

Package ‘ready4fun’

September 8, 2020

Title Readyforwhatsnext Function Authoring And Documentation
Tools

Version 0.0.0.9146

Description ready4fun is a collection of functions for
authoring code libraries of functions and datasets for use in mental
health simulations developed within the readyforwhatsnext
ecosystem.

License FALSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Depends R (>= 2.10)

Imports methods,
Hmisc,
devtools,
dplyr,
magrittr,
purrr,
readxl,
rlang,
sinew,
stats,
stringi,
stringr,
testit,
tibble,
tools,
usethis,
xfun

R topics documented:

ready4fun-package	2
abbreviations_lup	3
fn_type_lup_tb	3
get_from_lup_obj	4

import_xls_sheets	5
make_dmt_for_all_fns	5
make_fn_dmt_tbl	6
make_fn_type_lup	7
make_lines_for_fn_dmt	8
object_type_lup	9
read_fns	9
rowbind_all_tbs_in_r4_obj	10
write_abbr_lup	10
write_all_tbs_in_tbs_r4_to_csvs	11
write_and_doc_ds	12
write_and_doc_fn_fls	13
write_dmt_d_fn_type_lup	13
write_documented_fns	14
write_fn_type_dirs	15
write_nsimps_to_desc	15
write_pkg	16
write_pkg_setup_fls	16
write_pt_lup_db	17
write_std_imp	17
write_tb_to_csv	18
write_to_reset_pkg_files	18
write_ws	19
%>%	19

Index 20

ready4fun-package	<i>ready4fun: Readyforwhatsnext Function Authoring And Documentation Tools</i>
-------------------	--

Description

ready4fun is a collection of functions for authoring code libraries of functions and datasets for use in mental health simulations developed within the readyforwhatsnext ecosystem.

Details

To learn more about ready4fun, start with the vignettes: `browseVignettes(package = "ready4fun")`

Author(s)

Maintainer: Matthew Hamilton <matthew.hamilton@orygen.org.au> ([ORCID](#))

Authors:

- Glen Wiesner <Glen.Wiesner@vu.edu.au> ([ORCID](#))

Other contributors:

- Orygen [copyright holder, funder]
- VicHealth [funder]
- Victoria University [funder]

abbreviations_lup	<i>Common abbreviations lookup table</i>
-------------------	--

Description

A lookup table for abbreviations commonly used in object names in the ready4funpackage.

Usage

```
abbreviations_lup
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 198 rows and 3 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

plural_lgl Plural (a logical vector)

Source

<https://readyforwhatsnext.github.io/readyforwhatsnext/>

fn_type_lup_tb	<i>Function type lookup table</i>
----------------	-----------------------------------

Description

A lookup table to find descriptions for different types of functions used within the ready4fun package suite.

Usage

```
fn_type_lup_tb
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 16 rows and 6 columns.

Details

A tibble

fn_type_nm_chr Function type name (a character vector)

fn_type_desc_chr Function type description (a character vector)

first_arg_desc_chr First argument description (a character vector)

second_arg_desc_chr Second argument description (a character vector)

is_generic_lgl Is generic (a logical vector)

is_method_lgl Is method (a logical vector)

Source

<https://readyforwhatsnext.github.io/readyforwhatsnext/>

get_from_lup_obj	<i>Get from lookup table object</i>
------------------	-------------------------------------

Description

get_from_lup_obj() is a Get function that retrieves a pre-existing data object from memory, local file system or online repository. Specifically, this function implements an algorithm to get from lookup table object. Function argument data_lookup_tb specifies the where to look for the required object. The function returns Return object (an output object of multiple potential types).

