```
print.xtable <- function(x,
    type = getOption("xtable.type", "latex"),
    file = getOption("xtable.file", ""),
    append = getOption("xtable.append", FALSE),
    floating = getOption("xtable.floating", TRUE),
    floating.environment = getOption("xtable.floating.environment", "table"),
    table.placement = getOption("xtable.table.placement", "ht"),
    caption.placement = getOption("xtable.caption.placement", "bottom"),
    caption.width = getOption("xtable.caption.width", NULL),
    latex.environments = getOption("xtable.latex.environments", c("center")),
    tabular.environment = getOption("xtable.tabular.environment", "tabular"),
    size = getOption("xtable.size", NULL),
    hline.after = getOption("xtable.hline.after", c(-1,0,nrow(x))),
    NA.string = getOption("xtable.NA.string", ""),
    include.rownames = getOption("xtable.include.rownames", TRUE),
    include.colnames = getOption("xtable.include.colnames", TRUE),
    only.contents = getOption("xtable.only.contents", FALSE),
    add.to.row = getOption("xtable.add.to.row", NULL),
    sanitize.text.function = getOption("xtable.sanitize.text.function", NULL),
    sanitize.rownames.function = getOption("xtable.sanitize.rownames.function",
                                           sanitize.text.function),
    sanitize.colnames.function = getOption("xtable.sanitize.colnames.function",
                                           sanitize.text.function),
    math.style.negative = getOption("xtable.math.style.negative", FALSE),
    html.table.attributes = getOption("xtable.html.table.attributes", "border=1"),
    print.results = getOption("xtable.print.results", TRUE),
    format.args = getOption("xtable.format.args", NULL),
    rotate.rownames = getOption("xtable.rotate.rownames", FALSE),
    rotate.colnames = getOption("xtable.rotate.colnames", FALSE),
    booktabs = getOption("xtable.booktabs", FALSE),
    scalebox = getOption("xtable.scalebox", NULL),
    width = getOption("xtable.width", NULL),
    comment = getOption("xtable.comment", TRUE),
    timestamp = getOption("xtable.timestamp", date()),
    ...)
{
    type <- tolower(type)

    #validate that output type and various latex environments are valid
    validatedVars <- envirValidate(type, floating.environment, table.placement,
                                   caption.placement, tabular.environment, floating,
                                   hline.after, x)
    floating <- validatedVars$floating
    table.placement <- validatedVars$table.placement

    #assign caption and short.caption from attributes for usage
    captions <- captionOrganise(x)
    short.caption <- captions$short.caption
    caption <- captions$caption

    ## Claudio Agostinelli <claudio@unive.it> dated 2006-07-28 include.rownames,
    ## include.colnames
    pos <- 0
    if (include.rownames) pos <- 1

    ## add.to.row checks
    if (!is.null(add.to.row)) {
        #validate that add.to.row commands have been entered correctly, return add.to.row and npos
        res <- addToRowValidate(add.to.row, pos, x)
        add.to.row <- res$add.to.row
        npos <- res$npos
    } else {
        add.to.row <- list(pos = list(),command = character(0))
        npos <- 0
    }

    #assign the latex line rules to be used
    if (type == "latex") {
      ##this assigns the latex line rules depending on booktabs
      PHEADER <- lineRule(x, hline.after, booktabs)
    } else {
      PHEADER <- ""
    }

    lastcol <- rep(" ", nrow(x)+2)

    #line rule locations assigned into add.to.row
    add.to.row <- hlineLocations(booktabs, add.to.row, npos, hline.after, PHEADER)

    #assign position and command of addtorow commands
    if ( length(add.to.row$command) > 0 ) {
         addCmds <- assignAddCommands(add.to.row, lastcol)
         lastcol <- addCmds$lastcol
    }

    #params are a set of CAPS variables used to build the table
    if (type == "latex") {
        params <- latexParams(type, tabular.environment, floating,
                              floating.environment, table.placement,
                              latex.environments, x, include.rownames,
                              width, caption.placement, short.caption,
                              caption, lastcol, scalebox, size,
                              caption.width)
    } else {
        params <- htmlParams(html.table.attributes, caption.placement, x, pos)
    }

    #use formatted user arguments and buildParams to make and return final table
    result <- makeTable(params, file, append, comment, type, timestamp,only.contents, floating, caption,
                        caption.placement, x, include.colnames, include.rownames, sanitize.colnames.function,
                        sanitize, rotate.colnames,pos, sanitize.rownames.function, rotate.rownames, format.args,
                        sanitizeNumbers, math.style.negative, sanitize.text.function, NA.string, lastcol,
                        tabular.environment, booktabs, PHEADER)

    if (print.results){
        print(result)
    }

    return(invisible(result$text))
}
```