Package 'R2HTML'

January 20, 2025

Version 2.3.4

Date 2024-06-16

Title HTML Exportation for R Objects

Depends R (>= 2.0)

Imports stats, graphics, utils, grDevices, methods

Suggests boot, splines, survival, cluster, nlme, rpart, nnet

Description Includes HTML function and methods to write in an HTML file. Thus, making HTML reports is easy. Includes a function that allows redirection on the fly, which appears to be very useful for teaching purpose, as the student can keep a copy of the produced output to keep all that he did during the course. Package comes with a vignette describing how to write HTML reports for statistical analysis. Finally, a driver for 'Sweave' allows to parse HTML flat files containing R code and to automatically write the corresponding outputs (tables and graphs).

License GPL (>= 2)

URL https://github.com/nalimilan/R2HTML

NeedsCompilation no

Author Eric Lecoutre [aut],
Milan Bouchet-Valat [cre, ctb],
Thomas Friedrichsmeier [ctb]

Maintainer Milan Bouchet-Valat <nalimilan@club.fr>

Repository CRAN

Date/Publication 2024-06-16 17:40:02 UTC

Contents

`HTLMReplaceNA` *Internal R2HTML functions*

Description

Internal R2HTML functions

Usage

```
HTMLReplaceNA(Vec, Replace = " ")
HTMLCommand(x, file = HTMLGetFile(),
Num = "", menu = FALSE, target= "index<-main.html", append = TRUE, ...)
HTMLcode(x,...)
```

Arguments

`Vec`	string
`Replace`	string to use for missing values
`x`	a string corresponding to a R command
`file`	the target HTML file
`Num`	number of the command
`menu`	to build a menu of commands
`target`	As command is put in a left frame, name of the linked target HTML page
`append`	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
`...`	...

Details

These are not to be called by the user.

`HTM2clip` *Wrapper around HTML() to save output to the clipboard*

Description

Calls HTML() with appropriate filename and append attributes to write output to clipboard (currently only works on Windows).

Usage

```
HTML2clip(x,
          filename =file("clipboard",
              ifelse(.Platform$OS == "windows", "w",
                     stop("Writing to clipboard only supported on Windows"))),
          append = FALSE, ...)
```

Arguments

`x`	object to be output to HTML
`filename`	destination output file, defaults to clipboard
`append`	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'decides if the filevalue of the width HR optional argument, in pixel or percent
`...`	... (passed on to HTML())

Value

no value returned.

Note

This function was contributed by Gabor Grothendieck.

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
if (.Platform$OS == "windows")
HTML2clip(summary(lm(rating ~., attitude)))
```

HTML *Outputs an object to a HTML file*

Description

Generic method equivalent to print that performs HTML output for any R object.

Usage

```
HTML(x, ...)
```

Arguments

x	R object to export
...	...

Details

HTML function acts like cat, with a file argument, which has to be used and a append argument, which is set by default to TRUE. A convenient default value for file argument can be set by calling HTMLInitFile or HTMLSetFile, so that one may begin to set this variable and omit the argument thereafter. Most of the current classes are handled by HTML function. Some specific methods such as HTML.matrix or HTML.data.frame do own a lot of arguments to specify the way the data are exported.

Value

no value returned.

Author(s)

Eric Lecoutre

See Also

HTML.data.frame,HTML.function, HTMLInitFile, HTMLStart

Examples

```
dir.create(file.path(tempdir(),"R2HTML"))
target <- HTMLInitFile(file.path(tempdir(),"R2HTML"),filename="sample", BackGroundColor="#BBBBEE")
HTML("<br>Don't forget to use the CSS file in order to benefit from fixed-width font",file=target)
tmp <- as.data.frame(matrix(rnorm(100),ncol=10))
HTML(tmp,file=target)
HTMLEndFile()
```

`HTML.cormat` *Write a correlation matrix with HTML formatting*

Description

Although the class 'cormat' is not defined, the function is called `HTML.cormat`, to highlight the fact it should be called only with a correlation matrix. It is documented as a regular S3 method for technical reasons. Contrary to the signature shown, below, you will call it as `HTML.cormat`, explicitly, as shown in the examples.

Usage

