# Package 'IRdisplay'

October 12, 2022

**Title** 'Jupyter' Display Machinery

**Description** An interface to the rich display capabilities of 'Jupyter' front-
ends (e.g. 'Jupyter Notebook') <https://jupyter.org>.
Designed to be used from a running 'IRkernel' session <https://irkernel.github.io>.

**Version** 1.1

**URL** https://github.com/IRkernel/IRdisplay

**BugReports** https://github.com/IRkernel/IRdisplay/issues/

**Depends** R (>= 3.0.1)

**Suggests** testthat, withr

**Imports** methods, repr

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Thomas Kluyver [aut, cph],
Philipp Angerer [aut, cph, cre]
(<https://orcid.org/0000-0002-0369-2888>),
Jan Schulz [aut, cph]

**Maintainer** Philipp Angerer <phil.angerer@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-01-04 12:10:16 UTC

## R topics documented:

## display                          *Create and use multiple available reprs*

### Description

Both functions create a mimebundle for multiple reprs. display proceeds to publish it using
[publish_mimebundle](). prepare_mimebundle returns it (see *Value* for details)

### Usage

```
display(
  obj,
  ...,
  mimetypes = getOption("jupyter.display_mimetypes"),
  error_handler = stop
)

prepare_mimebundle(
  obj,
  mimetypes = getOption("jupyter.display_mimetypes"),
  metadata = NULL,
  error_handler = stop
)
```

### Arguments

| | |
|---|---|
| obj | The object to create representations for |
| mimetypes | Mimetypes to create reprs for. The defaults are defined by the option jupyter.display_mimetypes. (see: [IRdisplay-options]) |
| error_handler | Function used when errors in individual reprs occur |
| metadata, ... | Metadata to attach to the result (can be expanded by additional metadata) |

### Value

prepare_mimebundle returns a list with items corresponding to the parameters of [publish_mimebundle]()
(data and metadata)

### See Also

[publish_mimebundle]()

### Examples

```
bundle <- prepare_mimebundle(diag(3))

## Not run: ## (Run inside of an IRkernel)
display(help(display))
```

```
## End(Not run)
```

---

| display_<image> | *Display a specific image output* |
|---|---|

---

### Description

Either `data` or `file` must be passed.

### Usage

```
display_png(data = NULL, file = NULL, width = NULL, height = NULL)

display_jpeg(data = NULL, file = NULL, width = NULL, height = NULL)

display_pdf(data = NULL, file = NULL, width = NULL, height = NULL)

display_svg(data = NULL, file = NULL, width = NULL, height = NULL)
```

### Arguments

| | |
|---|---|
| data | The data as a [raw](#) vector ([character](#) vector for display_svg) |
| file | The path to a file or a [connection](#) containing the content |
| width | The width to display the image |
| height | The height to display the image |

### See Also

[display_<text>](#)

### Examples

```
## Not run: ## (Run inside of an IRkernel)
display_png(file = 'image.png')
display_svg('
<svg xmlns="http://www.w3.org/2000/svg" viewBox="-1 -1 2 2">
  <circle r="1"/>
</svg>
')
display_jpeg(file = url('https://dummyimage.com/600x400.jpg', 'wb'), width = 100)
## End(Not run)
```

---

display_<text>                    *Display a specific textual output*

---

### Description

Either data or file must be passed.

### Usage

```
display_text(data = NULL, file = NULL)

display_json(data = NULL, file = NULL)

display_javascript(data = NULL, file = NULL)

display_html(data = NULL, file = NULL)

display_markdown(data = NULL, file = NULL)

display_latex(data = NULL, file = NULL)
```

### Arguments

| | |
|---|---|
| data | The code or markup content as a [character](character) vector |
| file | The path to a file or a [connection](connection) containing the content |

### See Also

[display_<image>](display_<image>)

### Examples

```
## Not run: ## (Run inside of an IRkernel)
display_text('Just text')
display_markdown('[MD](http://commonmark.org) *formatted*')
display_javascript('execute(this)')
## End(Not run)
```

---

IRdisplay-options     *IRdisplay options*

---

## Description

Some [options](#) to control the formats [display](#) and [prepare_mimebundle](#) emit, and the function they use to display them.

## Usage

```
irdisplay_option_defaults
```

## Format

An object of class list of length 3.

## Options

jupyter.display_mimetypes The default is all MIME types supported by Jupyter.

jupyter.base_display_func Function used by [display](#) and all [display_<text>](#) / [display_<image>](#) functions. Has the signature function(data, metadata = NULL). Per default emits a [warning](#), and is set when running an IRkernel.

jupyter.clear_output_func Function used by [clear_output](#). Has the signature function(wait = TRUE). Per default emits a [warning](#), and is set when running an IRkernel.

---

publish_mimebundle     *Display data by mimetype, with optional alternative representations.*

---

## Description

publish_mimebundle calls the function stored as option value of jupyter.base_display_func, clear_output calls the value of option jupyter.clear_output_func. (see: [IRdisplay-options](#))

## Usage

```
publish_mimebundle(data, metadata = NULL)

clear_output(wait = TRUE)
```

## Arguments

| | |
|---|---|
| data | A named list mapping mimetypes to content ([character](#) or [raw](#) vectors) |
| metadata | A named list mapping mimetypes to named lists of metadata, e.g. list('image/png' = list(width = 5)) |
| wait | Wait to clear the output until new output is available. Default TRUE. If FALSE, clears the existing output immediately before the new output is displayed. |

**Functions**

- `clear_output`: Clear the output from the current cell.

**See Also**

[prepare_mimebundle](#)

**Examples**

```
## Not run: ## (Run inside of an IRkernel)
publish_mimebundle(list('text/html' = '<h1>Hi!</h1>'))
publish_mimebundle(
 list('image/svg+xml' = '<svg xmlns="http://www.w3.org/2000/svg"><circle r="100"/></svg>'),
 list('image/svg+xml' = list(width = 100, height = 100)))

for (i in 1:5) {
  Sys.sleep(.2)    # simulate work
  clear_output()   # clear previous iteration
  cat(i)           # alternative: IRdisplay::display(i)
  flush.console()  # make output available
}
## End(Not run)
```

# Index