A not so widely known feature of the verbatim handler in ConTExt is the ability to add comments in another style and MkIV even offers a bit more. Here some examples are shown.
Annotating verbatim content is done using a mechanism called escaping. For such special cases it's often best to define a specific instance.

\[\text{\texttt{\textbackslash definetyping}}\]
\[
\text{[annotatedtyping]}
\]
\[
\text{[escape=} / \text{, color=darkblue, before=} , \text{after=} ]}
\]

\[\text{\texttt{\textbackslash startannotatedtyping}}\]
\[
\text{bla = test /bgroup /sl oeps /egroup}
\]
\[
\text{/bgroup /bf some more /egroup}
\]
\[
\text{| another test}
\]
\[
\text{| somethingverylong /bgroup /it oeps /egroup}
\]
\[\text{\texttt{\textbackslash stopannotatedtyping}}\]

\[\text{bla = test oeps}
\]
\[
\text{some more}
\]
\[
\text{| another test}
\]
\[
\text{| somethingverylong oeps}
\]

In this example the / now serves as an escape character. Of course you can also use the normal backslash but then you need to use a command to specify it.

\[\text{\texttt{\textbackslash setuptyping}}\]
\[
\text{[annotatedtyping]}
\]
\[
\text{[escape=} \texttt{\textbackslash letterbackslash} \text{]}\]

Now we can say:

\[\text{\texttt{\textbackslash startannotatedtyping}}\]
\[
\text{bla = test /bgroup \sl oeps /egroup}
\]
\[
\text{/bgroup /bf some more /egroup}
\]
\[
\text{| another test}
\]
\[
\text{| somethingverylong /bgroup /it oeps /egroup}
\]
\[\text{\texttt{\textbackslash stopannotatedtyping}}\]

and get:

\[\text{bla = test oeps}
\]
\[
\text{some more}
\]
\[
\text{| another test}
\]
\[
\text{| somethingverylong oeps}
\]

You can also define an end symbol:
\setuptyping
[annotatedtyping]
[escape=//,*],
  color=darkblue]

\definestartstop
[cmt]
[style=\rm\bf]

Here the // starts the annotation and * ends it.

\startannotatedtyping
bla = test       // \black // \cmt{oops} *
               // \black // \cmt{some more} *
   | another test
   | somethingverylong // \black // \cmt{oops} *
\stopannotatedtyping

Contrary to the first example, all text in the annotation is treated as \TeX input:

\begin{verbatim}
bla = test       // oeps
                // some more
   | another test
   | somethingverylong // oeps
\end{verbatim}

You can consider using more balanced tagging, as in:

\startannotatedtyping
bla = test       // \black // \cmt{oops} *
               // \black // \cmt{some more} *
   | another test
   | somethingverylong // \black // \cmt{oops} *
\stopannotatedtyping

Watch how we limit the annotation to part of the text:

\startannotatedtyping
bla = test       << \rm\bf first >> test
                << \rm\bf second >> test
   | test
   | somethingverylong << \rm\bf fourth >> test
\stopannotatedtyping

The test a the end of the lines is verbatim again.

\begin{verbatim}
bla = test       << \rm\bf first >> test
\end{verbatim}
If no end symbol is given, the end of the line is used instead:

\setuptyping
[annotatedtyping]
[escape={//,},
color=darkblue]

Watch out: here we use {\///\}, and not just {\//\} (which would trigger the escaped variant).

\startannotatedtyping
bla = test // \black // \cmt{oeps}
| test // \black // \cmt{some more}
| somethingverylong // \black // \cmt{oeps}
\stopannotatedtyping

The result is:

bla = test // oeps
| test // some more
| somethingverylong // oeps

This can also be done easier by abusing the style option of \cmt:

\definestartstop
[cmt]
[color=black,
style=\black // \rm\bf\space]

When we give:

\startannotatedtyping
bla = test // \cmt{oeps}
| test // \cmt{some more}
| somethingverylong // \cmt{oeps}
\stopannotatedtyping

We get:
For cases like this, where we want to specify a somewhat detailed way to deal with a situation, we can use processors:

\defineprocessor
[escape]
[style=bold, color=black, left=, right=]

The previous definition of the annotation now becomes:

\setuptyping
[annotatedtyping]
[escape=escape >{//,}, color=darkblue]

This time no commands are needed in the annotation:

\startannotatedtyping
bla = test // first

// second

| test
| somethingverylong // fourth
\stopannotatedtyping

The processor is applied to all text following the //. Spaces before the text are stripped.

