KTUG 사설저장소의 ks*
패키지와 LaTeX3 패키지 제작

2015.01.31

Nova De Hi

차 례

LaTeX 코딩의 세 측면

1. **문서 마크업 인터페이스**. 일종의 태그로서 commands와 environments. 실제 문서를 작성하는 과정에서 문단의 유형, 장절의 구조 등을 마크업하는 것.

LaTeX 코딩의 세 측면

1. **문서 마크업 인터페이스**. 일종의 태그로서 commands와 environments. 실제 문서를 작성하는 과정에서 문단의 유형, 장절의 구조 등을 마크업하는 것.
2. **디자인 인터페이스**. 문서 마크업의 의미론적(semantic) 마크업이 실제로 출력될 형태를 디자인하기 위한 인터페이스.

LaTeX 코딩의 세 측면

1. **문서 마크업 인터페이스**. 일종의 태그로서 commands와 environments. 실제 문서를 작성하는 과정에서 문단의 유형, 장절의 구조 등을 마크업하는 것.
2. **디자인 인터페이스**. 문서 마크업의 의미론적(semantic) 마크업이 실제로 출력될 형태를 디자인하기 위한 인터페이스.
3. **프로그래밍 인터페이스**. TeX으로 성립하는 프로그래밍 언어로서의 측면. Expl3는 LaTeX 또는 TeX 위에 구축한 프로그래밍 언어이다.

Programming Language와 LaTeX

Pakin은 프로그래밍 언어와 TeX을 연결하는 네 가지 시도에 대하여 말한 바 있다. 여기에 한 가지 더 추가하여 다음 다섯 가지 방법을 살펴볼 수 있다.

Programming Language와 LaTeX

Pakin은 프로그래밍 언어와 TeX을 연결하는 네 가지 시도에 대하여 말한 바 있다. 여기에 한 가지 더 추가하여 다음 다섯 가지 방법을 살펴볼 수 있다.

1. `\write18`을 이용하는 방법 (e.g. python package)
2. TeX 엔진에 언어 인터프리터를 내장하는 방법 (e.g. LuaTeX)
3. 외부 인터프리터를 활용하는 매크로를 작성하는 방법 (e.g. PerlTeX)
4. TeX 매크로 자체로 새로운 언어를 만드는 방법 (e.g. Expl3)
5. TeX 매크로로 언어 인터프리터를 구현하는 방법 (e.g. basix, lisp-on-tex)

LaTeX3과 expl3

- LaTeX3 = The future version of LaTeX

LaTeX3과 expl3

- LaTeX3 = The future version of LaTeX
- Expl3 = LaTeX3의 프로그래밍 레이어

LaTeX3과 expl3

- LaTeX3 = The future version of LaTeX
- Expl3 = LaTeX3의 프로그래밍 레이어
- Expl3 = A new **programming language** based on TeX

LaTeX3과 expl3

- LaTeX3 = The future version of LaTeX
- Expl3 = LaTeX3의 프로그래밍 레이어
- Expl3 = A new **programming language** based on TeX
- ConTeXt나 plain TeX의 expl3 인터페이스

LaTeX3과 expl3

- LaTeX3 = The future version of LaTeX
- Expl3 = LaTeX3의 프로그래밍 레이어
- Expl3 = A new **programming language** based on TeX
- ConTeXt나 plain TeX의 expl3 인터페이스
- ∴ Expl3 ⊄ LaTeX3

LaTeX 매크로 작성의 어려움

- LaTeX 또는 TeX 매크로는 programming language이면서 동시에 document-level formatting markup인 데서 오는 혼선
- 극도로 낮은 readability
- space 간섭과 %의 남용
- API가 제공되지 않음
- expansion control의 어려움: n번째 인자를 미리 확장하려면 최대 $2^n - 1$개의 `\expandafter`를 써야함.
- 변수를 이용하여 매크로 명칭을 구성하는 것의 번거로움: `\csname ... \endcsname`
- LaTeX 2_ε의 "패키지"는 LaTeX 활용에 중대한 기여를 하였지만 패키지 간의 충돌과 중복 정의가 점점 문제로 떠오름.

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용
- 함수, 변수, 상수의 구분

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용
- 함수, 변수, 상수의 구분
- 함수명은 인자의 종류와 수를 미리 밝힘

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용
- 함수, 변수, 상수의 구분
- 함수명은 인자의 종류와 수를 미리 밝힘
- expansion control API 제공

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용
- 함수, 변수, 상수의 구분
- 함수명은 인자의 종류와 수를 미리 밝힘
- expansion control API 제공
- 변수의 scope (local/global)

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용
- 함수, 변수, 상수의 구분
- 함수명은 인자의 종류와 수를 미리 밝힘
- expansion control API 제공
- 변수의 scope (local/global)
- 변수의 data type: bool, box, int, fp, ...

