

HTML Coder

The purpose of this is to simply the creation of a web page involving graphic images, a block of text, and a Heading. I find it useful sometimes in converting untidy emails such as those set left of centre, with unnecessary massive TPI print, and inter line spacing. A number of such conversions can often be seen from contributions I receive each day at <http://www.sigord.co.uk/News.htm> A number of sites host the complete package such as at

http://download.cnet.com/HTMLcoder/3000-10247_4-10906797.html?tag=mncol

As you can see it allows for a variety of options such selecting the Text, heading and background colours along with the size of fonts, and width of the images. The height of the images are adjusted accordingly. Text is COPY and Pasted in, allowing for automatic reformatting to remove all extra line and word spacing. Also any single CHR\$ such as the dreaded
Gordon

```
nomainwin
' Beginners HTML Thanks to Alyce for image323.dll
dim dir$(10,3) : dim pic$(60,2)
head$ = "" : hfile$ = "" : tfile$ = "" : this$ = DefaultDir$
tcol$ = "000000" : bcol$ = "FFFFFF"
txtsize$ = "3" : headsize$ = "5" : imagewidth$ = "500"
open "image323.dll" for dll as #im
[main]
WindowWidth = 800 : WindowHeight = 600
button #h, "Text Colour", [textcol], UL, 140, 140
button #h, "Back Colour", [backcol], UL, 320, 140
button #h, "Heading", [heading], UL, 470, 140
button #h, "Colour", [headcol], UL, 540, 140
button #h, "Text Window", [text], UL, 175, 180
button #h, "Insert JPGs", [jpg], UL, 296, 180
button #h, "Save web page", [page], UL, 416, 180
button #h, "Save text", [file], UL, 550, 180
button #h, "Text Size ", [txtsize], UL, 180, 220
button #h, "Heading Size ", [headsize], UL, 350, 220
button #h, "Images Width", [imgwidth], UL, 520, 220
button #h, "Display HTML", [show], UL, 180, 280
button #h, "Test web page", [test], UL, 300, 280
button #h, "Display .txt", [tfile], UL, 430, 280
button #h, "* QUIT *", [quit], UL, 550, 280
open "Simple Web Pages" for graphics_nsb as #h
#h "trapclose [quit]; font arial 18 bold; place 280 40; down"
```

```
#h "\Simple Web Pages"
#h "font arial 8 bold; color black; place 250 56"
#h "\Produced with Liberty Basic - ver 2.1- Gordon Sweet"
#h "font arial 10 bold; place 180 80; color darkgreen"
#h
"\You can import any text by using COPY then Text Window Options to"
#h
"\PASTE, up to 50 JPGs must be placed in an adjoining folder to this"
#h
"\program. The HTML file that is saved and the folder must be uploaded
."
#h
"\Note the display of samples such as the colours etc is only temporar
y."
#h "font fixedsys 9; color black; place 250 160 " : #h "\Sample"
#h "place 610 160" : #h "\Sample"
#h "color darkblue"
#h "place 180 320" : #h "|Web page = ";hfile$
#h "place 180 340" : #h "|Text file = ";tfile$
#h "place 180 360" : #h "\HEADING = ";head$
#h "place 180 260" : #h
"\Current Text size 3, Heading size 5, Images width 500"
#h "place 420 164; box 440 140; flush"
'get DC of graphicbox
gDC=GetDC(hwnd(#h))
'create memory DC
mDC=CreateCompatibleDC(gDC)
hw=hwnd(#h)'graphicbox handle
wait

[textcol]
gosub [colours] : tcol$ = html$
#h "place 250 160; color " + col$ : #h "\Sample"
#h "color black"
wait

[backcol]
gosub [colours] : bcol$ = html$
#h "backcolor ";col$ : #h "place 420 164; boxfilled 440 140"
#h "backcolor white"
wait

[colours]
colordialog "red", col$
if val(col$) = 0 then col$ ="0 0 0" : notice "black selected"
r$ = word$(col$,1) : g$ = word$(col$,2) : b$ = word$(col$,3)
```

```
html$ = RIGHT$("0"+dechex$(val(r$)),2)
html$ = html$ + RIGHT$("0"+dechex$(val(g$)),2)
html$ = html$ + RIGHT$("0"+dechex$(val(b$)),2)
return
```

