

# Shiny HTML Tags Glossary

ADDED: 16 APR 2014  
BY: GARRETT GROLEMUND

In [Customize your Shiny UI with HTML \(/articles/html-tags.html\)](/articles/html-tags.html) you saw that Shiny provides a list of functions named `tags` . Each function in the list creates an HTML tag that you can use to layout your Shiny App. But what if you are unfamiliar with HTML tags? The glossary below explains what the most popular tags in `tags` do.

## tags

The `shiny::tags` object contains R functions that recreate 110 HTML tags.

names(tags)						
##	[1]	"a"	"abbr"	"address"	"area"	"article"
##	[6]	"aside"	"audio"	"b"	"base"	"bdi"
##	[11]	"bdo"	"blockquote"	"body"	"br"	"button"
##	[16]	"canvas"	"caption"	"cite"	"code"	"col"
##	[21]	"colgroup"	"command"	"data"	"datalist"	"dd"
##	[26]	"del"	"details"	"dfn"	"div"	"dl"
##	[31]	"dt"	"em"	"embed"	"eventsourcing"	"fieldset"
##	[36]	"figcaption"	"figure"	"footer"	"form"	"h1"
##	[41]	"h2"	"h3"	"h4"	"h5"	"h6"
##	[46]	"head"	"header"	"hgroup"	"hr"	"html"
##	[51]	"i"	"iframe"	"img"	"input"	"ins"
##	[56]	"kbd"	"keygen"	"label"	"legend"	"li"
##	[61]	"link"	"mark"	"map"	"menu"	"meta"
##	[66]	"meter"	"nav"	"noscript"	"object"	"ol"
##	[71]	"optgroup"	"option"	"output"	"p"	"param"
##	[76]	"pre"	"progress"	"q"	"ruby"	"rp"
##	[81]	"rt"	"s"	"samp"	"script"	"section"
##	[86]	"select"	"small"	"source"	"span"	"strong"
##	[91]	"style"	"sub"	"summary"	"sup"	"table"
##	[96]	"tbody"	"td"	"textarea"	"tfoot"	"th"
##	[101]	"thead"	"time"	"title"	"tr"	"track"
##	[106]	"u"	"ul"	"var"	"video"	"wbr"

You can use any of these functions by subsetting the `tags` list. For example to create an `h1` header in HTML, run:

```
tags$h1("My header")  
## <h1>My header</h1>
```

Some tags functions come with a helper function that makes accessing them easier. For example, the `shiny::h1` function is a wrapper for `tags$h1` .

However, you can access many of the functions in tags only through `tags` because they share a name with a common R function.

The glossary below explains what the most popular tag functions do. The tag functions that do not appear here are less popular, but still useful. You can look them up at an HTML documentation site such as w3schools (<http://www.w3schools.com/tags/>).

# a

Creates a link to a web page. You can access the “a” tag with the helper function `a` .

## Common Attributes Description

href	the address of the web page to link to
------	--

```
tags$a(href="www.rstudio.com", "Click here!")
## <a href="www.rstudio.com">Click here!</a>
```

# audio

Adds an audio element (e.g., a sound to your app).

## Common Attributes Description

autoplay	A valueless attribute. If present, audio starts playing automatically when loaded
controls	A valueless attribute. If present, play controls are displayed
src	The location of the audio file to play
type	The type of file to play

```
tags$audio(src = "sound.mp3", type = "audio/mp3", autoplay = NA, controls = NA)
## <audio src="sound.mp3" type="audio/mp3" autoplay controls></audio>
```

# b

Creates bold text.

```
tags$b("This text is bold.")
## <b>This text is bold.</b>
```

# blockquote

Creates a block of quoted text. Usually it is displayed in a special way.

## Common Attributes Description

cite	the source of the quote
------	-------------------------

```
tags$blockquote("Tidy data sets are all the same. Each messy data set is messy in its own way.", cite = "Hadley Wickham")
## <blockquote cite="Hadley Wickham">Tidy data sets are all the same. Each messy data set is messy in its own way.</blockquote>
```

# br

Creates a line break. You can use the helper function `br` .

```
tags$div(  
  "Some text followed by a break",  
  tags$br(),  
  "Some text following a break"  
)  
## <div>  
##   Some text followed by a break  
##   <br/>  
##   Some text following a break  
## </div>
```

## code

Creates text formatted as computer code. You can use the helper function `code` .

```
tags$code("This text will be displayed as computer code.")  
## <code>This text will be displayed as computer code.</code>
```

## div

Creates a section (e.g., “division”) of an HTML document. divs provide a useful hook for CSS styling. You can use the helper function `div` .

Common Attributes Description	
class	The class of the div, a useful way to style the div with CSS
id	The ID of the div, a useful way to style the div with CSS
style	CSS styling to apply to the div

```
tags$code("This text will be displayed as computer code.")  
## <code>This text will be displayed as computer code.</code>
```

## em

Creates emphasized (e.g., italicized) text. You can use the helper function `em` .

```
tags$em("This text is emphasized.")  
## <em>This text is emphasized.</em>
```

## embed

Embed a plug-in or third party application.