Expl3 언어의 특징

- white space를 모두 무시 : %를 쓸 필요가 없음
- underscore(_)와 colon(:) 사용
- 함수, 변수, 상수의 구분
- 함수명은 인자의 종류와 수를 미리 밝힘
- expansion control API 제공
- 변수의 scope (local/global)
- 변수의 data type: bool, box, int, fp, ...
- 프로그래밍을 위한 API 제공

Data Types

bool	논리 연산을 위한 자료형. true or false.
box	박스 레지스터.
coffin	연결점이 있는 박스.
dim	고정 길이값.
skip	가변 길이값.
fp	floating point.
int	정수, 카운터.
muskip	수학 모드 가변 길이값.
ior, iow	input/output stream (파일)
tl	token list
clist	comma-seperated list
seq	sequence
prop	속성이 있는 list

그밖의 modules

exp	expansion control
quark	quark과 scan marks
str	string
keys	key-value data type
galley	rectangular area containing texts or other stuffs.

cf. `http://latex-project.org/svnroot/experimental/trunk/`

Expl3의 LaTeX 인터페이스

```
1 \usepackage{expl3}
```

Expl3의 LaTeX 인터페이스

1. `\usepackage{expl3}`
2. `\ExplSyntaxOn`, `\ExplSyntaxOff`

Expl3의 LaTeX 인터페이스

1. `\usepackage{expl3}`
2. `\ExplSyntaxOn`, `\ExplSyntaxOff`
3. LaTeX 사용자 인터페이스 명령의 제공 : xparse
   - `\usepackage{xparse}`
   - `\NewDocumentCommand`
   - `\IfBoolean(TF)`
   - `\IfNoValue(TF)`
   - Splitting and Mapping Arguments

차 례

KTUG 사설저장소의 ks* 패키지

패키지의 제작에 Expl3을 활용한 KTUG 사설저장소의 패키지

jiwonlipsum	한글/한자 채우기 텍스트
hanjacnt	한자/한글식 숫자 읽기
graphicsonthefly	web상의 그림 다운로드 포함
ifpxltex	엔진별 확장
kocircnum	원숫자
ksforloop	for-loop
ksmisc	kslinematters, ksmisc, preparefont
kotex-sections	HLaTeX 식의 절 표제
hangulfontset	한글 폰트 선택

jiwonlipsum

- 제작동기 : 채우기 텍스트 명령의 제공
- 주요 기능 : `\jiwon`
- 한자 : 한글+한자 혼합 텍스트, KS X 1001 범위를 넘어서는 한자.
- 텍스트 : 연암 박지원의 일야구도하기와 호곡장론 일부를 번역

jiwonlipsum

- 제작동기: 채우기 텍스트 명령의 제공
- 주요 기능: `\jiwon`
- 한자: 한글+한자 혼합 텍스트, KS X 1001 범위를 넘어서는 한자.
- 텍스트: 연암 박지원의 일야구도하기와 호곡장론 일부를 번역

하수는 두 산 틈에서 나와 돌과 부딪쳐 싸우며, 그 놀란 파도와 성난 물머리와 우는 여울과 노한 물결과 슬픈 곡조와 원망하는 소리가 굽이쳐 돌면서, 우는 듯, 소리치는 듯, 바쁘게 호령하는 듯, 항상 장성을 깨뜨릴 형세가 있어, 전차 만승과 전기 만대나 전포 만가와 전고 만좌로써는 그 무너뜨리고 내뿜는 소리를 족히 형용할 수 없을 것이다. 모래 위에 큰 돌은 홀연히 떨어져 섰고, 강 언덕에 버드나무는 어둡고 컴컴하여 물지킴과 하수 귀신이 다투어 나와서 사람을 놀리는 듯한데, 좌우의 교리가 붙들려고 애쓰는 듯싶었다. 혹은 말하기를, “여기는 옛 전쟁터이므로 강물이 저같이 우는 것이다.” 하지만 이는 그런 것이 아니니, 강물 소리는 듣기 여하에 달렸을 것이다.

jiwonlipsum

Example

```
\RequirePackage{expl3}
\ProvidesExplPackage
  {jiwonlipsum}{2014/10/20}
  {0.4}{a Korean version of (kant)lipsum}
```

\ExplSyntaxOn과 \ExplSyntaxOff를 별도로 지정할 필요가 없음.