[heading]

```
prompt "Enter a heading";head$ : if head$ = "" then wait
#h "place 180 360" : #h "\HEADING = ";head$ : wait
```

[headcol]

```
gosub [colours] : hcol$ = html$
#h "place 610 160; color " + col$ : #h "\Sample"
#h "color black"
wait
```

[page]

```
DefaultDir$ = this$
prompt "Enter filename.htm";hfile$
if hfile$ = "" then wait
open hfile$ for output as #2
#2 "<html><head><title>Text & Images</title></head>"
t$ = "<body><body bgcolor="+bcol$+">"
#2 t$
t$ = "<font face=arial><font size="+headsiz$+"
"><font color="+hcol$+">"
#2 t$
t$ = "<br><b><center>"+head$+"</center></b><font>"
#2 t$
#2 "</center></b><font><br>"
t$ = "<font face=arial><font size="+textsiz$+"
"><font color="+tcol$+">"
#2 t$
#2 "<BLOCKQUOTE><b>"
#2 new$
#2 "</b></BLOCKQUOTE>"
#2 t$
for N = 1 to qtyFiles
    t$ = "<center></center><p>"
    if qtyFiles > 0 then #2 t$
next N
#2 "</body><html>"
close #2
#h "place 180 320" : #h "|Web page = ";hfile$;
#h "place 180 340" : #h "|Text file = ";tfile$
```

```
wait
```

```
[file]
```

```
DefaultDir$ = this$  
prompt "Enter filename.txt";tfile$  
if tfile$ = "" then wait  
open tfile$ for output as #2  
#2 new$  
close #2  
#h "place 180 320" : #h "|Web page = ";hfile$;  
#h "place 180 340" : #h "|Text file = ";tfile$  
wait
```

```
[textsize]
```

```
prompt "Enter text size 1/7";textsize$ : q = val(textsize$)  
if q < 1 or q > 7 then notice "INVALID ENTRY !" :textsize$ = "3"  
#h "place 180 260"  
#h "\Current Text size ";textsize$;  
" Heading size ";headsize$;" Images width ";imgwidth$;"  
wait
```

```
[headsize]
```

```
prompt "Enter heading size 1/7";headsize$ : q = val(headsize$)  
if q < 1 or q > 7 then notice "INVALID ENTRY !" : headsize$ = "5"  
#h "place 180 260"  
#h "\Current Text size ";textsize$;  
" Heading size ";headsize$;" Images width ";imgwidth$;"  
wait
```

```
[imgwidth]
```

```
prompt "Enter images width over 99"  
;imagewidth$ : q = val(imagewidth$)  
if q < 100 then notice "INVALID ENTRY !" : imagewidth$ = "500"  
#h "place 180 260"  
#h "\Current Text size ";textsize$;  
" Heading size ";headsize$;" Images width ";imagewidth$;"  
wait
```

```
[show]
```

```
if hfile$ = "" then notice "NO FILE !" : wait  
op$ = "notepad.exe "+hfile$ : run op$  
wait
```

```
[test]
```

```
if hfile$ = "" then notice "NO Web Page" : wait  
CALL ShellExecute hWnd, hfile$
```

```
wait
```

```
[tfile]
```

```
if tfile$ = "" then notice "NO FILE !" : wait  
op$ = "notepad.exe "+tfile$ : run op$  
wait
```

```
[text]
```

```
close #h  
menu #t, "OPTIONS", "Paste", [insert], "Clear", [clear],  
"Remove Blank Lines", [lines], _  
"Erase Character", [del], "Reformat Text", [format], |,  
"MAIN MENU", [quitclip]  
open "Text" for text as #t  
#t "!trapclose [quitclip]" : #t "!font fixedsys 9"  
wait
```

```
[insert]
```

```
#t "!cls" : #t "!paste" : #t "!contents? new$";  
#t "!contents? text$"; : #t "!cls"  
if len(text$) < 50 then notice "Text too short to reformat" : wait  
temp$ = "" : p = 1 : lp = 1 : l = len(text$)  
while p < l+1  
    k$ = mid$(text$, p, 1)  
    k = asc(k$) : p = p + 1  
    new$ = new$ + chr$(k)  
    temp$ = temp$ + chr$(k) : lp = lp + 1  
    if lp > 88 and k = 32 or lp > 124 then  
        #t temp$ : temp$ = "" : lp = 0  
    end if  
wend  
#t "!contents? new$"; : #t "!origin 1 1"  
wait
```

```
wait
```

```
[clear]
```

```
#t "!cls" : text$ = "" : #t "!copy" ;  
wait
```

```
[lines]
```

```
text$ = new$  
if len(text$) < 5 then notice "NO REAL TEXT !!" : wait  
new$ = "" : p = 1 : l = len(text$) : k1 = 0 : ls = 0  
while p < l+1  
    ok = 0
```

```
k$ = mid$(text$,p,1) : k = asc(k$)
if k = 13 and k1 <> 999 then ls = 0
if k = 9 or k > 31 and k < 127 then ok = 1
if k = 13 and ls = 0 then
    new$ = new$ + chr$(13)+chr$(10)
    ls = 1 : k1 = 999 : p = p + 1
end if
if ok = 1 then new$ = new$ + chr$(k) : k1 = k
p = p + 1
wend
text$ = new$ : #t "!cls" : #t text$
#t "!contents? new$"; : #t "!origin 1 1"
wait
```