```
## S3 method for class 'cormat'
HTML(x, file = HTMLGetFile(),
digits = 2, append = TRUE, align = "center",
caption = "", captionalign = "bottom",
classcaption = "captiondataframe", classtable = "cormat", useCSS = TRUE, ...)
```

Arguments

`x`	a correlation matrix
`file`	target HTLM output
`digits`	number of digits to use for rounding
`append`	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
`align`	alignment to be used: center, left or right
`caption`	optional caption to append to the table
`captionalign`	alignment to be used for the caption: could be bottom or top
`classcaption`	CSS class to use for caption
`classtable`	CSS class to be used for the whole table (in html <table> statement)
`useCSS`	whether to use CSS or not for traffic highlighting
`...`	...

Value

returns (invisibly) the input

Note

"Highlighting traffic" is a simple technique which allows to have a visual representation of data. It is particularly well suited for correlation matrices in order to have at a glance the underlying (linear) structure of data. If your output doesn't rely on CSS styles, you should use `useCSS=FALSE` option, whihc hard codes grey levels for correlations.

For CSS uses, you can (re)define colours and other attributes for correlations in `seq(0,1,length=11)`-defined intervals. Some definitions could be equivalent, not showing any difference. You can, by example, redefine CSS so that only correlations greater than 0.9 are showned, and lower the size of cells, which could be usefull for veryu big datasets.

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
tmpfic=HTMLInitFile(tempdir(),CSSFile="http://www.stat.ucl.ac.be/R2HTML/Pastel.css")
data(iris)
HTML(as.title("Fisher Iris dataset / Correlation matrix - normal matrix"),
     file=tmpfic)
HTML(cor(iris[,1:4]), file=tmpfic)
HTML(as.title("Fisher Iris dataset / Correlation matrix - traffic highlighting"),
     file=tmpfic)
HTML.cormat(cor(iris[,1:4]), file=tmpfic)

# File is generated, you can call the browser:
## Not run: browseURL(tmpfic)
```

HTML.data.frame *Write a data.frame (or matrix) to a HTML output*

Description

This function exports a data.frame to a HTML file. Many arguments allow to customize the layout of the HTML table.

Usage

```
## S3 method for class 'data.frame'
HTML(x, file = HTMLGetFile(),
    Border = 1, innerBorder = 0, classfirstline = "firstline",
    classfirstcolumn = "firstcolumn", classcellinside = "cellinside",
    append = TRUE, align = "center", caption = "", captionalign = "bottom",
    classcaption = "captiondataframe", classtable = "dataframe",
    digits = getOption("R2HTML.format.digits"),
    nsmall = getOption("R2HTML.format.nsmall"),
    big.mark = getOption("R2HTML.format.big.mark"),
    big.interval = getOption("R2HTML.format.big.interval"),
    decimal.mark = getOption("R2HTML.format.decimal.mark"),
    sortableDF = getOption("R2HTML.sortableDF"), row.names = TRUE, ...)
```

Arguments

x	a data.frame
file	target HTLM output
Border	the size of the border around the table. Could be 0,1,... but also NULL
innerBorder	the size of the border inside the table - see details
classfirstline	CSS class for the first line (header - variable names)
classfirstcolumn	CSS class for the first column (rownames)
classcellinside	CSS class for others cells
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
align	alignment to be used: center, left or right
caption	optional caption to append to the table
captionalign	alignment to be used for the caption: could be bottom or top
classcaption	CSS class to use for caption
classtable	CSS class to be used for the whole table (in html <table> statement)
digits	number of digits to use for rounding
nsmall	number of digits which will always appear to the right of the decimal point in formatting real/complex numbers in non-scientific formats. Allowed values '0 <= nsmall <= 20'
big.mark	character; if not empty used as mark between every 'big.interval' decimals before (hence 'big') the decimal point
big.interval	see 'big.mark' above; defaults to 3
decimal.mark	the character used to indicate the numeric decimal point
sortableDF	See details
row.names	logical. If 'TRUE' row.names are shown in the output; otherwise they are omitted
...	...

Details

For the moment, `HTML.matrix` and `HTML.data.frame` do have the same options. Tables are build using two different HTML tables, one beeing encapsulated within the other, which allows to have a table without borders inside but with a border outside. It is nevertheless recommended to rely on CSS to obtain such results...

Now `format` is called to format numerical values (modif. suggested by Arne Henningsen). The corresponding arguments are: `digits`, `nsmall`, `big.mark`, `big.intervall` and `decimal.mark`. For each argument, one can supply either a single value or a vector. In the second case, the size of the vector has to be the number of columns of the data.frame and formatting options will be used element-wise (column by column).