As some characters are special to \TeX, sometimes you need to escape the boundary sequence:

\defineprocessor
[myescape]
[style=\rm\tf,

\footnote{More mechanisms in Con\TeX\ MkIV will use that feature.}
color=black]
\setuptyping
[annotatedtyping]
[escape=myescape->{\letterhash\letterhash,},
color=darkgreen]

All text between the double hashes and the end of the line is now treated as annotation:

\startannotatedtyping
bla = test
  # first \bf test
  # second \sl test
  | test
  | somethingverylong # third \it test
\stopannotatedtyping

So we get:

bla = test
  first test
  second test
  | test
  | somethingverylong third test

We can beautify \TeX commenting as follows:

\defineprocessor[comment]
[style=\rm,
color=black,
left={\ttf\letterpercent\space}]
\setuptyping
[annotatedtyping]
[escape=comment->{\letterpercent\letterpercent,},
color=darkblue]

Here the double comments are turned into a single one and the text after it is typeset in a regular font:

\startannotatedtyping
bla = test
  % first \bf test
  % second \sl test
  | test
  | somethingverylong % third \it test
\stopannotatedtyping
This gives:

<table>
<thead>
<tr>
<th>bla = test</th>
<th>% first test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% second test</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>test</td>
<td>% third test</td>
</tr>
<tr>
<td>somethingverylong</td>
<td></td>
</tr>
</tbody>
</table>

It is possible to define several escapes. Let’s start with the delimited variant:

\defineprocessor
[escape_a]
[style=bold, color=darkred, left=(), right=)]

\defineprocessor
[escape_b]
[style=bold, color=darkgreen, left=(), right=)]

\setuptyping
[annotatedtyping]
[escape={escape_a >[[[],]],escape_b >[[(),]]},
          color=darkblue]

We can now alternate comments:

\startannotatedtyping
bla = test          [[ first ]] test [[( first )]]
| test               [[ second ]] test [[( second )]]
| somethingverylong  [[ fourth ]] test [[( fourth )]]
\stopannotatedtyping

When typeset this looks as follows:

<table>
<thead>
<tr>
<th>bla = test</th>
<th>(first) test (first)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(second) test (second)</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>test</td>
<td></td>
</tr>
<tr>
<td>somethingverylong</td>
<td>(fourth) test (fourth)</td>
</tr>
</tbody>
</table>

The line terminated variant can also have multiple escapes.
So this time we have two ways to enter regular \TeX mode:

\begin{verbatim}
\startannotatedtyping
bla = test !bf one {\em again} !bs two {\em again}
| test
| somethingverylong !bf three {\em again}
\stopannotatedtyping
\end{verbatim}

These somewhat meaningful tags result in:

\begin{verbatim}
bla = test one again
two again
| test
| somethingverylong three again
\end{verbatim}
A not so widely known feature of the verbatim handler in \CONTEXT\ is the ability to add comments in another style and \MKIV\ even offers a bit more. Here some examples are shown.

