It did a ghastly contrast bear
To those bright ringlets glistering fair.

---

The aim of this column is to provide odd hints or small pieces of code that might help in solving a problem or two while hopefully not making things worse through any errors of mine.

---

The macro \texttt{\member{⟨name⟩}} can be used several times to add ⟨name⟩ to the \texttt{\members} macro. The macro \texttt{\themembers} can then be called to print the contents of \texttt{\members} and clear \texttt{\members} so a new list may be started. If you want to print the list more than once then use \texttt{\showmembers} which prints, but does not clear, the list.

\begin{verbatim}
\member{Fred} \member{Joe}
\member{Susan} \member{Faye}
\themembers ⇒ Fred, Joe, Susan, Faye
\end{verbatim}

For more complex additions, for instance when the macro to be extended takes arguments, then the \texttt{patchcmd} package [4] could be the answer.

Once having created a list of members it might have to be changed because one or more members have left. This is more complicated and I present it only as an example of what could be done.

The \texttt{\deletemember{⟨name⟩}} will go over the list of members, creating a new temporary working list with the exception of the ⟨name⟩ member, then replace the original list with the working one.

\begin{verbatim}
\makeatletter
\let\xpf\expandafter% just a shorthand
\newcommand*{\deletemember}[1]{%\makeatother
% a new list of members
% define it as the argument (member name)
  \newcommand{\members}{#1}{%\makeatother
% a list exists, add the argument to it
  \g@addto@macro{\members}{, #1}{%\makeatother
% ignore spaces \members\newcommand*{\showmembers}{%\makeatother
% \let\members\relax\makeatother

% % the macro \member{⟨name⟩} can be used several times to add ⟨name⟩ to the \members macro. The macro \themembers can then be called to print the contents of \members and clear \members so a new list may be started. If you want to print the list more than once then use \showmembers which prints, but does not clear, the list.
\member{Fred} \member{Joe}
\member{Susan} \member{Faye}
\themembers ⇒ Fred, Joe, Susan, Faye

For more complex additions, for instance when the macro to be extended takes arguments, then the patchcmd package [4] could be the answer.

Once having created a list of members it might have to be changed because one or more members
to the working list.\footnote{1} The bunch of \texttt{\expandafter} around the call to \texttt{\transfer} expands \texttt{\member@} to its definition before it gets handed over as the argument to \texttt{\transfer}.

The macro \texttt{\transfer{⟨name⟩}} adds \texttt{⟨name⟩} to the macro \texttt{\@tempmembers} containing a list of comma separated names. It has the same general form as the earlier \texttt{\member} macro.

\begin{verbatim}
\newcommand*{\transfer}[1]{{% 
  \@ifundefined{\@tempmembers}{{% 
    \newcommand*{\@tempmembers}{#1}%; \% 
  }{% 
    \g@addto@macro{\@tempmembers}{,#1}%; \% 
  }}% 
}\makeatother
\end{verbatim}

Here are some examples of adding and deleting members to and from the original member list above.

\begin{verbatim}
\member{Alice} \member{Bob} \member{Claire} \member{David} \member{Erica}
\end{verbatim}

\texttt{\showmembers} ⇒ Fred, Joe, Susan, Faye, Alice, Bob, Claire, David, Erica

\begin{verbatim}
deletemember{David}\showmembers ⇒ Fred, Joe, Susan, Faye, Claire, Erica
deletemember{Fred}\showmembers ⇒ Joe, Susan, Faye, Alice, Bob, Claire, Erica
deletemember{Miriam}\showmembers ⇒ Joe, Susan, Faye, Alice, Bob, Claire, Erica, Xerxes, Zeno
\end{verbatim}

\subsection*{1.2 Piecing a paragraph}

Ron Aaron wanted a different kind of assembly. He wrote [1]:

\begin{quote}
What I wish to do is accumulate text into a paragraph ‘as I go’. My simple approach is to allocate a box, and then unbox and add the text. But this doesn’t work as I intend:
\end{quote}

\begin{verbatim}
\newbox\textbox 
def\addbox #1{% 
  \setbox\textbox = \vbox{\unvbox\textbox#1} 
}\addbox{Hello} \addbox{there!} \box\textbox
\end{verbatim}

What I get is each appended bit of text in a separate line. I’ve tried to \texttt{\unskip} and \texttt{\unkern} etc. after the \texttt{\unvbox} but whatever I do I get a list of lines . . .