Some options are used to pass default values. You can see those options with (by example): `getOption("R2HTML.format.decimal.mark")` and `options("R2HTML.format.decimal.mark"=",")`

Sortable data.frame uses a DHTML behavior. This requires the file '`tablesort.htc`' which comes with **R2HTML** to be placed in the same directory than the output. This functionality only works for HTML files located on a web server (not on local computer).

Value

no value returned.

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
tmpfic=HTMLInitFile(tempdir(),CSSFile=system.file("samples", "R2HTML.css", package="R2HTML"))
data(iris)
HTML(as.title("Fisher Iris dataset"),file=tmpfic)
HTML(iris, file=tmpfic)
# File is generated, you can call the browser:
## Not run: browseURL(tmpfic)

# Export one line of iris using default decimal separator
HTML(iris[1,],file="")

# Seeing default decimal separator:
getOption("R2HTML.format.decimal.mark")

# Modifying it:
options("R2HTML.format.decimal.mark"=",")
HTML(iris[1,],file="")

# Bypassing value set in option:
HTML(iris[1,],file="",decimal.mark="*")

# Using a vector for formatting options
HTML(iris[1:2,1:2],nsmall=c(3,1),file="")
```

HTML.function *Writes the code of a function to a target HTML file*

Description

Writes the code of a function to a target HTML file

Usage

```
## S3 method for class 'function'
HTML(x, file = HTMLGetFile(), append=TRUE,...)
```

Arguments

x	Name of a function
file	target HTLM output
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
...	...

Details

The code of the function is written in the target HTML file, using HTML <XMP> tag. A CSS class called "function" is used to personalise the output.

Value

no value returned.

Note

For a discussion about .HTML.file default value for file argument, refer to HTMLStart

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
## Define a function and export it's code to the file /test.html.
## Target file may be changed when submitting this code...

myfile <- paste(tempfile(),".html",sep="")
myfun <- function(x){
cat("\n Euclidian norm")
return(sqrt(sum(x^2)))
 }
HTML(myfun,file=myfile)
cat("\n Test output written in: ",myfile)
```

HTML.latex *Insert a piece of LaTeX into a HTML file*

Description

This makes use of AsciiMathML javascript functions. Standard LaTeX input will be turned into MathML and displayed through any brower extension that can handle MathML (such as MathPlayer)

Usage

```
as.latex(x, label=NULL,
inline=ifelse(is.null(label), TRUE, FALSE), count=ifelse(is.null(label), FALSE, TRUE))
## S3 method for class 'latex'
HTML(x, file = HTMLGetFile(), ...)
```

Arguments

x	String containing mathematics in a LaTeX notation
file	HTML target output file
label	String - Label to be displayed before the equation
inline	Boolean - Place of the equation within the output flux - see details
count	Boolean - Should the equation be numbered or not?
...	...

Details

Mathematical notations will be translated in MathML by the AsciiMathML javascript program of Peter Jipsen. Note that his functions allow translating equations with a notation simpler than LaTeX (see his page on AsciiMathML for details). Pieces of LaTeX could be put *inline* (within text) or on a single line : same opposition that the one between $...$ and $$...$$. In order to work, a reference to the javascipt file has to be present within the HTML file **and** the HTML body tag has also to include onload="translate()". All the necessary stuff is included in HTMLInitFile.

Value

no value returned.

Author(s)

Eric Lecoutre

References

AsciiMathML: http://www1.chapman.edu/~jipsen/mathml/asciimath.xml

See Also

HTMLInitFile,HTML

Examples

```
## Not run:
 fic = HTMLInitFile()
 HTML.title("sample page",1,file=fic)
 HTML("First paragraph",file=fic)
 cat("Some text and then an equation:",file=fic,append=TRUE)
 HTML(as.latex("\int_{-\infty}^{1}f(x)dx") ,file=fic)
 cat(". Nice isn't it?",file=fic,append=TRUE)
 HTML(as.latex("\int_{-\infty}^{1}f(x)dx",inline=FALSE) ,file=fic)
 HTML(as.latex("\int_{-\infty}^{1}f(x)dx",inline=FALSE,count=TRUE) ,file=fic)
 HTML(as.latex("\int_{-\infty}^{1}f(x)dx",inline=FALSE,label="My equation") ,file=fic)
 cat("file:", fic, "is created")
 browseURL(fic)
## End(Not run)
```

HTML.title *Writes a title in a target HTML output*

Description

A title is a string with the S3 class "title". The function `as.title` gives this class to an object, so that title method of HTML could apply to it. However, it is also possibly to call this method, explicitly, providing a plain string.