[del]

```
if new$ = "" then notice "NO TEXT !!" : wait
d$ = "" : prompt "Enter one character to delete"; d$
if len(d$) <> 1 then notice "ONE CHARACTER ONLY !!" : wait
#t "!contents? text$"; : #t "!cls"
if len(text$) < 50 then notice "Text too short to reformat" : wait
x =asc(d$) : new$= "" : p = 1 : l = len(text$)
while p < l+1
    ok = 0
    k$ = mid$(text$,p,1) : k = asc(k$)
    if k = 9 or k > 31 and k < 127 then ok = 1
    if k = 13 then
        new$ = new$ + chr$(13)+chr$(10)
        p = p + 1
    end if
    if ok = 1 and k <> x then new$ = new$ + chr$(k)
    p = p + 1
wend
text$ = new$ : #t "!cls" : #t text$
#t "!contents? new$"; : #t "!origin 1 1"
wait
```

[format]

```
#t "!contents? text$"; : #t "!cls"
if len(text$) < 50 then notice "Text too short to reformat" : wait
temp$ = "" : p = 1 : lp = 1
l =len(text$) : k1=0
while p < l+1 : ok = 0
    k$ = mid$(text$,p,1)
    k =asc(k$) : p= p + 1
    if k = 9 or k > 31 and k < 127 then ok = 1
    if k = 13 then k = 32 : ok = 1
```

```
if k = 32 and k1 = 32 then ok = 0
if ok = 1 then
    new$ = new$ + chr$(k) : k1 = k
    temp$ = temp$ + chr$(k) : lp = lp + 1
end if
if lp > 88 and k = 32 or lp > 124 then
    #t temp$ : temp$ = "" : lp = 0
end if
wend
#t "!contents? new$"; : #t "!origin 1 1"
wait
```

