A numeric vector with values between -1 and 1. Determines the color mixture of the scale colors specified in col_range
alpha	Optionally, the alpha (transparency) can be specified, with 0 being fully transparent and 1 being fully colored. This can be a vector to specify a different alpha for each value.
meta	A data.frame with a column for document_ids (doc_col). All other columns are added to the browser as document meta
col_range	The color used to highlight
doc_col	The name of the document id column
token_col	The name of the token column
doc_nav	The name of a column in meta, used to set a navigation tag
token_nav	Alternative to doc_nav, a column in the tokens, used to set a navigation tag
filename	Name of the output file. Default is temp file
unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. list(doc_id = 1, sentence = 1) will be [doc_id = 1, sentence = 2].
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
...	Additional formatting arguments passed to create_browser()

**Value**

The name of the file where the browser is saved. Can be opened conveniently from within R using `browseUrl()`

**Examples**

```
## as an example, scale word colors based on number of characters
scale = nchar(as.character(sotu_data$tokens$token))
scale[scale>6] = scale[scale>6] +20
scale = rescale_var(sqrt(scale), -1, 1)
scale[abs(scale) < 0.5] = NA
url = colorscaled_browser(sotu_data$tokens, value = scale, meta=sotu_data$meta)

view_browser(url) ## view browser in the Viewer

if (interactive()) {
browseURL(url) ## view in default webbrowser
}
```

---

colorscale\_tokens      *Color tokens using colorRamp*

---

**Description**

This is a convenience wrapper for `tag_tokens()` that can be used if tokens only need to be colored.

**Usage**

```
colorscale_tokens(
  tokens,
  value,
  alpha = 0.4,
  class = NULL,
  col_range = c("red", "blue"),
  unfold = NULL,
  span_adjacent = F,
  doc_id = NULL
)
```

**Arguments**

tokens	A character vector of tokens
value	A numeric vector with values between -1 and 1. Determines the color mixture of the scale colors specified in <code>col_range</code>
alpha	Optionally, the alpha (transparency) can be specified, with 0 being fully transparent and 1 being fully colored. This can be a vector to specify a different alpha for each value.
class	Optionally, a character vector of the class to add to the span tags. If NA no class is added
col_range	The colors used in the scale ramp.



unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. list(doc_id = 1, sentence = 1) will be [doc_id = 1, sentence = 2]. This only works if the tagged tokens are used in the html browser created with the <code>create_browser</code> function (as it relies on javascript).
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
doc_id	If span_adjacent is TRUE, The document ids are required to ensure that tags do not span from one document to another.

**Value**

a character vector of color-tagged tokens

**Examples**

```
colorscale_tokens(c('token_1', 'token_2', 'token_3'),
                 value = c(-1, 0, 1))
```

---

create_browser	<i>Convert tokens into full texts in an HTML file</i>
----------------	---

---

**Description**

Convert tokens into full texts in an HTML file

**Usage**

```
create_browser(
  tokens,
  meta = NULL,
  doc_col = "doc_id",
  token_col = "token",
  space_col = NULL,
  doc_nav = NULL,
  token_nav = NULL,
  filename = NULL,
  css_str = NULL,
  header = "",
  subheader = "",
  n = TRUE,
  navfilter = TRUE,
  top_nav = NULL,
  thres_nav = 1,
  colors = NULL,
  style_col1 = "#7D1935",
  style_col2 = "#F5F3EE"
)
```