Usage

```
## S3 method for class 'title'
HTML(x, HR = 2, CSSclass=NULL,
file = HTMLGetFile(), append=TRUE,...)
as.title(x)
```

Arguments

x	string
HR	rank attribute of the HTML <H?> tag
CSSclass	CSS class to use for personalised reports
file	the target HTML file
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
...	...

Value

no value returned.

Note

For a discussion about .HTML.file default value for file argument, refer to HTMLStart

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
## Write a title in the file /test.html.
## Target file may be changed when submitting this code...

myfile <- paste(tempfile(),".html",sep="")

tit1 <- as.title("This is method 1")

HTML(tit1, file=myfile)

HTML.title("This is method 2",file=myfile, HR=3)
cat("\n Test output written in: ",myfile)
```

HTMLbr *Facility functions to write HTML code*

Description

Write
, <li> and <hr> tags, which are often used, to an output file.

Usage

```
HTMLbr(x=1, file = HTMLGetFile(), append=TRUE)
HTMLli(txt="", file = HTMLGetFile(), append=TRUE)
HTMLhr(file = HTMLGetFile(), Width = "100%", Size = "1",
CSSclass=NULL, append=TRUE)
```

Arguments

x	number of to put
txt	text to appear after the tag
file	HTML target output file
Width	value of the width HR optional argument, in pixel or percent
Size	value of the size HR optional argument
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
CSSclass	CSS class to use for personalised reports

Value

no value returned.

Note

For a discussion about .HTML.file default value for file argument, refer to HTMLStart

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
## Insert a line to a HTML file
## Change the path/name of the file to redirect to your test file

myfile <- paste(tempfile(),".html",sep="")
HTMLhr(file=myfile)
cat("\n Test output written in: ",myfile)
```

HTMLChangeCSS *Change the current CSS file for dynamic use of package*

Description

When using in dynamic mode, a call to HTMLStart copy the R2HTML.css file to the specified output directory (temp by default). HTMLChangeCSS copy a new CSS file to this destination (or to working directory). Then, the produced HTML files are now based on this new CSS.

Usage

```
HTMLChangeCSS(newCSS = "R2HTML", from = NULL)
```

Arguments

newCSS	Name of the CSS to use (without the extension)
from	Source directory where to search the CSS file

Value

A boolean: wether this has been done or not.

Note

In order to work properly, this assumes you hae used R2HTML.css file (the default one), as this is this file which will be replaced by the new one.

Author(s)

Eric Lecoutre

See Also

HTMLStart

Examples

```
## Not run:
HTMLStart()
(x=diag(3))
HTMLChangeCSS("Pastel")
# refresh the browser

## End(Not run)
```

HTMLCSS *Insert HTML code to refer to an external CSS file*

Description

Allow to use CSS file in a report

Usage

```
HTMLCSS(file = HTMLGetFile(), append = TRUE, CSSfile = "R2HTML.css")
```

Arguments

file	the target HTML file
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
CSSfile	name of the CSS file to refer

Value

no value returned.

Author(s)

Eric Lecoutre

References

For information about CSS, have a look at W3 web site http://www.w3.org/Style/CSS/

Examples

```
myfile <- file.path(tempdir(),"tmp.html")
HTMLCSS(myfile,CSSfile="myownCSS.CSS")
```

HTMLgrid *Creates a HTML grid using ActiveWidget grid - www.activewidgets.com*

Description

All HTMLgrid functions do use the component ActiveWidget grid. Basically, we use this component to display data, so the functions export a data.frame. Data could be stored within the HTML file (HTMLgrid_inline) or in an external raw text file which would be required asynchronously (HTMLgrid).