[jpg]

```
DefaultDir$ = left$(DefaultDir$,2)+"\HTMLcode"
filedialog "Select any JPG file","*.jpg", File$
if File$ = "" then wait

sFile$ = noPath$(File$) : plen = len(File$)-len(sFile$)
path$ = left$(File$,plen) : ext$ = "*.jpg"
files path$, ext$, dir$()
qtyFiles = val(dir$(0, 0))
if qtyFiles > 50 then notice
"LIMITED to maximum of 50)" : qtyFiles = 50
'reformat the file information
for x = 1 to qtyFiles
    dir$(x, 1) = right$(" " + dir$(x, 1), 9)
    pic$(x,1) =dir$(x, 0)
next x

FL = len(path$)-1
for TEST = FL to 1 step -1
    if mid$(path$,TEST,1)="\" then
        PL = FL-TEST : exit for
    end if
next TEST
dpath$ = right$(path$,PL+1)

#h "place 10 390 " : #h "|";space$(20);qtyFiles;
"JPGs FILES IN";dpath$

for j = 1 to qtyFiles
    jpg$ = jpg$ + " - ";pic$(j,1)
    if len(jpg$) > 70 then #h "\";jpg$ : jpg$ = ""
    pic$(j,1) = dpath$ + pic$(j,1) : bmp$ = pic$(j,1)
    gosub [picsize] : f = val(imagewidth$)
    height = int(f/bw * bh)
```

```
        pic$(j,2) = str$(height)
next : #h "\";jpg$
wait
```

```
[picsize]
    hImage=LoadImageFile(hw,bmp$)
    if hImage=0 then
        notice "Function failed."
        cursor normal
        wait
    end if
    if hDemo<>0 then unloadbmp "demo"
    loadbmp "demo",hImage : hDemo=hbmp("demo")
    if bmp$="" then notice "NO image" : goto [quit]
    bw=BitmapWidth(hDemo) : bh=BitmapHeight(hDemo)
    return
```

```
[quitclip] close #t : goto [main]
```

```
[quit]
    if hDemo<>0 then unloadbmp "demo"
    if hImage<>0 then ok = DeleteObject(hImage)
    call ReleaseDC hwnd(#h), gDC
    call DeleteDC mDC
    close #h : close #im : end
```

```
Function GetDC(hwnd)
    CallDLL #user32, "GetDC",_
        hwnd As Long,_'window or control handle
    GetDC As Long'returns device context
End Function
```

```
Sub ReleaseDC hwnd, hDC
    CallDLL#user32,"ReleaseDC",_
        hwnd As Long,_'window or control handle
        hDC As Long,_'handle of DC to delete
    result As Long
End Sub
```

```
Function CreateCompatibleDC(hDC)
    CallDLL #gdi32,"CreateCompatibleDC",_
        hDC As Long,_'window DC
    CreateCompatibleDC As Long'memory DC
End Function
```

```
Sub DeleteDC hDC
```

```
    CallDLL #gdi32, "DeleteDC", _
        hDC As Long, _ 'memory DC to delete
        r As Boolean
End Sub

Function BitmapWidth(Hbmp)
    struct BITMAP, _
        bmType As Long, _
        bmWidth As Long, _
        bmHeight As Long, _
        bmWidthBytes As Long, _
        bmPlanes As Word, _
        bmBitsPixel As Word, _
        bmBits As Long

    nSize=Len(BITMAP.struct)
    CallDLL #gdi32, "GetObjectA", Hbmp As Long, _
        nSize As Long, BITMAP As struct, _
        results As Long

    BitmapWidth=BITMAP.bmWidth.struct
End Function

Function BitmapHeight(Hbmp)
    struct BITMAP, _
        bmType As Long, _
        bmWidth As Long, _
        bmHeight As Long, _
        bmWidthBytes As Long, _
        bmPlanes As Word, _
        bmBitsPixel As Word, _
        bmBits As Long

    nSize=Len(BITMAP.struct)
    CallDLL #gdi32, "GetObjectA", Hbmp As Long, _
        nSize As Long, BITMAP As struct, _
        results As Long

    BitmapHeight=BITMAP.bmHeight.struct
End Function

Function DeleteObject(hObject)
    CallDLL #gdi32, "DeleteObject", _
        hObject As Long, _ 'handle of object
        DeleteObject As Long 'returns whatever
End Function
```

```
Function LoadImageFile(hWnd, file$)
    'load an image from file,
    'bmp, jpg, emf, wmf, ico
    'returns handle of memory bmp
    callDll #im, "LoadImageFile", hWnd as ulong, _
        file$ as ptr, LoadImageFile as ulong
End Function

function noPath$(t$)
    while instr(t$, "\")
        t$ = mid$(t$, 2)
    wend
    noPath$ = t$
end function

Sub ShellExecute hWnd, cf$
    parameter = _SW_SHOWNORMAL' set up for viewing
    lpszOp$ = "open" + Chr$(0)' "open" or "play" or "print"
    lpszFile$ = cf$ + Chr$(0)
    lpszDir$ = DefaultDir$ + Chr$(0)
    lpszParams$="" + Chr$(0)
    CallDll #shell32, "ShellExecuteA", hWnd As long, lpszOp$ As
ptr, lpszFile$ As ptr, _
        lpszParams$ As ptr, lpszDir$ As ptr, parameter As long
, result As long
End Sub
```