**Arguments**

tokens	A data.frame with a column for document ids (doc_col) and a column for tokens (token_col)
meta	A data.frame with a column for document_ids (doc_col). All other columns are added to the browser as document meta
doc_col	The name of the document id column
token_col	The name of the token column
space_col	Optionally, a column with space indications (" ", "\n", etc.) per token (which is how some NLP parsers indicate spaces)
doc_nav	The name of a column (factor or character) in meta, used to create a navigation bar for selecting document groups.
token_nav	Alternative to doc_nav, a column in the tokens. Navigation filters will then be used to select documents in which the value occurs at least once.
filename	Name of the output file. Default is temp file
css_str	A character string, to be directly added to the css style header
header	Optionally, specify the header
subheader	Optionally, specify a subheader
n	If TRUE, report N in header
navfilter	If TRUE (default) enable filtering with nav(igation) bar.
top_nav	A number. If token_nav is used, navigation filters will only apply to the top x values with highest token occurrence in a document
thres_nav	Like top_nav, but specifying a threshold for the minimum number of tokens.
colors	Optionally, a vector with color names for the navigation bar. Length has to be identical to unique non-NA items in the navigation.
style_col1	Color of the browser header
style_col2	Color of the browser background

**Value**

The name of the file where the browser is saved. Can be opened conveniently from within R using `browseUrl()`

**Examples**

```
url = create_browser(sotu_data$tokens, sotu_data$meta, token_col = 'token', header = 'Speeches')

view_browser(url) ## view browser in the Viewer

if (interactive()) {
  browseURL(url) ## view in default webbrowser
}
```

---

create_meta_tables	<i>HTML tables for meta data per document</i>
--------------------	---

---

**Description**

Each row of the data.frame is transformed into a html table with two columns: name and value. The columnnames of meta are used as names.

**Usage**

```
create_meta_tables(meta, ignore_col = NULL)
```

**Arguments**

meta	a data.frame where each row represents the meta data for a document
ignore_col	optionally, a character vector with names of metadata columns to ignore

**Value**

a character vector where each value contains a string for an html table.

**Examples**

```
tabs = create_meta_tables(sotu_data$meta)
tabs[1]
```

---

highlighted_browser	<i>Convert tokens into full texts in an HTML file with highlighted tokens</i>
---------------------	---

---

**Description**

Convert tokens into full texts in an HTML file with highlighted tokens

**Usage**

```
highlighted_browser(  
  tokens,  
  value,  
  meta = NULL,  
  col = "yellow",  
  doc_col = "doc_id",  
  token_col = "token",  
  doc_nav = NULL,  
  token_nav = NULL,  
  filename = NULL,  
  unfold = NULL,
```

```

    span_adjacent = T,
    ...
  )

```

### Arguments

tokens	A data.frame with a column for document ids (doc_col) and a column for tokens (token_col)
value	Either a logical vector or a numeric vector with values between 0 and 1. If a logical vector is used, then tokens with TRUE will be highlighted (with the color specified in pos_col). If a numeric vector is used, the value determines the alpha (transparency), with 0 being fully transparent and 1 being fully colored.
meta	A data.frame with a column for document_ids (doc_col). All other columns are added to the browser as document meta
col	The color used to highlight
doc_col	The name of the document id column
token_col	The name of the token column
doc_nav	The name of a column in meta, used to set a navigation tag
token_nav	Alternative to doc_nav, a column in the tokens, used to set a navigation tag
filename	Name of the output file. Default is temp file
unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. list(doc_id = 1, sentence = 1) will be [doc_id = 1, sentence = 2].
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
...	Additional formatting arguments passed to create_browser()

### Value

The name of the file where the browser is saved. Can be opened conveniently from within R using `browseUrl()`

### Examples

```

## as an example, highlight words based on word length
highlight = nchar(as.character(sotu_data$tokens$token))
highlight = highlight / max(highlight)
highlight[highlight < 0.3] = NA
url = highlighted_browser(sotu_data$tokens, value = highlight, sotu_data$meta)

view_browser(url) ## view browser in the Viewer

if (interactive()) {
  browseURL(url) ## view in default webbrowser
}

```

---

highlight_col	<i>Create a highlight color for a html style attribute</i>
---------------	--

---

**Description**

Designed to be used together with the `attr_style` function. The return value can directly be used to set the color in an html tag attribute (e.g., `color`, `background-color`)