Usage

```
get_from_lup_obj(
  data_lookup_tb,
  match_value_xx,
  match_var_nm_1L_chr,
  target_var_nm_1L_chr,
  evaluate_lgl = TRUE
)
```

Arguments

data_lookup_tb Data lookup (a tibble)

match_value_xx Match value (an output object of multiple potential types)

match_var_nm_1L_chr

Match var name (a character vector of length one)

target_var_nm_1L_chr

Target var name (a character vector of length one)

evaluate_lgl Evaluate (a logical vector), Default: TRUE

Value

Return object (an output object of multiple potential types)

import_xls_sheets	<i>Import Excel workbook sheets</i>
-------------------	-------------------------------------

Description

import_xls_sheets() is an Import function that reads a data object in its native format and converts it to an R object. Specifically, this function implements an algorithm to import excel workbook sheets. The function returns Tibble list (a list of tibbles).

Usage

```
import_xls_sheets(range_1L_chr, sheet_names_chr, path_1L_chr)
```

Arguments

range_1L_chr	Range (a character vector of length one)
sheet_names_chr	Sheet names (a character vector)
path_1L_chr	Path (a character vector of length one)

Value

Tibble list (a list of tibbles)

make_dmt_for_all_fns	<i>Make documentation for all functions</i>
----------------------	---

Description

make_dmt_for_all_fns() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make documentation for all functions. The function returns All functions documentation (a tibble).

Usage

```
make_dmt_for_all_fns(
  paths_ls = make_fn_nms(),
  undocumented_fns_dir_chr = make_undmtd_fns_dir_chr(),
  custom_dmt_ls = list(details_ls = NULL, inc_for_main_user_lgl_ls =
    list(force_true_chr = NA_character_, force_false_chr = NA_character_), args_ls_ls =
    NULL),
  fn_type_lup_tb,
  abbreviations_lup = NULL
)
```

Arguments

paths_ls Paths (a list), Default: make_fn_nms()
 undocumented_fns_dir_chr Undocumented functions directory (a character vector), Default: make_undmtd_fns_dir_chr()
 custom_dmt_ls Custom documentation (a list), Default: list(details_ls = NULL, inc_for_main_user_lgl_ls = list(force_true_chr = NA_character_, force_false_chr = NA_character_), args_ls_ls = NULL)
 fn_type_lup_tb Function type lookup table (a tibble)
 abbreviations_lup Abbreviations (a lookup table), Default: NULL

Value

All functions documentation (a tibble)

make_fn_dmt_tbl	<i>Make function documentation table</i>
-----------------	--

Description

make_fn_dmt_tbl() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function documentation table. The function returns Function documentation table (a tibble).

Usage

```
make_fn_dmt_tbl(
  fns_path_chr,
  fns_dir_chr = make_undmtd_fns_dir_chr(),
  pkg_nm_1L_chr,
  custom_dmt_ls = list(title_ls = NULL, desc_ls = NULL, details_ls = NULL,
    inc_for_main_user_lgl_ls = NULL, output_ls = NULL, example_ls = NULL, args_ls_ls =
    NULL),
  append_1L_lgl = T,
  fn_type_lup_tb = NULL,
  abbreviations_lup = NULL,
  object_type_lup = NULL,
  test_for_write_R_warning_fn = NULL
)
```

Arguments

fns_path_chr Functions path (a character vector)
 fns_dir_chr Functions directory (a character vector), Default: make_undmtd_fns_dir_chr()
 pkg_nm_1L_chr Package name (a character vector of length one)
 custom_dmt_ls Custom documentation (a list), Default: list(title_ls = NULL, desc_ls = NULL, details_ls = NULL, inc_for_main_user_lgl_ls = NULL, output_ls = NULL, example_ls = NULL, args_ls_ls = NULL)
 append_1L_lgl Append (a logical vector of length one), Default: T

fn_type_lup_tb Function type lookup table (a tibble), Default: NULL
 abbreviations_lup
 Abbreviations (a lookup table), Default: NULL
 object_type_lup
 Object type (a lookup table), Default: NULL
 test_for_write_R_warning_fn
 Test for write warning (a function), Default: NULL

Value

Function documentation table (a tibble)

make_fn_type_lup	<i>Make function type</i>
------------------	---------------------------

Description

make_fn_type_lup() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make function type lookup table. The function returns Function type lookup table (a tibble).