Annotating verbatim content is done using a mechanism called escaping. For such special cases it’s often best to define a specific instance.

bla = test /bgroup /sl oeps /egroup
In this example the \texttt{\textbackslash} now serves as an escape character. Of course you can also use the normal backslash but then you need to use a command to specify it.

\begin{verbatim}
\startbuffer[setup]
  \setuptyping[annotatedtyping]
  \[option=TEX]\getbuffer[define]
\stopbuffer
\typebuffer[define,example][option=TEX]\getbuffer[define]
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
\end{verbatim}

Now we can say:

\begin{verbatim}
\startbuffer[example]
  \startannotatedtyping
  bla = test \bgroup \sloeps \egroup \bgroup \bf some more \egroup
  | another test \bgroup \it oeps \egroup
  | somethingverylong \bgroup \it oeps \egroup
\stopannotatedtyping
\stopbuffer
\typebuffer[example][option=TEX]\getbuffer[setup]
\end{verbatim}

and get:

\begin{verbatim}
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground
\end{verbatim}

You can also define an end symbol:

\begin{verbatim}
\startbuffer[setup]
  \setuptyping
\end{verbatim}
source code of this document

```latex
\begin{verbatim}
[annotatedtyping]
[escape=FLICT, color=darkblue]
\definestartstop
  [cmt]
  [style=\rm\bf]
\stopbuffer
\typebuffer[setup][option=TEX] \getbuffer[setup]
\end{verbatim}

Here the \texttt{\textbackslash \type{//}} starts the annotation and \texttt{\type{**}} ends it.

\begin{verbatim}
\startbuffer
  \example
  \startannotatedtyping
  bla = test // \black // \cmt{oeps} *
  // \black // \cmt{some more} *
  | another test
  | somethingverylong // \black // \cmt{oeps} *
\stopannotatedtyping
\stopbuffer
\end{verbatim}

\begin{verbatim}
\typebuffer[example][option=TEX]
\end{verbatim}

Contrary to the first example, all text in the annotation is treated as \texttt{\TEX} input:

\begin{verbatim}
\starttextbackground
  \example
  \getbuffer[example]
\stoptextbackground
\end{verbatim}

You can consider using more balanced tagging, as in:

\begin{verbatim}
\startbuffer
\setuptyping
  [annotatedtyping]
  [escape=<<,>>, color=darkblue]
\stopbuffer
\end{verbatim}

Watch how we limit the annotation to part of the text:

\begin{verbatim}
\typebuffer[example][option=TEX]
\end{verbatim}

\end{verbatim}
```
source code of this document

\begin{verbatim}
bla = test \<< \textbf{first} \>> test
  \| \| test
  \| somethingverylong \<< \textbf{fourth} \>> test
\end{verbatim}

\begin{verbatim}
\typebuffer[example][option=TEX]
\end{verbatim}

The $\texttt{\{test\}}$ a the end of the lines is verbatim again.

\begin{verbatim}
\starttextbackground
\[example\]
\getbuffer[example]
\stoptextbackground
\end{verbatim}

If no end symbol is given, the end of the line is used instead:

\begin{verbatim}
\startbuffer
\setup
\setuptyping
  \[annotatedtyping\]
  \[escape={//},\],
  \[color=darkblue\]
\stopbuffer
\typebuffer[setup][option=TEX] \getbuffer[setup]
\end{verbatim}

Watch out: here we use $\texttt{\{\{\}}$ and not just $\texttt{\{\}}$ (which would trigger the escaped variant).

\begin{verbatim}
\definestartstop[cmt][style=\textbf{rm}]
\startbuffer[example]
\startannotatedtyping
bla = test \black \cmt{oeps}
  \| \| test
  \| somethingverylong \black \cmt{oeps}
\stopannotatedtyping
\stopbuffer
\typebuffer[example][option=TEX]
\end{verbatim}

The result is:

\begin{verbatim}
\starttextbackground[example]
\getbuffer[example]
\end{verbatim}
This can also be done easier by abusing the \texttt{\texttt{type \{style\} \texttt{option} of \texttt{\texttt{type \{cmt\}:}}

\startbuffer[setup]
definestartstop
  [cmt]
  [color=black, style=\black \rm\bf\space]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]

When we give:

\startbuffer[example]
\startannotatedtyping
bla = test \texttt{// \texttt{cmt\{oeps\}}
  \texttt{// \texttt{cmt\{some more\}}
    | test
    | somethingverylong \texttt{// \texttt{cmt\{oeps\}}
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]

We get:

\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground

For cases like this, where we want to specify a somewhat detailed way
to deal with a situation, we can use processors: \footnote{More mechanisms in \CONTEXT\ \MKIV\ will use that feature.}

