

## 문제

인자를 하나 받아서 “Hello, #1!” 형식으로 출력하는 명령 `\hellocmd`를 작성하여라.

입력: `\hellocmd{world}`

출력: Hello, world!

**xparse** xparse에서 다음 명령은 이미 알고 있다고 가정한다. 만약 다음 명령이 무엇을 하려는 것인지 잘 모르겠다면 xparse 문서를 반드시 숙지하여야 한다.

- `\NewDocumentCommand`.
- `\RenewDocumentCommand`.
- `\IfNoValueTF`.
- `\IfBooleanTF`.
- 인자형 지시자: m, o, O, s.

**interface3.pdf** 이 문서는 expl3의 (거의 유일하고 완전한) reference이다. 항상 열어두고 작업하는 것이 좋다. `texdoc interface3`.

이제 주어진 문제를 풀어보자.

```
\ExplSyntaxOn
\NewDocumentCommand \hellocmd { m }
{
  Hello, ~ #1 !
}
\ExplSyntaxOff
```

`\ExplSyntaxOn`과 `\ExplSyntaxOff`로 expl3 언어로 코딩하는 부분임을 표시한다. 이 범위 안에서는

1. 스페이스는 모두 무시된다.
2. `:`과 `_`가 명령의 일부로 쓰인다(즉, letter이다).

그러므로 행끝 문자를 없애기 위해서 %를 행 끝에 붙여야 했던 불편이 없다. 그 대신 스페이스를 입력 문자 열에 남겨야 할 적에 반드시 ~ (틸데)로 그것을 명시적으로 표시해주어야 한다.

위의 명령을 실행하면 다음처럼 된다.

`\hellocmd{world}`

Hello, world!

이번에는 다음과 같은 코드를 생각해보자.

```

\ExplSyntaxOn
\cs_new:Npn \hello_fn:n #1
{
    Hello,~ #1 !
}

\NewDocumentCommand \hellocmd { m }
{
    \hello_fn:n { #1 }
}
\ExplSyntaxOff

```

이 코드는 문서 명령이 어떤 것인지를 명백하게 보여준다. 문서 명령 `\hellocmd`가 하는 일은 인자를 하나 받아서 `expl3`로 작성된 “함수” `\hello_fn:n`에 넘겨주는 일만을 한다. 실제로 어떤 작용을 하는 것은 이 함수가 하게 되는 것이다. 그러나 `\begin{document}` 이후에 `\hello_fn:n`이라는 명령을 쓸 수는 없기 때문에 문서 명령 `\hellocmd`를 정의하게 되었다.

한편, 우리가 넘겨받은 인자는 평범한 알파벳으로 이루어진 문자열(토큰열)이다. 앞으로 공부하게 될 중요한 자료형은 다음과 같은 것이 있다.

**token list** 가장 기본적인 문자열이다. 매크로나 그룹도 하나의 토큰으로 간주한다. 토큰열에서 모든 문자는 그 카테고리 코드를 보존하고 있다.

**string** 토큰열과 같지만 카테고리 코드가 모두 12(other)로 간주된다. 따라서 매크로는 그냥 문자열일 뿐이고 실행(확장)되지 않는다.

**integer** 정수형. 보통의 형검사를 하는 언어에서의 `int`와 매우 유사하다.

**floating point** . 부동소수점 실수형. 유효숫자가 16자리인 실수이다.

**dimension**  $\text{T}_{\text{E}}\text{X}$ 의 길이값 형식으로서 단위(sp, pt, mm, etc.)를 포함한다.

**skip**  $\text{T}_{\text{E}}\text{X}$ 의 skip 형식으로서 둘 이상의 dim으로 구성된다.

**keys** `<key> = <value>` 형식의 데이터.

여기에 다음과 같은 리스트 형식의 자료형이 존재한다.

**sequence** `expl3` 특유의 리스트 자료형이다. 여러 개의 아이템을 일관되게 다루기 위한 것으로 stack처럼 활용할 수 있다.

**clist** comma separated list. seq의 특별한 경우로서 각 아이템이 ,로 구분된 것이다. 간략화한 seq라고 생각하면 된다.

**property list** key=value 형식의 리스트이다.

이밖에, file, regex, sys, quark, cs, box, coffin 등의 자료형이 더 있지만 천천히 배워나갈 것이다.

