Ex 10

Harikrishna Reddy Benakana

2024-04-10

## HTML in R

The call to the with\_html function generates the following HTML: ```

A heading

Some text &some bold text.

<imgmyimg.png />

This HTML does not generate an image because the img tag is missing src, width, and height attributes. Fix the void\_tag function for img, so the img tag will have src, width, and height attributes.

# Define the html function  
html <- function(x) structure(x, class = "html")  
print.html <- function(x, ...) {  
 out <- paste0("<HTML> ", x)  
 cat(paste(strwrap(out), collapse = "\n"), "\n", sep = "")  
}  
  
# Define the escape functions  
escape <- function(x) UseMethod("escape")  
escape.html <- function(x) x  
escape.character <- function(x) {  
 x <- gsub("&", "&amp;", x)  
 x <- gsub("<", "&lt;", x)  
 x <- gsub(">", "&gt;", x)  
   
 html(x)  
}  
escape.list <- function(x) {  
 lapply(x, escape)  
}  
  
# Define the named and unnamed functions  
named <- function(x) {  
 if (is.null(names(x))) return(NULL)  
 x[names(x) != ""]  
}  
unnamed <- function(x) {  
 if (is.null(names(x))) return(x)  
 x[names(x) == ""]  
}  
  
# Define the html\_attributes and html\_attribute functions  
html\_attributes <- function(list) {  
 if (length(list) == 0) return("")  
   
 attr <- map2\_chr(names(list), list, html\_attribute)  
 paste0(" ", unlist(attr), collapse = "")  
}  
  
html\_attribute <- function(name, value = NULL) {  
 if (length(value) == 0) return(name) # for attributes with no value  
 if (length(value) != 1) stop("`value` must be NULL or length 1")  
   
 if (is.logical(value)) {  
 # Convert T and F to true and false  
 value <- tolower(value)  
 } else {  
 value <- escape\_attr(value)  
 }  
 paste0(name, "='", value, "'")  
}  
escape\_attr <- function(x) {  
 x <- escape.character(x)  
 x <- gsub("\'", '&#39;', x)  
 x <- gsub("\"", '&quot;', x)  
 x <- gsub("\r", '&#13;', x)  
 x <- gsub("\n", '&#10;', x)  
 x  
}  
  
# Define the tag function  
tag <- function(tag) {  
 force(tag)  
 function(...) {  
 args <- list(...)  
 attribs <- html\_attribute(named(args))  
 children <- unlist(escape(unnamed(args)))  
   
 html(paste0(  
 "<", tag, attribs, ">",  
 paste(children, collapse = ""),  
 "</", tag, ">"  
 ))  
 }  
}  
  
# Define the void\_tag function using the tag function  
void\_tag <- function(tag) {  
 force(tag)  
 function(...) {  
 args <- list(...)  
 if (length(unnamed(args)) > 0) {  
 stop("Tag ", tag, " can not have children", call. = FALSE)  
 }  
 attribs <- html\_attribute(named(args))  
   
 html(paste0("<", tag, attribs, " />"))  
 }  
}  
  
# Create the img function using void\_tag for the img tag  
img <- void\_tag("img")  
  
# Processing all tags  
tags <- c("a", "abbr", "address", "article", "aside", "audio",   
 "b","bdi", "bdo", "blockquote", "body", "button", "canvas",   
 "caption","cite", "code", "colgroup", "data", "datalist",   
 "dd", "del","details", "dfn", "div", "dl", "dt", "em",   
 "eventsource","fieldset", "figcaption", "figure", "footer",   
 "form", "h1", "h2", "h3", "h4", "h5", "h6", "head", "header",   
 "hgroup", "html", "i","iframe", "ins", "kbd", "label",   
 "legend", "li", "mark", "map","menu", "meter", "nav",   
 "noscript", "object", "ol", "optgroup", "option", "output",   
 "p", "pre", "progress", "q", "ruby", "rp","rt", "s", "samp",   
 "script", "section", "select", "small", "span", "strong",   
 "style", "sub", "summary", "sup", "table", "tbody", "td",   
 "textarea", "tfoot", "th", "thead", "time", "title", "tr",  
 "u", "ul", "var", "video")  
  
void\_tags <- c("area", "base", "br", "col", "command", "embed",  
 "hr", "img", "input", "keygen", "link", "meta", "param",   
 "source", "track", "wbr")  
  
tag\_fs <- c(  
 setNames(lapply(tags, tag), tags),  
 setNames(lapply(void\_tags, void\_tag), void\_tags)  
)  
  
tag\_fs$p("Some text.", tag\_fs$b("Some bold text"),  
 tag\_fs$i("Some italic text"))

# Define the with\_html function  
with\_html <- function(code) {  
 eval(substitute(code), tag\_fs)  
}  
  
# Generate the HTML with the image tag  
output\_html <- with\_html(body(  
 h1("A heading", id = "first"),  
 p("Some text &", b("some bold text.")),  
 img(src = "/Users/harikrishnareddy/Desktop/R/WassnaaAl-Mawee/image.png", width = 100, height = 100)  
))  
  
print(output\_html)

## <HTML> <body><h1first>A heading</h1><p>Some text &amp;<b>some bold  
## text.</b></p><img/Users/harikrishnareddy/Desktop/R/WassnaaAl-Mawee/image.png  
## /><img100 /><img100 /></body>