Usage

```
HTMLgrid(x, file = HTMLGetFile(), append = TRUE,
includeref = FALSE, align = "center", digits = getOption("R2HTML.format.digits"),
nsmall = getOption("R2HTML.format.nsmall"),
big.mark = getOption("R2HTML.format.big.mark"),
big.interval = getOption("R2HTML.format.big.interval"),
decimal.mark = getOption("R2HTML.format.decimal.mark"),
asDF = TRUE, browse = FALSE, classes = NULL, showimages = TRUE)
HTMLgrid_inline(x,file = HTMLGetFile(), append=TRUE,
includeref=FALSE, align="center", digits=getOption("R2HTML.format.digits"),
nsmall = getOption("R2HTML.format.nsmall"),
big.mark = getOption("R2HTML.format.big.mark"),
```

```
big.interval = getOption("R2HTML.format.big.interval"),
decimal.mark = getOption("R2HTML.format.decimal.mark"),
asDF=TRUE,browse=FALSE, classes=sapply(x,class), showimages=TRUE)
HTMLgrid_summary(x,file=NULL,append=TRUE, digits=getOption("R2HTML.format.digits"),
nsmall = getOption("R2HTML.format.nsmall"),
big.mark = getOption("R2HTML.format.big.mark"),
big.interval = getOption("R2HTML.format.big.interval"),
decimal.mark = getOption("R2HTML.format.decimal.mark"), browse=FALSE)
HTMLgrid_references(file=)
```

Arguments

x	a data.frame
file	target HTLM output - see details below
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
includeref	logical. If 'TRUE', references to necessary CSS+Javascript files will be includes. See details.
align	"center", "left" or "right"
digits	number of digits to use for rounding
nsmall	number of digits which will always appear to the right of the decimal point in formatting real/complex numbers in non-scientific formats. Allowed values '0 <= nsmall <= 20'
big.mark	character; if not empty used as mark between every 'big.interval' decimals before (hence 'big') the decimal point
big.interval	see 'big.mark' above; defaults to 3
decimal.mark	the character used to indicate the numeric decimal point
asDF	logical. If 'TRUE', output will be considered as a data frame (which also mean by default we show icons for data type)
browse	logical. If 'TRUE', the output file will directly be called within a browser.
classes	vector of classes for the object. If NULL, the default, will be created using real classes. Could be used to bypass defaukt formatting associated with each class
showimages	logicial. Display or not little icons in columns headers

Details

Those exportation methods require severall external files, including the runtime version of ActiveWidgets grid. To include the necessary references, you can use HTMLgrid_references. Be careful: if you intend to put multiple grids on a same output, the references has to be present only once in the output. \ If you set the file argument to NULL, a new temp file will be created.

Value

file	target HTLM output

Note

Presentation relies on pure CSS + Javascript. It may be possible than future upgrade change the presentation of grids created with this version.

Author(s)

Eric Lecoutre

References

ActiveWidgets Grid 1.0 - http://www.activewidgets.com

Examples

```
data(iris)
fic <- HTMLInitFile(useGrid=TRUE,useLaTeX=FALSE)
fic <- HTMLgrid_inline(iris,file=fic)
cat("\n Browse file 'fic':",fic)
## Not run: browseURL(fic)
```

HTMLInitFile *Begins / Ends a new HTML report output*

Description

`HTMLInitFile` handles the beginning and `HTMLEndFile` the ending of a HTML report, by writing the HTML <body><head><title></title></head>...</body> tags and their options. When working manually, the user may need to use it's own functions or to explicitly write to a file using cat("", file=).

`HTMLInitFile` and `HTMLSetFile` sets the default file path to be used by `HTML` functions, and `HTMLGetFile` retrieves it.

Usage

```
HTMLInitFile(outdir = tempdir(), filename="index", extension="html",
HTMLframe=FALSE, BackGroundColor = "FFFFFF", BackGroundImg = "",
Title = "R output", CSSFile="R2HTML.css", useLaTeX=TRUE, useGrid=TRUE)
HTMLEndFile(file = HTMLGetFile())
HTMLSetFile(file)
HTMLGetFile()
```

Arguments

`outdir`	directory to store the output
`filename`	target HTML report filename
`extension`	target HTML report extension (htm, html,...)

HTMLframe	should the output be handled by frames [boolean]
BackGroundColor	option bgcolor for HTML tag <body>
BackGroundImg	option background for HTML tag <body>
Title	string to pass to HTML <title> tag
CSSFile	path and name of a CSS file to use
useLaTeX	boolean - add required references to javascript AsciiMathML in order to use as.latex
useGrid	boolean - add required references to javascript grid in order to use R2HTML grid fonctions
file	target HTML file to set as default or to end

Value

physical path of the main HTML file that will serve for the report.

