

## Paweł Jackowski

\relax ex machina

Guess what is the meaning of macros in the following cases:

```
\edef\stra{\csname undefined\endcsname}
\edef\strb{\ifnum0=1\else\fi}
\edef\strc{\ifnum0=0\else\fi}
\edef\strd{\relax}

\meaning\stra % -> \undefined (plus side effect alike \let\undefined\relax)
\meaning\strb % -> (empty)
\meaning\strc % -> \relax
\meaning\strd % -> \relax
```

...and can you explain why `\ifx\strc\strd` is false, although both control sequences have the same meaning?

```
\ifx\strc\strd true\else false\fi \message{?\strc?\strd?}
```

Since the behavior of is somewhat weird, I've learned to dislike the acting as the universal string delimiter in cases such as

```
\def\gobbler#1\relax{}
\expandafter\gobbler\strc whatever \relax

\def\iterator#1{%
  \ifx\relax#1\else\message{Can you see that #1?!}%
  \expandafter\iterator\fi}
\expandafter \iterator \stra whatever \relax
```

Instead I propose

```
\def\endstr{\noexpand\endstr}
```

- no endless loop in spite of recursion (expands to itself)
- alike `\relax`, `\endstr` returns nothing (a sort of...) if typeset
- side effect is that `\endstr` stops assignments (as `\relax` do)
- `\ifx\endstr\relax` is false, so we can distinguish them
- but be careful: `\if\endstr\relax` is true