**Usage**

```
highlight_col(value, col = "yellow")
```

**Arguments**

value	Either a logical vector or a numeric vector with values between 0 and 1. If a logical vector is used, then tokens with TRUE will be highlighted (with the color specified in <code>pos_col</code> ). If a numeric vector is used, the value determines the alpha (transparency), with 0 being fully transparent and 1 being fully colored.
col	The color used to highlight

**Value**

The string used to specify a color in an html tag attribute

**Examples**

```
highlight_col(c(NA, 0, 0.1, 0.5, 1))

## used in combination with attr_style()
attr_style(color = highlight_col(c(NA, 0, 0.1, 0.5, 1)))

## note that for background-color you need inversed quotes to deal
## with the hyphen in an argument name
attr_style(`background-color` = highlight_col(c(NA, 0, 0.1, 0.5, 1)))

tag_attr(class = c(1, 2),
         style = attr_style(`background-color` = highlight_col(c(FALSE, TRUE))))
```

---

highlight_tokens	<i>Highlight tokens</i>
------------------	-------------------------

---

**Description**

This is a convenience wrapper for `tag_tokens()` that can be used if tokens only need to be colored.

**Usage**

```
highlight_tokens(
  tokens,
  value,
  class = NULL,
  col = "yellow",
  unfold = NULL,
  span_adjacent = F,
  doc_id = NULL
)
```

**Arguments**

tokens	A character vector of tokens
value	Either a logical vector or a numeric vector with values between 0 and 1. If a logical vector is used, then tokens with TRUE will be highlighted (with the color specified in pos_col). If a numeric vector is used, the value determines the alpha (transparency), with 0 being fully transparent and 1 being fully colored.
class	Optionally, a character vector of the class to add to the span tags. If NA no class is added
col	The color used to highlight
unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. list(doc_id = 1, sentence = 1) will be [doc_id = 1, sentence = 2]. This only works if the tagged tokens are used in the html browser created with the <a href="#">create_browser</a> function (as it relies on javascript).
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
doc_id	If span_adjacent is TRUE, The document ids are required to ensure that tags do not span from one document to another.

**Value**

a character vector of color-tagged tokens

**Examples**

```
highlight_tokens(c('token_1', 'token_2', 'token_3'),
  value = c(FALSE, FALSE, TRUE))
```

```
highlight_tokens(c('token_1', 'token_2', 'token_3'),
  value = c(0, 0.3, 0.6))
```

---

html_template	<i>create the html template</i>
---------------	---------------------------------

---

**Description**

create the html template

**Usage**

```
html_template(template, css_str = NULL, col1 = "#7D1935", col2 = "#F5F3EE")
```

**Arguments**

template	The name of the template to be used
css_str	A character string, to be directly added to the css style header
col1	The first style color (top bar color)
col2	The second style color (background color)

**Value**

A list with the html header and footer

---

rescale_var	<i>Rescale a numeric variable</i>
-------------	-----------------------------------

---

**Description**

Rescale a numeric variable

**Usage**

```
rescale_var(x, new_min = 0, new_max = 1, x_min = min(x), x_max = max(x))
```

**Arguments**

x	a numeric vector
new_min	The minimum value of the output
new_max	The maximum value of the output
x_min	The lowest possible value in x. By default this is the actual lowest value in x.
x_max	The highest possible value in x. By default this is the actual highest value in x.

**Value**

a numeric vector

**Examples**

```
rescale_var(1:10)
rescale_var(1:10, new_min = -1, new_max = 1)
```

---

save_html	<i>Wrap html body in the template and save</i>
-----------	--

---

**Description**

Wrap html body in the template and save

**Usage**

```
save_html(data, template, filename = NULL)
```

**Arguments**

data	The html body data
template	The html header/footer template
filename	The name of the file to save the html. Default is a temp file

**Value**

The (local) url to the html file

---

scale_col	<i>Create a scale color for a html style attribute</i>
-----------	--

---

**Description**

Designed to be used together with the attr\_style function. The return value can directly be used to set the color in an html tag attribute (e.g., color, background-color)

**Usage**

```
scale_col(value, alpha = 1, col_range = c("red", "blue"))
```

**Arguments**

value	A numeric vector with values between -1 and 1. Determines the color mixture of the scale colors specified in col_range
alpha	Optionally, the alpha (transparency) can be specified, with 0 being fully transparent and 1 being fully colored. This can be a vector to specify a different alpha for each value.
col_range	The colors used in the scale.



