


1 先ずはおさらいから

- `\global\def` と `\gdef` が相同
- `\global\edef` と `\xdef` が相同
- `\let\bgroup={`
- `\let\egroup=}`
- `\begingroup` と `\endgroup` は primitive

2 黄色本 1.4

- `\globaldefs` は 0, 正, 負で挙動が変わる話。
正だといつでも global
負だといつでも local
0だといつもの挙動



<問題>

```
\globaldefs=-1
\def\macroA{AAA}\def\macroB{ZZZ}
{\globaldefs=1
\def\macroA{PPP}\def\macroB{QQQ}
}
\globaldefs=-1
\macroA , \macroB
```

の出力結果は。

答え。「PPP, QQQ」。となる。`\def` した瞬間の `\globaldefs` の値を見ていることがわかる。

- `\expandafter` でグループ終了をごまかすトリックについて、本に載っている例で解説。
- 一旦 `\relax` に変えることで展開を抑制するトリックについて、本に載っている例で解説。

3 八街：第2章

- とりあえず `\@for`, `\@tfor` の紹介。`\@for` はリストの中身を展開してから処理するが, `\@tfor` はリストの中身を展開せずに処理する, という違いがある。
- 実験してみたところ, `\@for` は一回までしか展開しないようだ
つまり, リストを展開するとはいえ, `\@for` の内部実装は `\edef` 系では**ない**, ということ。

```

1115 \def\@nnil{\@nil}
1116 \def\@empty{}
1117 \long\def\@fornoop#1\@#2#3{}
1118 \long\def\@for#1:=#2\do#3{%
1119   \expandafter\def\expandafter\@fortmp\expandafter{#2}%
1120   \ifx\@fortmp\@empty \else
1121     \expandafter\@forloop#2,\@nil,\@nil\@#1{#3}\fi
1122   \long\def\@forloop#1,#2,#3\@#4#5{\def#4{#1}\ifx #4\@nnil \else
1123     #5\def#4{#2}\ifx #4\@nnil \else#5\@forloop #3\@#4{#5}\fi\fi}
1124   \long\def\@iforloop#1,#2\@#3#4{\def#3{#1}\ifx #3\@nnil
1125     \expandafter\@fornoop \else
1126     #4\relax\expandafter\@iforloop\fi#2\@#3{#4}}

```

- 八街に載っている例の解説。
- この章で、八街に誤植は見当たらず。