Common Attributes Description	
src	The source of the file to embed
type	The MIME type of the embedded content

height	The height of the embedded content
width	The width of the embedded content

```
tags$embed(src = "animation.swf")
## <embed src="animation.swf"/>
```

## h1, h2, h3, h4, h5, h6

Adds hierarchical headings. `h1` creates a first level heading and `h2` through `h6` create a hierarchy of decreasing subheadings. You can use the helper functions `h1` , `h2` , `h3` , `h4` , `h5` , `h6` .

```
tags$div(
  tags$h1("Heading"),
  tags$h2("Subheading"),
  tags$h3("Subsubheading"),
  tags$h4("Subsubsubheading"),
  tags$h5("Subsubsubsubheading"),
  tags$h6("Subsubsubsubsubheading")
)
## <div>
##   <h1>Heading</h1>
##   <h2>Subheading</h2>
##   <h3>Subsubheading</h3>
##   <h4>Subsubsubheading</h4>
##   <h5>Subsubsubsubheading</h5>
##   <h6>Subsubsubsubsubheading</h6>
## </div>
```

## hr

Adds a horizontal line (e.g., horizontal rule). You can use the helper function `hr` .

```
tags$hr()
## <hr/>
```

## i

Creates italicized text.

```
tags$i("This text is italicized.")
## <i>This text is italicized.</i>
```

## iframe

Creates an inline frame to embed an HTML document in.

Common Attributes	Description
-------------------	-------------

src	The URL of the HTML document to embed
srcdoc	A raw HTML document to embed
scrolling	Should iframe display scrollbars ( yes , no , auto )
seamless	A valueless attribute. Should the iframe seem like part of the web page?
height	The height of the iframe
width	The width of the iframe
name	The name of the iframe

```
tags$iframe(src = "www.rstudio.com", seamless=NA)
## <iframe src="www.rstudio.com" seamless></iframe>
```

## img

Creates an image. You can use the helper function `img` .

### Common Attributes Description

src	The source of the image to embed
height	The height of the image
width	The width of the image

```
tags$img(src = "www.rstudio.com", width = "100px", height = "100px")
## 
```

## link

Creates a link to a separate document. Normally used with CSS style sheets.

## ol and li

Create an ordered list (i.e., a numbered list).

```
tags$ol(
  tags$li("First list item"),
  tags$li("Second list item"),
  tags$li("Third list item")
)
## <ol>
##   <li>First list item</li>
##   <li>Second list item</li>
##   <li>Third list item</li>
## </ol>
```

## p

Create a paragraph (a block of text that begins on its own line). You can access the `p` tag with the helper function `p` too.

```
tags$div(  
  tags$p("First paragraph"),  
  tags$p("Second paragraph"),  
  tags$p("Third paragraph")  
)  
## <div>  
##   <p>First paragraph</p>  
##   <p>Second paragraph</p>  
##   <p>Third paragraph</p>  
## </div>
```

## pre

Create pre-formatted text, text that looks like computer code. You can use the helper function `pre` .

```
tags$pre("This text is preformatted.")  
## <pre>This text is preformatted.</pre>
```

## script

Add a client-side script such as javascript. You must wrap the actual script in `HTML` to prevent it from being passed as text.

```
tags$script(HTML("if (window.innerHeight < 400) alert('Screen too small');"))
```

## span

Create a group of inline elements. Normally used to style a string of text. You can use the helper function `span` .

```
tags$div(  
  HTML(paste("This text is ", tags$span(style="color:red", "red"), sep = ""))  
)  
## <div>This text is <span style="color:red">red</span></div>
```

## strong

Create bold text. You can use the helper function `strong` .

```
tags$strong("This text is strongly emphasized.")  
## <strong>This text is strongly emphasized.</strong>
```

## style

Create style specifications. A way to add CSS styles directly to your Shiny App.

# sub, sup

Create subscript or super script.

```
tags$div(  
  HTML(paste("E = mc", tags$sup(2), sep = "")),  
  HTML(paste("H", tags$sub(2), "0", sep = ""))  
)  
## <div>  
##   E = mc<sup>2</sup>  
##   H<sub>2</sub>0  
## </div>
```

# ul and li

Create an unordered list (i.e., a list of bullet points).

```
tags$ul(  
  tags$li("First list item"),  
  tags$li("Second list item"),  
  tags$li("Third list item")  
)  
## <ul>  
##   <li>First list item</li>  
##   <li>Second list item</li>  
##   <li>Third list item</li>  
## </ul>
```

# video

Add a video.

## Common Attributes Description

autoplay	A valueless attribute. If present, video starts playing automatically when loaded
controls	A valueless attribute. If present, Shiny will display play controls.
src	The location of the video file to play
height	The height of the video
width	The width of the video

```
tags$video(src = "video.mp4", type = "video/mp4", autoplay = NA, controls = NA)  
## <video src="video.mp4" type="video/mp4" autoplay controls></video>
```

---

We love it when R users help each other, but RStudio does not monitor or answer the comments in this thread. If you'd like to get specific help, we recommend the Shiny Discussion Forum (<https://groups.google.com/forum/#!forum/shiny-discuss>) for in depth discussion of Shiny related questions and How to get help (<http://shiny.rstudio.com/articles/help.html>) for a list of the best ways to get help with R code.

Shiny is an RStudio (<http://www.rstudio.com>) project. © 2014 RStudio, Inc.