**Value**

The string used to specify a color in a html tag attribute

**Examples**

```
scale_col(c(NA, -1, 0, 0.5, 1))

## used in combination with attr_style()
attr_style(color = scale_col(c(NA, -1, 0, 0.5, 1)))

## note that for background-color you need inversed
## quotes to deal with the hyphen in an argument name
attr_style(`background-color` = scale_col(c(NA, -1, 0, 0.5, 1)))

tag_attr(class = c(1, 2),
         style = attr_style(`background-color` = scale_col(c(-1,1))))
```

---

 set\_col

*Transpose a color into the string format used in html attributes*

---

**Description**

Transpose a color into the string format used in html attributes

**Usage**

```
set_col(col, alpha = 1)
```

**Arguments**

col	The name of the color
alpha	Optionally, the alpha (transparency), with 0 being fully transparent and 1 being fully colorized.

**Value**

The string used to specify a color in an html tag attribute

**Examples**

```
set_col('red')
set_col('red', alpha=0.5)
```

---

sotu_data	<i>Tokens from Bush and Obamas State of the Union addresses</i>
-----------	---

---

**Description**

Tokens from Bush and Obamas State of the Union addresses

**Usage**

```
data(sotu_data)
```

**Format**

sotu\_data: A data.frame with tokens and a data.frame with meta data

---

sotu_lda	<i>Word assignments, docXtopic matrix and topicXword matrix of an LDA model of the SOTU data</i>
----------	--

---

**Description**

Word assignments, docXtopic matrix and topicXword matrix of an LDA model of the SOTU data

**Usage**

```
data(sotu_lda)
```

**Format**

sotu\_lda: Word assignments is a data.frame with document, lemma and topic columns. topic\_word\_mat and doc\_topic\_mat are matrices

---

tag_attr	<i>create attribute string for html tags</i>
----------	--

---

**Description**

create attribute string for html tags

**Usage**

```
tag_attr(...)
```

**Arguments**

...            named arguments are used as attributes, with the name being the name of the attribute (e.g., class, style). All argument must be vectors of the same length, or length 1 (used as a constant). NA values can be used to skip an attribute. If all attributes are NA, an NA is returned

**Value**

a character vector with attribute strings. Designed to be usable as the attr\_str in add\_tag(). If ... is empty, NA is returned

**Examples**

```
add_tag('TEXT', 'span')
add_tag('TEXT', 'span', tag_attr(class='CLASS'))
```

---

tag_tokens	<i>add span tags to tokens</i>
------------	--------------------------------

---

**Description**

This is the main function for adding colors, onclick effects, etc. to tokens, for which <span> tags are used. The named arguments are used to set the attributes.

**Usage**

```
tag_tokens(
  tokens,
  tag = "span",
  span_adjacent = F,
  doc_id = NULL,
  unfold = NULL,
  ...
)
```

**Arguments**

tokens	a vector of tokens.
tag	The name of the tag to be used
span_adjacent	If TRUE, include adjacent tokens with identical attributes within the same tag
doc_id	If span_adjacent is TRUE, The document ids are required to ensure that tags do not span from one document to another.
unfold	Either a character vector or a named list of vectors of the same length as tokens. If given, all tokens with a tag can be clicked on to unfold the given text. If a list of vectors is given, the values of the columns are concatenated with the column name. E.g. list(doc_id = 1, sentence = 1) will be [doc_id = 1, sentence = 2]. This only works if the tagged tokens are used in the html browser created with the <a href="#">create_browser</a> function (as it relies on javascript).
...	named arguments are used as attributes in the span tag for each token, with the name being the name of the attribute (e.g., class, . Each argument must be a vector of the same length as the number of tokens. NA values can be used to ignore attribute for a token, and if a token has NA for each attribute, it is not given a span tag.

