첫 숙제에 관하여

신현 님의 질문

첫 글자만 대문자로 바꾸는 방법을 문의하셨습니다.

TEX과 LATEX의 방법 단어가 인자로 들어왔을 때 첫글자와 그 뒷글자로 분리하는 전통적 방법은 다음과 같습니다.

```
\def\mytmpfn#1#2\end{\def\tmpA{#1}\def\tmpB{#2}}
\def\headandtail#1{\expandafter\mytmpfn #1\end}

\headandtail{dream}
[\tmpA] (\tmpB)
```

[d] (ream)

\uppercase를 사용할 때는 그 뒤에 letter가 아니면 효과가 없습니다. 따라서 매크로가 들어올 적에 이를 확장해주어야 하는데 그러면 대략 다음과 같은 모양이 됩니다.

```
\def\mytmpfn#1#2\end{\def\tmpA{#1}\def\tmpB{#2}}
\def\headandtail#1{\expandafter\mytmpfn #1\end}

\headandtail{dream}
\expandafter\uppercase\expandafter{\tmpA} (\tmpB)
```

D (ream)

일찍부터 이것이 매우 불편하여, LATEX에서 \MakeUppercase를 정의해두고 있습니다. \expandafter를 자동으로 해주는 것이라고 생각하면 됩니다.

```
\def\mytmpfn#1#2\end{\def\tmpA{#1}\def\tmpB{#2}}
\def\headandtail#1{\expandafter\mytmpfn #1\end}

\headandtail{dream}
\MakeUppercase{\tmpA} (\tmpB)
```

D (ream)

그런데, 또 생각해보면 \MakeUppercase는 하나의 인자를 취하여 그것을 확장하도록 정의되어 있으므로, 위와 같이 미리 분리해서 넘길 것 없이 그냥 한 개의 인자만 전달되도록 하면 될 것입니다. (\uppercase는 이것이 되지 않습니다.) 다음 두 결과를 비교하여 봅시다.

```
\MakeUppercase dream \\
\MakeUppercase{dream}
```

Dream
DREAM

expl3의 방법 김정우 님이 제출하신 답안에 첫 글자만 분리해내어 처리하는 방법이 적용되어 있습니다.

첫 글자만 분리하는 방법으로서,

```
\ExplSyntaxOn
\tl_set:Nn \l_tmpa_tl { dream }
[ \tl_head:N \l_tmpa_tl ] ( \tl_tail:N \l_tmpa_tl )
\ExplSyntaxOff
```

[d](ream)

`\tl_head:N`과 `\tl_head:n`의 차이에 대하여 생각해보세요.

```
\tl_head:n { dream }
\tl_head:N \l_tmpa_tl
\tl_head:n { \l_tmpa_tl }
```

그러므로 첫 글자에 대해서만 조작을 가하려면

```
\ExplSyntaxOn
\tl_set:Nn \l_tmpa_tl { dream }
\MakeUppercase { \tl_head:N \l_tmpa_tl } ~(\tl_tail:N \l_tmpa_tl)
\ExplSyntaxOff
```

D (ream)

이것으로 좋다는 것을 알 수 있습니다. `\MakeUppercase`는 인자를 먼저 확장하도록 되어 있다는 점을 지적하였습니다.

한편, expl3에는 대문자나 소문자로 변환하는 함수가 미리 준비되어 있습니다.

`\tl_upper_case:n`, `\tl_lower_case:n`, `\tl_mixed_case:n`.

이 세 명령은 이름에는 `n`이라고 되어 있지만 기본적으로 인자를 먼저 확장하는 특별한 함수들입니다. 그렇기 때문에 `o`, `x` 등의 확장형이 따로 없습니다. `\tl_mixed_case:n`을 사용하여 답안을 제출한 분이 이재호 님입니다.

str_함수 실제 case 관련 함수는 `\char_upper_case:N`과 `\str_upper_case:n`이 더 있습니다. `char_` 종류의 명령은 expl3 내부적으로 사용되는 경우가 많으므로 크게 신경쓸 것이 없고, `\str_upper_case:n`이 문제인데, 이것은 주로 매크로나 함수 이름의 case에 신경써야 하는 프로그래머를 위한 명령입니다. 예컨대,

```
\ExplSyntaxOn
\str_upper_case:n { \tmp }
\ExplSyntaxOff
```

\TMP

이와 같이 매크로 이름의 케이스를 바꿀 수 있습니다. 또한 두 문자열을 비교하려 할 때 case를 무시하고 비교할 수 있도록 하는 `\str_fold_case:n` 같은 것도 역시 프로그래밍을 위해서 필요한 함수입니다.

이런 상황이 아니라면 `\str_...` 역시 지금 당장은 크게 신경쓸 것이 없습니다. 지금의 예와 같이 사용자 문자열의 케이스를 문제삼을 적에는 `\tl_<upper|lower|mix>_case:n` 이 가장 적당합니다.

결어

첫 숙제를 성실하게 제출해주셔서 매우 기쁩니다. 그리고 여러 가지 방법을 볼 수 있어서 즐거웠습니다. 앞으로도 (혹시 잘 모르겠는 것이 있더라도) 시도해본 흔적을 보여주시면 잘 모르겠는 부분을 설명할 수 있게 될 것입니다.

이 문서와 숙제 전체에 관하여 만나서 얘기하는 날 더 질문/답변할 수 있을 것입니다.

제출 답안

이재호

```
\ExplSyntaxOn
\cs_new:Npn \hello_fn_one:n #1
  {
    Hello, ~ \textit{#1} !
  }

\NewDocumentCommand \hellocmdone { m }
  {
    \hello_fn_one:n { #1 }
  }

\cs_new:Npn \hello_fn_two:n #1
  {
    Hello, ~ \tl_mixed_case:n { #1 } !
  }

\NewDocumentCommand \hellocmdtwo { m }
  {
    \hello_fn_two:n { #1 }
  }
\ExplSyntaxOff

\hellocmdone{jaeho} \\
\hellocmdtwo{jaeho}.
```

Hello, *jaeho*!
Hello, Jaeho!.

박승원

```
%\usepackage{stringstrings} %%% preamble

\ExplSyntaxOn
\NewDocumentCommand \hellocmdit { m }
{
    Hello, ~ \textit{#1} !
}

\NewDocumentCommand \hellocmdcap { m }
{
    Hello, ~ \capitalizewords{#1} !
}
```

```
\ExplSyntaxOff

\hellocmdit{seungwon} \\
\hellocmdcap{seungwon}
```

Hello, *seungwon*!
Hello, Seungwon!

김정우

```
\ExplSyntaxOn

\NewDocumentCommand \hellocmdit { m }
{
    Hello, ~ \textit { #1 } !
}

\cs_new:Npn \first_upper:n #1
{
    \str_upper_case:f { \tl_head:n { #1 } } \tl_tail:n { #1 }
}

\NewDocumentCommand \hellocmdup { m }
{
    Hello, ~ \first_upper:n { #1 } !
}
\ExplSyntaxOff

\hellocmdit{jungwoo}\\
\hellocmdup{jungwoo}
```

Hello, *jungwoo*!
Hello, Jungwoo!

신현

```
\ExplSyntaxOn

\NewDocumentCommand \hellocmdemph { m }
{
    Hello, \emph{~ #1} !
}

\NewDocumentCommand \hellocmdcase { m }
```

```
{
    Hello, \uppercase { #1 }
}

\ExplSyntaxOff
\hellocmdemph{hyun}\\
\hellocmdcase{hyun}
```

Hello, *hyun*!
Hello,HYUN