Usage

```
make_fn_type_lup(
  fn_type_nm_chr = character(0),
  fn_type_desc_chr = character(0),
  first_arg_desc_chr = character(0),
  second_arg_desc_chr = character(0),
  is_generic_lgl = logical(0),
  is_method_lgl = logical(0)
)
```

Arguments

fn_type_nm_chr Function type name (a character vector), Default: character(0)
 fn_type_desc_chr
 Function type description (a character vector), Default: character(0)
 first_arg_desc_chr
 First argument description (a character vector), Default: character(0)
 second_arg_desc_chr
 Second argument description (a character vector), Default: character(0)
 is_generic_lgl Is generic (a logical vector), Default: logical(0)
 is_method_lgl Is method (a logical vector), Default: logical(0)

Value

Function type lookup table (a tibble)

make_lines_for_fn_dmt *Make lines for function documentation*

Description

make_lines_for_fn_dmt() is a Make function that creates a new R object. Specifically, this function implements an algorithm to make lines for function documentation. The function is called for its side effects and does not return a value.

Usage

```
make_lines_for_fn_dmt(
  fn_name_1L_chr,
  fn_type_1L_chr,
  fn = NULL,
  fn_desc_1L_chr = NA_character_,
  fn_out_type_1L_chr = NA_character_,
  fn_title_1L_chr = NA_character_,
  example_1L_lgl = F,
  export_1L_lgl = T,
  class_name_1L_chr = "",
  details_1L_chr = "DETAILS",
  args_ls = NULL,
  import_chr = NA_character_,
  doc_in_class_1L_lgl = F,
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

Arguments

fn_name_1L_chr Function name (a character vector of length one)

fn_type_1L_chr Function type (a character vector of length one)

fn Function (a function), Default: NULL

fn_desc_1L_chr Function description (a character vector of length one), Default: 'NA'

fn_out_type_1L_chr Function out type (a character vector of length one), Default: 'NA'

fn_title_1L_chr Function title (a character vector of length one), Default: 'NA'

example_1L_lgl Example (a logical vector of length one), Default: F

export_1L_lgl Export (a logical vector of length one), Default: T

class_name_1L_chr Class name (a character vector of length one), Default: ""

details_1L_chr Details (a character vector of length one), Default: 'DETAILS'

args_ls Arguments (a list), Default: NULL

import_chr Import (a character vector), Default: 'NA'

doc_in_class_1L_lgl Document in class (a logical vector of length one), Default: F

abbreviations_lup

Abbreviations (a lookup table), Default: NULL

object_type_lup

Object type (a lookup table), Default: NULL

object_type_lup

*Object abbreviations lookup table***Description**

A lookup table to identify R object types from an abbreviation that can be used as object name suffices.

Usage

object_type_lup

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 109 rows and 3 columns.

Details

A tibble

short_name_chr Short name (a character vector)

long_name_chr Long name (a character vector)

plural_lgl Plural (a logical vector)

Source

<https://readyforwhatsnext.github.io/readyforwhatsnext/>

read_fns

*Read functions***Description**

`read_fns()` is a Read function that reads an R script into memory. Specifically, this function implements an algorithm to read functions. Function argument `fns_dir_1L_chr` specifies the path to object. The function returns Functions path (a character vector).

Usage

```
read_fns(fns_dir_1L_chr = "data-raw/fns/")
```

Arguments

`fns_dir_1L_chr` Functions directory (a character vector of length one), Default: `'data-raw/fns/'`

Value

Functions path (a character vector)

rowbind_all_tbs_in_r4_obj

Rowbind all tibbles in readyforwhatsnext S4 object

Description

rowbind_all_tbs_in_r4_obj() is a Rowbind function that performs custom rowbind operations on table objects. Specifically, this function implements an algorithm to rowbind all tibbles in readyforwhatsnext s4 object. The function returns Tibbles (a readyforwhatsnext S4).

Usage