Trying out Ron’s example the result is:

\begin{verbatim}
Hello, there!
\end{verbatim}

The squashed vertical spacing between the lines is real, not an artifact of this article.

In responding, Philip Taylor [5], having said that using a \texttt{\vbox} would be difficult, then gave two suggestions; either use an \texttt{\hbox} directly or a token-list register. His \texttt{\hbox} solution (and my example) is:

\begin{verbatim}
\newbox\textbox 
def\addbox #1{% 
  \setbox \textbox = \hbox{\bgroup 
    \unhbox \textbox#1% 
  \egroup} 
}\addbox{Hello} \addbox{ World!} \addbox{ Now isn’t that a rather common saying?} \unhbox\textbox 
\end{verbatim}

with the result:

\begin{verbatim}
Hello World! Now isn’t that a rather common saying?
\end{verbatim}

At various points after this I have used code like

\begin{verbatim}
\addbox{ (n) text}
\end{verbatim}

as an example of assembling a paragraph piece by piece and at the end showing the result via:

\begin{verbatim}
\unhbox\textbox 
\addbox{(1) Start of a paragraph.}
\end{verbatim}

Philip’s second solution uses a token register:

\begin{verbatim}
\newtoks\texttoks 
def\addtoks #1{% 
  \texttoks = \expandafter {\the \texttoks #1} 
}\addtoks {Goodbye} \addtoks {\emph{vain} world. Ah, the weariness in that statement does one no good.} \the \texttoks
\end{verbatim}

and the example produces:

\begin{verbatim}
Goodbye \emph{vain} world. Ah, the weariness in that statement does one no good.
\end{verbatim}

\begin{verbatim}
\addbox{(2) After an interruption add more.}
\end{verbatim}

Note that with both of Philip’s solutions you have to explicitly incorporate spaces where you want
them to occur in the assembled paragraph. It seemed, though, that Ron really wanted to use a \vbox but I have neither seen nor been able to come up with satisfactory code.

\addbox{ (3) This is the end of the piecewise paragraph.}

\unhbox\textbox

Now print the piecewise paragraph giving:

\begin{itemize}
  \item (1) Start of a paragraph.
  \item (2) After an interruption add more.
  \item (3) This is the end of the piecewise paragraph.
\end{itemize}

Beneath those rugged elms, that yew-tree’s shade,
Where heaves the turf in many a mouldering heap,
Each in his narrow cell for ever laid,
The rude forefathers of the hamlet sleep.

\vbox{\elegy}

\textit{Elegy Written in a Country Churchyard, Thomas Gray}

\section{Table talk}

Arbo \cite{2} wanted a tabular layout like the one shown in Figure 1 and tried using code like this to produce it.

\begin{tabular}{|r|c|l|}
\hline
First & Second & Third \\
Text & \multicolumn{3}{c|}{C 1 & C 2 & C 3 & C 4} & More text \\
Words & \multicolumn{2}{c|}{C 5 & C 6 & C 7} & text \\
Title & \multicolumn{2}{c}{C 8 & C 9} & Some text \\
\hline
\end{tabular}

References

\begin{itemize}
  \item \cite{1} Ron Aaron. How to append text to a paragraph (in an existing vbox)? Post to \texttt{xetex} mailing list, 16 July 2010.
  \item \cite{2} Arbo. How to produce multiple columns within a multicolumn. Post to \texttt{comp.text.tex} newsgroup, 2 November 2010.
  \item \cite{3} Donald Arseneau. Re: How to produce multiple columns within a multicolumn. Post to \texttt{comp.text.tex} newsgroup, 2 November 2010.
  \item \cite{4} Michael J. Downes. The patchcmd package, 2000. \texttt{http://ctan.org/pkg/patchcmd}.
  \item \cite{5} Philip Taylor. Re: How to append text to a paragraph (in an existing vbox)? Post to \texttt{xetex} mailing list, 16 July 2010.
\end{itemize}

\begin{table}[h]
\begin{tabular}{|c|c|c|c|}
\hline
First & Second & Third \\
Text & C 1 & C 2 & C 3 & C 4 & More text \\
Words & C 5 & C 6 & C 7 & text \\
Title & C 8 & C 9 & Some text \\
\hline
\end{tabular}
\caption{Desired tabular layout}
\end{table}