

# Package ‘hypertext’

February 27, 2026

**Title** 'HTML' Element Construction

**Version** 1.0.0

**Description** Provides a deterministic, framework-agnostic Domain-Specific Language for building 'HTML' nodes and rendering them to a string.

**License** MIT + file LICENSE

**URL** <https://github.com/sigflux/hypertext>

**BugReports** <https://github.com/sigflux/hypertext/issues>

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Kennedy Mwavu [aut, cre] (ORCID:  
<<https://orcid.org/0009-0006-3157-7234>>),  
Sigflux [cph, fnd]

**Maintainer** Kennedy Mwavu <[mwavukennedy@gmail.com](mailto:mwavukennedy@gmail.com)>

**Repository** CRAN

**Date/Publication** 2026-02-27 20:00:02 UTC

## Contents

render . . . . .	2
tags . . . . .	2
<b>Index</b>	<b>4</b>

---

render	<i>Render an HTML node tree to a character string</i>
--------	---

---

**Description**

Converts an `hypertext.tag` object (and all its descendants) into an HTML string. Text children are escaped; nested `hypertext.tag` children are rendered recursively.

**Usage**

```
render(x)

## S3 method for class 'hypertext.tag'
render(x)

## Default S3 method:
render(x)

## S3 method for class 'list'
render(x)
```

**Arguments**

`x` An `hypertext.tag` object, a character string, or a list of these.

**Value**

A single character string of HTML.

---

tags	<i>HTML tag functions</i>
------	---------------------------

---

**Description**

A named list of functions, one per HTML5 element. Access individual tags with `tags$div(...)`, `tags$p(...)`, etc.

**Usage**

```
tags
```

**Format**

An object of class `list` of length 115.

**Details**

Named arguments become HTML attributes; unnamed arguments become child nodes or text content.

**Boolean / valueless attributes**

Pass NA or TRUE as the attribute value to produce a valueless attribute (e.g. disabled). FALSE or NULL suppresses the attribute entirely.

**Multi-value attributes**

Character vectors of length > 1 are collapsed with a space, so `class = c("a", "b")` renders as `class="a b"`.

**Examples**

```
tags$p(class = "lead", "Hello, world!")
render(tags$div(id = "app", tags$h1("Title")))
```

# Index

\* **datasets**  
tags, [2](#)

render, [2](#)

tags, [2](#)