연습삼아서 인자로 받은 문자열을 `tl` 변수에 넣어서 활용하는 방법을 생각해보자.

```

\cs_new:Npn \hello_fn:n #1
{
  \tl_set:Nn \l_tmpa_tl { #1 }
  \l_tmpa_tl
}

```

`\cs_new:Npn`은 새로운 함수(cs)를 정의하라는 명령으로서 자주 쓰일 것이므로 이 문법을 잘 보아두어야 한다. 이 자체가 하나의 함수이다. N은 control sequence, p는 parameter, n은 “중괄호로 둘러싸인 토큰열”을 가리킨다. 이 인자형 지시자는 매우 중요한데, 지금 다 알아둘 필요는 없지만 다음 몇 가지는 숙지해두자.

**n** 가장 많이 쓰이는 인자형이다. 중괄호로 묶어서 전달되는 토큰열. {...}가 하나의 n에 해당한다.

**N** 한 개의 매크로를 가리킨다. 예를 들면 `\l_tmpa_tl` 자체가 하나의 N이다.

**o, x, f, e** 인자의 확장형을 가리키는데 expansion 문제는 나중에 천천히 다루게 된다.

**V** 한 개의 매크로를 가리키지만 그 매크로 자체를 전달하는 것이 아니라 그 값(value)을 넘겨받는다는 의미이다.

따라서 `\tl_set:Nn`은 그 뒤에 차례로 한 개의 매크로와 한 개의 중괄호 범위가 올 것임을 예상할 수 있다. 이 함수의 의미는 n으로 주어지는 토큰열을 N이라는 매크로에 할당하라는 것이다. 당연히 N은 tl 자료형이어야 한다.

이 때 `\l_tmpa_tl`은 어떤 데이터를 저장할 매크로인데, 이를 “변수”라고 한다. 변수는 어떤 자료형인지를 명시하는 것이 좋다. l은 이것이 지역변수임을 의미하고 tmpa는 고유한 이름이며 tl은 데이터 타입이다.

이름이 tmpa인 것과 tmpb인 것은 expl3 자체에 정의된 scratch 변수이다. 즉 이 변수명을 선언하지 않아도 바로 사용할 수 있다. 스크래치 변수는 l과 g에 대하여 두 개씩 있어서 모두 네 개가 각 데이터별로 정의되어 있다. 자신만의 변수를 만들려면 `\tl_new:N`으로 변수를 선언해야 할 때가 있다.

위의 코드에서 둘째 줄 `\l_tmpa_tl`은 그 자체로 그 위치에 이 변수의 내용을 식자한다. 이렇게 하는 것을 “입력문자열에 남긴다”고 표현한다. 그리고 그것이 이 함수의 반환값 비슷한 것이 된다는 점에 유의하라. expl3는 명시적인 반환값은 별도로 존재하지 않고(따라서 엄밀한 의미의 function이 아니다), 입력 문자열에 어떤 것을 남김으로써 반환값처럼 처리하게 한다. 요컨대, expl3에는 return 문이 없다.

## 용어

**문서 명령** Expl3로 작성하는 control sequence들은 \_(underscore)와 :(colon)을 포함하므로 문서 중에 이를 사용할 수 없다. expl3를 이용하여 작성한 함수를 문서 중에 사용하기 위하여 만드는 매크로를 “문서 명령(document commands)”라고 부른다. 문서 명령을 작성하도록 도와주는 것이 xparse이며 expl3와 필수적으로 함께 쓰여야 한다.

**입력 문자열 input stream.** TeX에게 전해주는 토큰(열)을 가리키는 말로 쓴다.

**인자형 지시자** expl3 함수의 일부로서 그 함수가 흡수(absorb)할 인자의 형식을 미리 지정한다. 참고로, xparse의 인자형 지시자(m, o, 등)는 expl3의 인자형 지시자와 기능과 목적이 다르므로 혼동하면 안 된다.

**자료형** expl3의 data type.

**함수** 인자를 취하여 일정한 작용을 하도록 정의한 expl3의 매크로(데이터) 형식

**변수** 데이터를 저장하고 있는 매크로

범위 `\begingroup`에서 `\endgroup`까지, 즉 `{`에서 `}`까지를 가리킨다. 지역(local) 변수는 이 범위 내에  
서만 유효하다. 범위와 상관없이 문서 전체에 유효한 변수를 전역적(global)이라고 한다.

#### 연습문제

**기본** 1. 문제에서 제시한 `\hellocmd`에서 인자로 주어진 이름을 이탤릭체로 식자하도록 해보아라.

**발전** 2. 문제에서 제시한 `\hellocmd`를, 넘어온 인자의 첫 글자를 대문자로 바꾸어 출력하도록  
작성하여라.

입력: `\Hellocmd{world}`

출력: Hello, *world*! 또는 Hello, World!