Author(s)

Eric Lecoutre

See Also

HTML, as.latex, HTMLgrid

Examples

```
# Store in target the name of a output file
dir.create(file.path(tempdir(),"R2HTML"))
target <- HTMLInitFile(file.path(tempdir(),"R2HTML"),"index", BackGroundColor="#BBBBEE")
# Use target to write a dataframe
HTML(as.title("Here is the data frame"),file=target)
HTML("<br>Don't forget to use the CSS file in order to benefit from fixed size police",
     file=target)
tmp <- as.data.frame(matrix(rnorm(100),ncol=10))
HTML(tmp,file=target)
HTMLEndFile()
```

HTMLInsertGraph *Insert a graph in a HTML report*

Description

Write the HTML <img> tag to an output, so that a existant graph could be displayed in the HTML report

Usage

```
HTMLInsertGraph(GraphFileName="", Caption="", GraphBorder=1,
Align="center", WidthHTML=500, HeightHTML=NULL,
file=HTMLGetFile(), append=TRUE,...)
```

Arguments

GraphFileName	Name of the target graph (GIF, JPEG or PNG)
Caption	If non empty, text to be written under the graph, as its caption
GraphBorder	Size of the border, in pixels
Align	Alignment of the graph (center, left or right)
WidthHTML	Width of the image in HTML
HeightHTML	Height of the image in HTML (NULL for not specified)
file	Name of the target HTML file (the report)
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
...	...

Details

The steps to add a graph to a HTML file are the following: first create the graph, by using a device convenient for web pages, such as GIF, JPEG or PNG. Ensure to write it in the same directory than the HTML file. Then call HTMLInsertGraph.

Value

no value returned.

Author(s)

Eric Lecoutre

See Also

HTMLplot

Examples

```
directory=tempdir()
HTMLoutput=file.path(directory,"output.html")
graph1="graph1.png"
# Write graph to a file
## Not run: png(file.path(directory,graph1))
## Not run: plot(table(rpois(100,5)), type = "h", col = "red",
             lwd=10,main="rpois(100,lambda=5)")
## End(Not run)
## Not run: dev.off()
# Insert graph to the HTML output
HTMLInsertGraph(graph1,file=HTMLoutput,caption="Sample discrete distribution plot")
```

HTMLplot *Insert a graphic into an HTML output*

Description

Exports the active graphic to a JPEG or GIF file and add it to a target HTML output, by writing the <IMG> tag.

Usage

```
HTMLplot(Caption = "", file = HTMLGetFile(), append = TRUE,
GraphDirectory = ".", GraphFileName = "", GraphSaveAs = "png", GraphBorder = 1,
Align = "center", Width = 500, Height = 500, WidthHTML = NULL, HeightHTML = NULL,
GraphPointSize = 12, GraphBackGround = "white", GraphRes = 72, plotFunction = NULL, ...)
```

Arguments

Caption	text to be placed below the graphic, as a caption
file	the target HTML file
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
GraphDirectory	path where file should be stored; ignore in a HTMLStart/HTMLStop session
GraphFileName	name of the file to produce (could be missing)
GraphSaveAs	an existing exportation device, such as jpg or gif
GraphBorder	HTML border option for the <IMG> tag
Align	HTML align option for the <IMG> tag
Width	width of the image to create (passed to the driver)
Height	height of the image to create (passed to the driver) (NULL: not specified)
WidthHTML	Width of the image in HTML
HeightHTML	Height of the image in HTML (NULL for not specified)
GraphPointSize	To be passed to the device creator function
GraphBackGround	To be passed to the device creator function
GraphRes	To be passed to the device creator function
plotFunction	Function to be evaluated for the on-the-fly creation of the graph to be exported
...	...

Details

Note that this function is coded to work automatically when using automatic exportation with HTMLStart. When using manualy, user should pay attention to the GraphDirectory option, so that graph files are in the same directory as HTML output files. When using to write reports in a non interactive way, first generate the graphic using a device and then use HTMLInsertGraph.

Value

no value returned.

