

이 문서는 《예제로 배우는 expl3》(2019/08/20)에 대한 2020년 2월의 보충입니다. 연습문제를 풀어보았다면 소스를 KTUG wiki에 업로드하십시오.

lowercase, uppercase

《예제로 배우는 expl3》의 시작 부분에 주어지는 단어의 첫 글자를 대문자로 바꾸는 문제가 나온다. 그리고 `\tl_upper_case`, `\tl_lower_case`, `\tl_mixed_case` 등에 관한 설명이 나오는데, 이 함수들이 2019년 11월 이후 삭제되고 `l3text` 모듈의 것으로 대체되었기 때문에 이제 낡은 정보가 되었다. (다만 `\str_uppercase:n` 등은 `\str_uppercase:n`으로 이름이 바뀌었다.) 이제 다음 함수들을 기억하면 된다.

- `\text_lowercase:n`
- `\text_uppercase:n`
- `\text_titlecase:n`

interface 문서를 보면 알겠지만, 이와 관련되어 다음 형식의 함수들이 더 존재한다.

- `\text_lowercase:nn`
- `\text_uppercase:nn`
- `\text_titlecase:nn`

이들은 유니코드 정의에 따라 각 언어별로 케이스를 지정할 수 있게 하는 것인데, 우리 한국어와는 상관없는 유럽어 등의 문제이다. `expl3`가 유니코드 관련 기능을 확장한 것은 특기할 만한 일이지만, 여기서는 논외로 하겠다. 그리고 `\text_titlecase:n`에 대응하는 `\text_titlecase_first:n`이라는 것은 (역시 유럽어에서) `uppercase`를 처음 한 글자 이상 두 글자까지 한꺼번에 바꾸어야 하는 언어의 경우에, 첫 글자만 바꾸도록 지시하는 것이다. 영어에서라면 둘 사이에 차이가 없다.

이 대소문자 변환 함수들은 그 다음에 오는 인자를 무조건 `\text_expand`로 먼저 확장한다. 따라서 토큰 열이나 `string` 매크로가 인자로 들어오는 경우에도 성공적으로 변환이 이루어진다.

단 한 개의 토큰에 대해서 대소문자 변환을 하기 위한 `\char_uppercase:N`, `\char_str_uppercase:N` 류의 함수들이 더 있는데, (이것이 필요할 때가 있어 보이지만) 이러한 것이 있다는 것을 언급하는 것으로 그치겠다.

이제 문서 5페이지에 나오는 예제를 다음과 같이 수정하여 둔다.

<pre>\ExplSyntaxOn \str_uppercase:n { \TMP } \ExplSyntaxOff</pre>	<pre>\TMP</pre>
---	-----------------

그리고, 홍길동 군의 이름을 다음과 같이 식자하는 예를 보이자.

<pre>\ExplSyntaxOn \tl_set:Nn \l_givename_tl { kildong } \tl_set:Nn \l_familyname_tl { hong } \text_titlecase:n { \l_givename_tl },~ \text_titlecase:n { \l_familyname_tl } \ExplSyntaxOff</pre>	<pre>Kildong, Hong</pre>
--	--------------------------

99 bottles of beer

문제

“99 bottles of beer”를 expl3로 출력하여라.

초급 프로그래밍 예제로 유명한 “99 bottles of beer”는 다음과 같은 노래를 출력하라는 것이다.

99 bottles of beer on the wall, 99 bottles of beer.
Take one down and pass it around, 98 bottles of beer on the wall.

98 bottles of beer on the wall, 98 bottles of beer.
Take one down and pass it around, 97 bottles of beer on the wall.

[.....]

1 bottle of beer on the wall, 1 bottle of beer.
Take one down and pass it around, no more bottles of beer on the wall.

No more bottles of beer on the wall, no more bottles of beer.
Go to the store and buy some more, 99 bottles of beer on the wall.

다음 조건을 잘 구현하여야 한다.

- (1) 숫자(병의 수)는 99에서 1씩 줄어든다. 마지막은 0이고 다시 99로 돌아온다.
- (2) 한 병이 줄면 숫자를 1 줄여서 적어야 한다.
- (3) 1일 때는 “bottle”이고 그밖의 경우에는 “bottles”이다.
- (4) 0일 때는 “no more bottles”이다.
- (5) 0이 되면 두 번째 행이 달라진다.

이를 각종 프로그래밍 언어로 구현해둔 사이트 <http://www.99-bottles-of-beer.net/>에 가보면 수많은 언어의 예제를 볼 수 있다. 다음은 Basic 아류인 Gambas 샘플이다.

```
STATIC PUBLIC SUB Main()  
DIM X AS Integer  
DIM ProcStr AS String  
FOR X=99 TO 0 STEP -1  
ProcStr=X & " bottles of beer"  
IF X=1 THEN ProcStr="1 bottle of beer"  
IF X=0 THEN ProcStr="No more bottles of beer"  
IF X<>99 THEN PRINT "Take one down and pass it around, " & ProcStr & " on the  
↵ wall."  
PRINT ProcStr & " on the wall, " & ProcStr & "."  
NEXT  
PRINT "Go to the store and buy some more, 99 bottles of beer on the wall."  
END
```

