



Inhaltsverzeichnis

Modul:Format link/Doku

Dies ist die Dokumentationsseite für Modul:Format link

This module, migrated from origins in [Module:Hatnote](#), provides functionality for formatting links for display, including that powering the [Vorlage:TI](#) template.

It can pretty-format section links with the section symbol ("§") and appropriate whitespace, it automatically escapes category and file names with the [colon trick](#), and includes functionality for italicizing the page or section name, and for detecting and categorizing results that produce [red links](#).

Inhaltsverzeichnis

1 Use from wikitext	2
2 Use from other Lua modules	2
2.1 _formatLink	2
2.2 formatPages	3
3 Errors	4

Use from wikitext

The functions in this module cannot be used directly from `#invoke`, and must be used through templates instead. Please see [Template:Format link](#) for documentation on usage of that template.

Use from other Lua modules

To load this module from another Lua module, use the following code:

```
local mFormatLink = require('Module:Format link')
```

You can then use the functions as documented below.

_formatLink

```
mFormatLink._formatLink{
    link = 'Link',
    display = 'Display',
    target = 'Target',
    italicizePage = true,
    italicizeSection = true,
    categorizeMissing = 'Pages using formatted red links'
}
```

Formats *link* as a wikilink. Categories and files are automatically escaped with the **colon trick**, and links to sections are automatically formatted as *page § section*, rather than the MediaWiki default of *page#section*.

Several options modify the output:

- If the *display* value is present, then it will be used as a display value. Any manual piping (using the `{{!}}` magic word or similar) present in *link* will be overridden by the *display* value if present.
- If the *target* value is present, then it will override *link* as a target, but the result will still be *displayed* using either the value from *display*, or the result of formatting *link*.
- If *italicizePage* is true, then the page portion of the link is italicized if present.
- If *italicizeSection* is true, then the section portion of the link is italicized if present.
- If *categorizeMissing* is a non-empty string, then that value is used as a category name, and that category will be applied if the resulting target of the link (no matter whether through *link* or through *target*) doesn't exist.

Examples

```
mFormatLink._formatLink{link = 'Foo#Bar'} → [[:Foo#Bar|Foo §&nbsp;Bar]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = 'Baz', display = 'Qux'} → [[:Baz|Qux]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = 'Foo|Bar', display = 'Baz'} → [[:Foo|Baz]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = '#Foo', target = 'Example#Foo'} → [[:Example#Foo|§&nbsp;Foo]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = 'The Lord of the Rings#Plot', italicizePage = true} → [[:The Lord of the Rings#Plot|"The Lord of the Rings" §&nbsp;Plot]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = 'Cybercrime Prevention Act of 2012#Disini v. Secretary of Justice', italicizeSection = true} → [[:Cybercrime Prevention Act of 2012#Disini v. Secretary of Justice|Cybercrime Prevention Act of 2012 §&nbsp;"Disini v. Secretary of Justice"]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = 'Nonexistent page', categorizeMissing = 'Example'} → [[:Nonexistent page]][[Category:Example]] → Vorlage:Format link
```

```
mFormatLink._formatLink{link = 'Existing', categorizeMissing = 'Example'} → [[:Existing]] → Vorlage:Format link
```

formatPages

```
mFormatLink.formatPages(options, pages)
```



This derived function is useful for lists that format many links. It formats an array of pages using the `_formatLink` function, and returns the result as an array. Options in the *options* table are applied, and use the same names as the options for `_formatLink`.

Example

`Vorlage:Code` → `Vorlage:Code`

Errors

If `_formatLink` is used and neither a *link* nor a *target* argument is provided, then the module will produce an error message instead of its usual output, as it *cannot* then produce valid output.

You can solve this error by providing appropriate parameters to `_formatLink`, or you may want to ensure that a more descriptive error is provided by a downstream template or module when it would otherwise call `_formatLink` with inadequate arguments.