

Link-Style

Manual

v0.1.0

2024-04-24

MIT

This packages provides helpers for styling link.

Lennart SCHUSTER

<https://github.com/indicatelovelace/link-style>

Link Style solves the problem of styling different types of links which are not directly selectable as subelements. This includes Phone Numbers, Mail Addresses and URLs.

Table of contents

N.1. About	2
N.2. Usage	2
N.3. API reference	3

I. Index

N.1. About

A common use case for this package is styling emails, phone addresses, document links and internet links differently. Another one is styling different websites differently. This is particularly useful for replacing links to common platforms with icons (Github, LinkedIn, etc.).

N.2. Usage

To use this package, first you have to put

```
#show: make-ref  
...
```

in your document. You can add styling rules:

```
// Shows strict mails strong  
#update-link-style(key: mailto-strict, value: strong)  
// Replace mails with emoji  
#update-link-style(key: mailto, value: it => [#emoji.mail])
```

Styling rules consist of a key and a value. The key is a regex or a string to match an expression. You can find helpers for common link formats in Section N.3. The value is a function that takes and returns content, just like you would use a closure in a common show rule. Care that the order of the rules is of importance. This is the order in which the matchers are applied. Therefore in the above example, valid mail addresses would be styled bold. Other `mailto:` links would be shown as `...`. Inserting the same regex again will replace the rule. In order to effectively use the order of the rules, you can also insert before or after an element. Care that you need to match the key:

```
// Shows strict mails strong  
#update-link-style(key: l-url(base: "typst.org", value: strong)  
#update-link-style(key: l-url(base: "typst.org/docs", before: l-url(base:  
"typst.org"), value: emph)
```

Now the docs page will be `emph`, instead of `strong`.

You can also remove a rule:

```
#remove-link-style(key: mailto-strict)
```

Now for every mail, that was matched by `mailto-strict` before would now be matched by `mailto`.

N.3. API reference

<code>#l-mailto()</code>	<code>#l-phone()</code>	<code>#make-link()</code>
<code>#l-mailto-strict()</code>	<code>#l-url()</code>	<code>#remove-update-style()</code>

`#l-mailto()`

Helper for matching Mails. This one matches every ‘mailto:’ link

`#l-mailto-strict()`

Helper for matching Mails. This one is more strict and matches only, if a mail is an address that is structurally correct.

`#l-phone(<region>: "\"US\"")`

Helper to match phone numbers. Note that these are country dependent.

Argument

<code><region>: "\"US\""</code>	string
---------------------------------------	--------

ISO 3166-1 alpha-2 region string. Currently supported are US, DE, FR, GB and limited to non mobile.

`#l-url(<base>: "\".*?\\"..*?\\"")`

Helper to match urls. Matches any well formed url around the base url. E.g. if base is ‘google.com’, ‘https://google.com/docs’ and ‘http://google.com:8080’ are matched as well.

Argument

<code><base>: "\".*?\\"..*?\\""</code>	string
--	--------

Base url of a hostname. This does not check the url for legality. Though if it is, only legal urls are matched. By default this matches any url like link. Care that you have to escape any symbols with special meaning (e.g. ‘typst.app’ → `regex("typst\\.app")`)

`#update-link-style(<key>: none, <value>: none, <before>: none, <after>: none) → none`

Update the styling array. Entries are inserted. If before and after are none, entries are appended to the end. If a key already exist, the entry is replaced. The order is kept intact.

Argument

<code><key>: none</code>	regex string
--------------------------------	---------------

Matcher for the dest field of the link function. If the key is a string, it is matched if it is contained. If it is a regex expression, it is matched if the expression produces at least one match.

Argument

<code><value>: none</code>	function
----------------------------------	----------

A function that takes content and returns content.

Argument

<code><before>: none</code>	regex string
-----------------------------------	---------------

Insert before another key. Excludes after.

— Argument —

`<after>`: `none`

regex | string

Insert after another key. Excludes before.

#remove-update-style(<key>: none) → none

Removes an entry from the link-style array, if it exists.

— Argument —

`<key>`: `none`

regex | string

Matcher for the dest field of the link function. If the key is a string, it is matched if it is contained. If it is a regex expression, it is matched if the expression produces at least one match.

#make-link() [<body>]

Function for creating applying the link-style.

— Argument —

`<body>`

content

The document body.

#link-style

Array of tuples, consisting of the matcher and corresponding styling function. Care that the order of insertion is the order in which the matchers are evaluated. In practice, this means you should place the more specific matches above more generic ones.

Part I. Index

L

#**l-mailto** 3
#**l-mailto-strict** 3
#**l-phone** 3
#**l-url** 3

M

#**make-link** 4

R

#**remove-update-style** 4

U

#**update-link-style** 3