Author(s)

Eric Lecoutre

See Also

HTMLStart, HTMLInsertGraph

Examples

```
## Plots a graphic and insert it into the file /test.html.
## Target file and also graph directory should be changed when submitting this code...

myfile <- paste(tempfile(),".html",sep="")
plot(sin, -pi, 2*pi,main="Sinus")
# HTMLplot(file=myfile,GraphDirectory="/",Caption="Look at this curve!")
```

HTMLStart *Start / Stop the automatic redirection of output to HTML files*

Description

Add the automatic redirection of output to an HTML file. The R session is modified in the following way: a new prompt is proposed (by default HTML>) and each parsed command is also evaluated using HTML generic method, so that the user benefits of both a normal and a HTML output. Please read carefully the details below.

Usage

```
HTMLStart(outdir = tempdir(), filename = "index", extension = "html",
echo = FALSE, autobrowse = FALSE, HTMLframe = TRUE, withprompt = "HTML> ",
CSSFile = "R2HTML.css", BackGroundColor = "FFFFFF", BackGroundImg = "",
Title = "R output")
HTMLStop()
```

Arguments

outdir	physical directory to store the output
filename	name of the target HTML main file
extension	extension of the target HTML file (htm, html,...)
echo	should the parsed commands be written in the output? [boolean]
autobrowse	should the browser be invoked each time a command is issued? [boolean]
HTMLframe	should the output have a HTML frame structure? [boolean]

withprompt	prompt to display while using HTMLStart/HTMLStop
CSSFile	path and name of a CSS file to use
BackGroundColor	option bgcolor for HTML tag <body>
BackGroundImg	option background for HTML tag <body>
Title	string to pass to HTML <title> tag

Details

The user may need to know the following points which describe how R2HTML does work:

- Each parsed command is evaluated and the returned value is passed to the generic function HTML. This evaluation is assured by addTaskCallback function, which is used to add a specific task each time R has to parse an expression.

- A new environment is built, where internal variables such as physical path are stored. This environment is not visible by the user. It is destroyed when calling HTMLStop.

Value

no useful output is returned.

Note

The argument echo is very usefull for teaching purposes.

Author(s)

Eric Lecoutre

See Also

HTML

Examples

```
# Perform's one's own direct report

dir.create(file.path(tempdir(),"R2HTML"))
HTMLStart(file.path(tempdir(),"R2HTML"),HTMLframe=FALSE, Title="My report",autobrowse=FALSE)
as.title("This is my first title")
x <- 1
y<- 2
x+y
HTMLStop()


## Use for interactive teaching course
if (interactive()){
dir.create(file.path(tempdir(),"R2HTML"))
HTMLStart(file.path(tempdir(),"R2HTML"),echo=TRUE)
```

```
as.title("Manipulation vectors")
1:10
sum(1:10)
c(1:10,rep(3,4))
HTMLStop()
}
```

HTMLstem *Insert a stem-and-leaf plot in the HTML output*

Description

Insert a stem-and-leaf plot in the HTML output.

Usage

```
HTMLstem(x, file = HTMLGetFile(), append = TRUE, ...)
```

Arguments

x	a numeric vector.
file	the target HTML file
append	logical. If 'TRUE' output will be appended to 'file'; otherwise, it will overwrite the contents of 'file'
...	any other argument that may be passed to stem, such as scale,...

Details

As stem internal function does not return anything but directly print to console, there is no way to automatically export it to the HTML output. Thus, HTMLstem simply captures the output and write it to the HTML file. When using the package in a interactive way, you should call HTMLstem.

Value

no value returned.

Author(s)

Eric Lecoutre

See Also

stem,HTML

Examples

```
data(islands)
tmpfic=paste(tempfile(),"html",sep=".")
HTMLstem(log10(islands),tmpfic)
cat("\n stem-and-leaf writen to:", tmpfic,"\n")
```

RweaveHTML *A driver to parse HTML noweb files with Sweave tool*

Description

This driver parses HTML files containing R code and replace pieces of code with their output. Graphs are incorporated as png.

Usage

```
RweaveHTML()
```

Value

None value is returned. From a .snw noweb file, the corresponding .html is produced (as eventuals png files for graphs).

Note

In order to work properly, noweb codes have to be located at the beginning of a line (no indentation). See samples in the samples directory of the package.

Author(s)

Eric Lecoutre

See Also

Sweave

Examples

```
## Not run:
library(tools)
Sweave("file.snw",driver=RweaveHTML)

## End(Not run)
```

Index