```
rowbind_all_tbs_in_r4_obj(tbs_r4, second_tbs_r4, r4_name_1L_chr)
```

Arguments

tbs_r4 Tibbles (a readyforwhatsnext S4)
second_tbs_r4 Second tibbles (a readyforwhatsnext S4)
r4_name_1L_chr Readyforwhatsnext S4 name (a character vector of length one)

Value

Tibbles (a readyforwhatsnext S4)

write_abbr_lup

Write abbreviation

Description

write_abbr_lup() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write abbreviation lookup table. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_abbr_lup(  
  short_name_chr = NA_character_,  
  long_name_chr = NA_character_,  
  no_plural_chr = NA_character_,  
  custom_plural_ls = NULL,  
  overwrite_1L_lgl = T,  
  seed_lup = NULL,  
  url_1L_chr,  
  pkg_nm_1L_chr = get_dev_pkg_nm()  
)
```

Arguments

short_name_chr	Short name (a character vector), Default: 'NA'
long_name_chr	Long name (a character vector), Default: 'NA'
no_plural_chr	No plural (a character vector), Default: 'NA'
custom_plural_ls	Custom plural (a list), Default: NULL
overwrite_1L_lgl	Overwrite (a logical vector of length one), Default: T
seed_lup	Seed (a lookup table), Default: NULL
url_1L_chr	Url (a character vector of length one)
pkg_nm_1L_chr	Package name (a character vector of length one), Default: get_dev_pkg_nm()

```
write_all_tbs_in_tbs_r4_to_csvs
```

Write all tibbles in tibbles readyforwhatsnext S4 to comma separated variables files

Description

write_all_tbs_in_tbs_r4_to_csvs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write all tibbles in tibbles readyforwhatsnext s4 to comma separated variables files. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_all_tbs_in_tbs_r4_to_csvs(
  tbs_r4,
  r4_name_1L_chr,
  lup_dir_1L_chr,
  pfx_1L_chr
)
```

Arguments

tbs_r4	Tibbles (a readyforwhatsnext S4)
r4_name_1L_chr	Readyforwhatsnext S4 name (a character vector of length one)
lup_dir_1L_chr	Lookup table directory (a character vector of length one)
pfx_1L_chr	Prefix (a character vector of length one)

write_and_doc_ds

Write and document dataset

Description

write_and_doc_ds() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write and document dataset. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_and_doc_ds(
  db,
  overwrite_1L_lgl = T,
  db_1L_chr,
  title_1L_chr,
  desc_1L_chr,
  format_1L_chr = "A tibble",
  url_1L_chr = NA_character_,
  vars_ls = NULL,
  R_dir_1L_chr = "R",
  abbreviations_lup = NULL,
  object_type_lup = NULL
)
```

Arguments

db	PARAM_DESCRIPTION
overwrite_1L_lgl	Overwrite (a logical vector of length one), Default: T
db_1L_chr	Database (a character vector of length one)
title_1L_chr	Title (a character vector of length one)
desc_1L_chr	Description (a character vector of length one)
format_1L_chr	Format (a character vector of length one), Default: 'A tibble'
url_1L_chr	Url (a character vector of length one), Default: 'NA'
vars_ls	Vars (a list), Default: NULL
R_dir_1L_chr	R directory (a character vector of length one), Default: 'R'
abbreviations_lup	Abbreviations (a lookup table), Default: NULL
object_type_lup	Object type (a lookup table), Default: NULL