경우를 나누어서 생각해보면,

1. 카운터가 99일 때: “99 bottles”로 시작하는 첫 행을 적고, “Take one down and pass it around,”까지 인쇄한다.

```
99 bottles of beer on the wall, 99 bottles of beer.\  
Take one down and pass it around,
```

2. 카운터가 98부터 2까지일 때: 먼저 “*n* bottles of beer on the wall.”을 한 번 적고 개행한 다음에, 이것을 두 번 더 반복한다. 그리고 “Take...”를 적는다.

```
98 bottles of beer on the wall. \par
98 bottles of beer on the wall, 98 bottles of beer.\\
Take one down and pass it around,
```

3. 카운터가 1일 때, “1 bottle of...”를 한 번 적고 행을 바꾼 다음 다시 두 번 더 적고, “Take...”를 인쇄한다.

```
1 bottle of beer on the wall. \par
1 bottle of beer on the wall, 1 bottle of beer.\\
Take one down and pass it around,
```

4. 카운터가 0일 때. “no more bottles of”를 한 번 적고 행을 바꾼 다음, 다시 두 번 더 적고, “Go to the store”로 시작하는 마지막 행을 적은 후, 개행하고 종료한다.

```
no more bottles of beer on the wall. \par
No more bottles of beer on the wall, no more bottles of beer.\\
Go to the store and buy some more, 99 bottles of beer on the wall.
```

문서 명령을 `\PrintSong`이라고 하자. 옵션 인자를 하나 받아서 최대 몇 병인지로 삼고 옵션이 없으면 99로 한다.

```
\ExplSyntaxOn
\int_new:N \g_maxbottle_int
\NewDocumentCommand \PrintSong { 0{99} }
{
  \int_gset:Nn \g_maxbottle_int { #1 }
  \int_step_function:nnnN { \g_maxbottle_int } { -1 } { 0 }
  ~ \print_the_song:n
}
```

공통적으로 쓰이는 문자열을 매크로로 정의해둔다.

```
\tl_set:Nn \ofbeer { of~beer }
\tl_set:Nn \onthewall { on~the~wall }
\tl_set:Nx \ofbeeronthewall { \ofbeer{}~\onthewall }
\tl_set:Nn \takeone { Take~one~down~and~pass~it~around }
\tl_set:Nn \gotothestore { Go~to~the~store~and~buy~some~more }
```

이제 `\print_the_song:n`이라는 함수를 정의한다. 인자는 `\int_step`문으로부터 넘어오는 인덱스 카운터이다.

```
\cs_new:Npn \print_the_song:n #1
{
  \int_case:nnF { #1 }
  {
    { 1 } { 1~bottle~\ofbeeronthewall . \par
            1~bottle~\ofbeeronthewall , ~ 1~bottle~\ofbeer . \\
            \takeone,~
          }
    { 0 } { no~more~bottles~\ofbeeronthewall . \par
            No~more~bottles~\ofbeeronthewall , ~ no~more~bottles~\ofbeer
            ~ . \\
          }
  }
}
```

```

        \gotothestore,~\int_use:N
        ↪ \g_maxbottle_int{}~bottles~\ofbeeronthewall .
    }
    { \g_maxbottle_int } {
        #1~bottles~\ofbeeronthewall , ~ #1~bottles~\ofbeer. \\
        \takeone,~
    }
}
{
    #1~bottles~\ofbeeronthewall . \par
    #1~bottles~\ofbeeronthewall , ~ #1~bottles~\ofbeer . \\
    \takeone,~
}
}
\ExplSyntaxOff

```

본문에서는 이 명령을 verse 환경 안에 넣으면 된다. 99는 너무 많고 지루하니 5개만 보이기로 한다.

```

\begin{verse}
\PrintSong[5]
\end{verse}

```

5 bottles of beer on the wall, 5 bottles of beer.
 Take one down and pass it around, 4 bottles of beer on the wall.

4 bottles of beer on the wall, 4 bottles of beer.
 Take one down and pass it around, 3 bottles of beer on the wall.

3 bottles of beer on the wall, 3 bottles of beer.
 Take one down and pass it around, 2 bottles of beer on the wall.

2 bottles of beer on the wall, 2 bottles of beer.
 Take one down and pass it around, 1 bottle of beer on the wall.

1 bottle of beer on the wall, 1 bottle of beer.
 Take one down and pass it around, no more bottles of beer on the wall.

No more bottles of beer on the wall, no more bottles of beer.
 Go to the store and buy some more, 5 bottles of beer on the wall.

연습문제

만약 `\return_a_verse:n`이라는 함수가 있어서, 인자로 주어지는 수에 해당하는 연을 출력하도록 만들고 싶다. 이 함수를 정의하고 이를 이용하여 위의 문제를 다시 해결하되, 1은 “one”으로 나타내어라. 예컨대, `\return_a_verse:n { 5 }`의 결과는

5 bottles of beer on the wall, 5 bottles of beer.
 Take one down and pass it around, 4 bottles of beer on the wall.

로 출력된다.