**Details**

If a token does not have any attributes, the <span> tag is not added.

Note that the `attr_style()` function can be used to conveniently set the style attribute. Also, the `set_col()`, `highlight_col()` and `scale_col()` functions can be used to set the color of style attributes. See the example for illustration.

**Value**

a character vector of tagged tokens

**Examples**

```
tag_tokens(tokens = c('token_1', 'token_2', 'token_3'),
           class = c(1,1,2),
           style = attr_style(color = set_col('red'),
                             `background-color` = highlight_col(c(FALSE,FALSE,TRUE))))

## tokens without attributes are not given a span tag
tag_tokens(tokens = c('token_1', 'token_2', 'token_3'),
           class = c(1,NA,NA),
           style = attr_style(color = highlight_col(c(TRUE,TRUE,FALSE))))

## span_adjacent can be used to put tokens with identical tags within one tag
## but then a doc_id has to be given as well
tag_tokens(tokens = c('token_1', 'token_2', 'token_3'),
           class = c(1,1,NA),
           span_adjacent=TRUE,
           doc_id = c(1,1,1))
```

---

view_browser	<i>View a browser (HTML) in the R viewer</i>
--------------	--

---

**Description**

View a browser (HTML) in the R viewer

**Usage**

```
view_browser(url)
```

**Arguments**

url                    An URL, created with \*\_browser

**Examples**

```
url = create_browser(sotu_data$tokens, sotu_data$meta, token_col = 'token', header = 'Speeches')  
  
## the url  
  
view_browser(url)    ## view browser in the Viewer
```

---

wrap_documents	<i>Wrap tokens into document html strings</i>
----------------	---

---

**Description**

Pastes the tokens into articles, and returns an <article> html element.

**Usage**

```
wrap_documents(  
  tokens,  
  meta,  
  doc_col = "doc_id",  
  token_col = "token",  
  space_col = NULL,  
  nav = doc_col,  
  token_nav = NULL,  
  top_nav = NULL,  
  thres_nav = NULL  
)
```

**Arguments**

tokens	A data.frame with a column for document ids (doc_col) and a column for tokens (token_col)
meta	A data.frame with a column for document_ids (doc_col). All other columns are added to the browser as document meta
doc_col	The name of the document id column
token_col	The name of the token column
space_col	Optionally, a column with space indications (e.g., newline) per token (which is how some NLP parsers indicate spaces)
nav	The column in meta used for nav. Defaults to 'doc_id'
token_nav	Alternative to nav (which uses meta), a column in tokens used for navigation
top_nav	If token_nav is used, navigation filters will only apply to the top x values with highest token occurrence in a document
thres_nav	Like top_nav, but specifying a threshold for the minimum number of tokens.

**Value**

A named vector, with document ids as names and the document html strings as values

**Examples**

```
docs = wrap_documents(sotu_data$tokens, sotu_data$meta)
head(names(docs))
docs[[1]]
```

# Index

## \* datasets

sotu\_data, 18

sotu\_lda, 18

add\_tag, 2

attr\_style, 3

categorical\_browser, 4

category\_highlight\_tokens, 5

colorscale\_tokens, 8

colored\_scaled\_browser, 6

create\_browser, 6, 9, 9, 14, 20

create\_meta\_tables, 11

highlight\_col, 13

highlight\_tokens, 13

highlighted\_browser, 11

html\_template, 15

rescale\_var, 15

save\_html, 16

scale\_col, 16

set\_col, 17

sotu\_data, 18

sotu\_lda, 18

tag\_attr, 19

tag\_tokens, 19

view\_browser, 21

wrap\_documents, 21