write_and_doc_fn_fls *Write and document function files*

Description

write_and_doc_fn_fls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write and document function files. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_and_doc_fn_fls(
  fns_dmt_tb,
  r_dir_1L_chr = "R",
  path_to_user_dmt_dir_1L_chr = "../.../Documentation/Code/User",
  path_to_dvpr_dmt_dir_1L_chr = "../.../Documentation/Code/Developer",
  make_pdfs_1L_lgl = T,
  dev_pkgs_chr = NA_character_
)
```

Arguments

fns_dmt_tb	Functions documentation (a tibble)
r_dir_1L_chr	R directory (a character vector of length one), Default: 'R'
path_to_user_dmt_dir_1L_chr	Path to user documentation directory (a character vector of length one), Default: '../.../Documentation/Code/User'
path_to_dvpr_dmt_dir_1L_chr	Path to developer documentation directory (a character vector of length one), Default: '../.../Documentation/Code/Developer'
make_pdfs_1L_lgl	Make pdfs (a logical vector of length one), Default: T
dev_pkgs_chr	Development packages (a character vector), Default: 'NA'

write_dmt_fn_type_lup *Write documented function type*

Description

write_dmt_fn_type_lup() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write documented function type lookup table. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_dmt_d_fn_type_lup(
  fn_type_lup_tb = make_fn_type_lup(),
  overwrite_1L_lgl = T,
  pkg_nm_1L_chr = get_dev_pkg_nm(),
  url_1L_chr = url_1L_chr,
  abbreviations_lup = NULL
)
```

Arguments

`fn_type_lup_tb` Function type lookup table (a tibble), Default: `make_fn_type_lup()`

`overwrite_1L_lgl` Overwrite (a logical vector of length one), Default: `T`

`pkg_nm_1L_chr` Package name (a character vector of length one), Default: `get_dev_pkg_nm()`

`url_1L_chr` Url (a character vector of length one), Default: `url_1L_chr`

`abbreviations_lup` Abbreviations (a lookup table), Default: `NULL`

`write_documented_fns` *Write documented functions*

Description

`write_documented_fns()` is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write documented functions. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_documented_fns(tmp_fn_dir_1L_chr, R_dir_1L_chr)
```

Arguments

`tmp_fn_dir_1L_chr` Temporary function directory (a character vector of length one)

`R_dir_1L_chr` R directory (a character vector of length one)

write_fn_type_dirs	<i>Write function type directories</i>
--------------------	--

Description

write_fn_type_dirs() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write function type directories. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_fn_type_dirs(path_1L_chr = "data-raw")
```

Arguments

path_1L_chr Path (a character vector of length one), Default: 'data-raw'

write_nsimps_to_desc	<i>Write namespace imports to description</i>
----------------------	---

Description

write_nsimps_to_desc() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write namespace imports to description. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_nsimps_to_desc(dev_pkgs_chr = NA_character_, incr_ver_1L_lgl = T)
```

Arguments

dev_pkgs_chr Development packages (a character vector), Default: 'NA'

incr_ver_1L_lgl

Incr ver (a logical vector of length one), Default: T

write_pkg	<i>Write package</i>
-----------	----------------------

Description

write_pkg() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write package. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_pkg(package_1L_chr, R_dir_1L_chr = "R")
```

Arguments

package_1L_chr Package (a character vector of length one)
R_dir_1L_chr R directory (a character vector of length one), Default: 'R'

write_pkg_setup_fls	<i>Write package setup files</i>
---------------------	----------------------------------

Description

write_pkg_setup_fls() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write package setup files. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_pkg_setup_fls(  
  path_to_pkg_rt_1L_chr = ".",  
  dev_pkg_nm_1L_chr = get_dev_pkg_nm(),  
  make_tmpl_vignette_1L_lgl = F,  
  incr_ver_1L_lgl = T,  
  delete_contents_of_R_dir = F  
)
```

Arguments

path_to_pkg_rt_1L_chr
 Path to package root (a character vector of length one), Default: '.'
dev_pkg_nm_1L_chr
 Development package name (a character vector of length one), Default: get_dev_pkg_nm()
make_tmpl_vignette_1L_lgl
 Make tmpl vignette (a logical vector of length one), Default: F
incr_ver_1L_lgl
 Incr ver (a logical vector of length one), Default: T
delete_contents_of_R_dir
 PARAM_DESCRIPTION, Default: F

write_pt_lup_db	<i>Write prototype lookup table database</i>
-----------------	--

Description

write_pt_lup_db() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write prototype lookup table database. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_pt_lup_db(R_dir_1L_chr = "R")
```