\startbuffer[setup]
defineprocessor
  [escape]
  [style=\bold, color=black, left=\texttt{)}\texttt{,right=\texttt{)}]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
source code of this document

The previous definition of the annotation now becomes:

```
\startbuffer[setup]
\setup_typing
  [annotated_typing]
  [escape=escape->{///,},
    color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```

This time no commands are needed in the annotation:

```
\startbuffer[example]
\start_annotated_typing
bla = test  // first
      // second
    | test
    | somethingverylong  // fourth
\stop_annotated_typing
\stopbuffer

\typebuffer[example][option=TEX]
```

The processor is applied to all text following the `\type {///}`. Spaces before the text are stripped.

```
\start_text_background[example]
  \getbuffer[example]
\stop_text_background
```

As some characters are special to `\TEX`, sometimes you need to escape the boundary sequence:

```
\startbuffer[setup]
\define_processor
  [myescape]
  [style=\rm\tf,
    color=black]
\setup_typing
  [annotated_typing]
  [escape=myescape->{\\letterhash\\letterhash,},
    color=darkgreen]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]
```
All text between the double hashes and the end of the line is now treated as annotation:

\startbuffer[example]
\startannotatedtyping
bla = test  # first \bf test
     | test  # second \sl test
     | somethingverylong  # third \it test
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]

So we get:

\starttextbackground[example]
\getbuffer[example]
\stoptextbackground

We can beautify \TEX\ commenting as follows:

\startbuffer[setup]
\defineprocessor
  [comment]
  [style=\rm,
    color=black,
    left={\tt\ttt\letterpercent\space}]
\setuptyping
  [annotatedtyping]
  [escape=comment >{\letterpercent\letterpercent,},
    color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]

Here the double comments are turned into a single one and the text after it is typeset in a regular font:

\startbuffer[example]
\startannotatedtyping
bla = test  \% first \bf test
     | test  \% second \sl test
     | somethingverylong  \% third \it test
source code of this document

\stopannotatedtyping
\stopbuffer
\typebuffer[example][option=TEX]

This gives:
\starttextbackground[example]
  \getbuffer[example]
\stoptextbackground

It is possible to define several escapes. Let's start with the delimited variant:

\startbuffer[setup]
\defineprocessor[escape_a]
  [style=bold, color=darkred, left=, right=]
\defineprocessor[escape_b]
  [style=bold, color=darkgreen, left=, right=]
\setuptyping[annotatedtyping]
  [escape={escape_a ->{[[,]]},escape_b ->{[(),]}},
   color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]

We can now alternate comments:

\startbuffer[example]
\startannotatedtyping
bla = test
  [[ first ]] test [[ first ]]
  [[ second ]] test [[ second ]]
  | test
  | somethingverylong [[ fourth ]] test [[ fourth ]]
\stopannotatedtyping
source code of this document

\stopbuffer

\typebuffer[example][option=TEX]

When typeset this looks as follows:

\starttextbackground
\getbuffer[example]
\stoptextbackground

The line terminated variant can also have multiple escapes.

\startbuffer
\setup[example]
\defineprocessor
[annotated_bf]
[style=\rm\bf, color=darkred]
\defineprocessor
[annotated_bs]
[style=\rm\bs, color=darkyellow]
\setuptyping
[annotated_typing]
[escape={annotated_bf ->{bf,}, annotated bs ->{bs,}}, color=darkblue]
\stopbuffer

\typebuffer[setup][option=TEX] \getbuffer[setup]

So this time we have two ways to enter regular \TeX\ mode:

\startbuffer[example]
\startannotatedtyping
bla = test !bf one {\em again}
| test !bs two {\em again}
| somethingverylong !bf three {\em again}
\stopannotatedtyping
\stopbuffer

\typebuffer[example][option=TEX]

These somewhat meaningful tags result in:

\starttextbackground[example]
source code of this document

\getbuffer[example]
\stoptextbackground

\setups [listing] \setups [lastpage] \stoptext