

`\futurelet` $\langle control\ sequence \rangle$ $\langle token_1 \rangle$ $\langle token_2 \rangle$

This command tells T_EX to make $\langle token_2 \rangle$ the meaning of $\langle control\ sequence \rangle$ (as would be done with `\let`), and then to process $\langle token_1 \rangle$ and $\langle token_2 \rangle$ normally. `\futurelet` is useful at the end of macro definitions because it gives you a way of looking beyond the token that T_EX is about to process before it processes it.

Example:

```
\def\predict#1{\toks0={#1}\futurelet\next\printer}
% \next will acquire the punctuation mark after the
% argument to \predict
\def\printer#1{A \punc\ lies ahead for \the\toks0. }
\def\punc{%
  \ifx\next;semicolon\else
    \ifx\next,comma\else
      ‘\next’\fi\fi}
\predict{March}; \predict{April}, \predict{July}/
```

produces:

A semicolon lies ahead for March. A comma lies ahead for April. A “/” lies ahead for July.