Arguments

R_dir_1L_chr R directory (a character vector of length one), Default: 'R'

write_std_imp	<i>Write standard import</i>
---------------	------------------------------

Description

write_std_imp() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write standard import. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_std_imp(R_dir_1L_chr = "R")
```

Arguments

R_dir_1L_chr R directory (a character vector of length one), Default: 'R'

write_tb_to_csv	<i>Write tibble to comma separated variables file</i>
-----------------	---

Description

write_tb_to_csv() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write tibble to comma separated variables file. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_tb_to_csv(
  tbs_r4,
  slot_nm_1L_chr,
  r4_name_1L_chr,
  lup_dir_1L_chr,
  pfx_1L_chr
)
```

Arguments

tbs_r4	Tibbles (a readyforwhatsnext S4)
slot_nm_1L_chr	Slot name (a character vector of length one)
r4_name_1L_chr	Readyforwhatsnext S4 name (a character vector of length one)
lup_dir_1L_chr	Lookup table directory (a character vector of length one)
pfx_1L_chr	Prefix (a character vector of length one)

write_to_reset_pkg_files	<i>Write to reset package files</i>
--------------------------	-------------------------------------

Description

write_to_reset_pkg_files() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write to reset package files. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_to_reset_pkg_files(
  delete_contents_of_1L_chr,
  package_1L_chr = get_dev_pkg_nm(),
  package_dir_1L_chr = getwd(),
  description_ls = NULL,
  keep_version_lgl = T
)
```

Arguments

- delete_contents_of_1L_chr
Delete contents of (a character vector of length one)
- package_1L_chr Package (a character vector of length one), Default: get_dev_pkg_nm()
- package_dir_1L_chr
Package directory (a character vector of length one), Default: getwd()
- description_ls Description (a list), Default: NULL
- keep_version_lgl
Keep version (a logical vector), Default: T

write_ws	<i>Write workspace</i>
----------	------------------------

Description

write_ws() is a Write function that writes a file to a specified local directory. Specifically, this function implements an algorithm to write workspace. The function is called for its side effects and does not return a value. **WARNING:** This function writes R scripts to your local environment. Make sure to only use if you want this behaviour

Usage

```
write_ws(path_1L_chr)
```

Arguments

- path_1L_chr Path (a character vector of length one)

%>%	<i>Pipe operator</i>
-----	----------------------

Description

Implements: <https://github.com/sckott/analogsea/issues/32> and <https://github.com/rstudio/ggvis/blob/master/R/pipe.R>.

Arguments

- lhs, rhs An object and a function to apply to it

Index

- * **datasets**
 - abbreviations_lup, [3](#)
 - fn_type_lup_tb, [3](#)
 - object_type_lup, [9](#)
- %>%, [19](#)
- abbreviations_lup, [3](#)
- fn_type_lup_tb, [3](#)
- get_from_lup_obj, [4](#)
- import_xls_sheets, [5](#)
- make_dmt_for_all_fns, [5](#)
- make_fn_dmt_tbl, [6](#)
- make_fn_type_lup, [7](#)
- make_lines_for_fn_dmt, [8](#)
- object_type_lup, [9](#)
- read_fns, [9](#)
- ready4fun (ready4fun-package), [2](#)
- ready4fun-package, [2](#)
- rowbind_all_tbs_in_r4_obj, [10](#)
- write_abbr_lup, [10](#)
- write_all_tbs_in_tbs_r4_to_csvs, [11](#)
- write_and_doc_ds, [12](#)
- write_and_doc_fn_fls, [13](#)
- write_dmt_fn_type_lup, [13](#)
- write_documented_fns, [14](#)
- write_fn_type_dirs, [15](#)
- write_nsimps_to_desc, [15](#)
- write_pkg, [16](#)
- write_pkg_setup_fls, [16](#)
- write_pt_lup_db, [17](#)
- write_std_imp, [17](#)
- write_tb_to_csv, [18](#)
- write_to_reset_pkg_files, [18](#)
- write_ws, [19](#)