hanjacnt

- 제작동기: `http://ktug.kldp.net/jsboard/read.php?table=ktugbd&no=4809&page=118`
- 숫자의 "읽기". 12,345 $\longrightarrow$ 만이천삼백사십오
- 한국어 숫자 읽기 관행: 1만의 '일'을 새기지 않음
- 큰 수
- 카운터
- *seq*와 *int* datatype 활용

hanjacnt

Example

```
\cs_new:Npn \hanjanum_read_two:nn #1 #2
{
   \int_compare:nF { #1 = 0 }
   {
      \int_compare:nF { #1 = 1 } { \number_to_hangul:n { #1 } }
      \_mark_sib
   }
   \int_compare:nF { #2 = 0 } {
      \number_to_hangul:n { #2 }
   }
}
```

hanjacnt

Example

```
\NewDocumentCommand \hanjacnt_hanja_number_big { m }
{
   \seq_set_split:Nnn \tmp_test_seq {} { #1 }
   \seq_remove_all:Nn \tmp_test_seq { , }
   \check_hanja_fin:
   \int_case:nn { \seq_count:N \tmp_test_seq }
   {
      { 1 } { \seq_pop_right:NN \tmp_test_seq \_tmp_dan
              \x_hanjanum_read_one:n \_tmp_dan }
```

graphicsonthefly

- url로 주어진 웹상의 그림(png, pdf, jpg, gif, bmp)을 현재 작업 중인 폴더로 다운로드하고 문서에 포함
- 그림의 다운로드는 wget 또는 curl을 실행. `--shell-escape` 필요.
- gif, bmp 그림은 ImageMagick convert 필요. png로 변환하여 포함
- animated gif는 animate 패키지를 이용하여 여러 장의 png로 분리하여 처리 (animation 효과는 Adobe Reader 필요)
- url을 처리하기 위하여 **v**-타입 옵션, regular expression replace(l3regex) 사용.

graphicsonthefly

Example

```
\DeclareDocumentCommand \prepareimgonthefly { s m v }
{
   \IfBooleanTF { #1 } { \bool_set_true:N \_exist_star }
                { \bool_set_false:N \_exist_star }
   \seq_set_split:Nnn \_test { . } { #3 }
   \seq_pop_right:NN \_test \_test_grf_type

   \regex_replace_once:nnN
      { ([bgjpBGJP][dimnpDIMNP][efgpEFGP])(.*) }
      { \1 } \_test_grf_type

   \str_case_x:nn { \_test_grf_type }
   {
      { png } { \tl_set:Nn \grf_type { png } }
```

preparefont

- 온라인 상의 폰트 파일(zip, ttf, otf)을 작업 중인 폴더로 다운로드
- `--shell-escape` 요구됨 — arara 권장
- 기본적인 설계는 graphicsonthefly와 동일함
- 폰트가 문제가 되는 MWE 작성을 위해 활용 가능

```
% arara: xelatex: { shell: yes, synctex: yes }
```

kocircnum

- 원숫자 식자를 위한 패키지. ①, ②, ③, …
- hcrcircnum (Dohyun Kim), hzmisc (Hoze), tikzcircnum (Nova De Hi)를 합친 것
- 다른 두 패키지를 Expl3 문법으로 재코딩
- *keys* datatype의 활용 (key=value data structure)

kocircnum

Example

```
\keys_define:nn { hcrsetup }
{
   shape     .tl_set:N  = \l_tmpa_circnum_shape,
   shape     .default:n = circle,
   hcrcolor  .tl_set:N  = \l_tmpa_circnum_hcrcolor,
   hcrcolor  .default:n = white,
   circle    .meta:n    = { shape    = circle    },
   box       .meta:n    = { shape    = rectangle },
   rectangle .meta:n    = { shape    = rectangle },
   white     .meta:n    = { hcrcolor = white     },
   black     .meta:n    = { hcrcolor = black     },
   reset     .code:n    = { \restorehcrcircnumsetup },
```

hangulfontset

- 한글 폰트 설정 패키지
- l3keys2e 패키지의 활용 (*keys* data type + LaTeX 2_ε 패키지 제작)

hangulfontset

Example

```
\RequirePackage{l3keys2e}
...
\keys_define:nn { hangulfontsetting }
{
   hfntset   .tl_set:N = \test_option_str,
   mjfeature .tl_set:N = \test_option_mj_feature,
   gtfeature .tl_set:N = \test_option_gt_feature,
...
\ProcessKeysOptions { hangulfontsetting }
```

ksforloop

1. plain TEX의 반복문

   `\loop ... \repeat`

2. ifthen의 반복문

   `\whiledo{<test>}{<while>}`

3. LATEX의 (또는 xfor) `\@for`
4. tikz의 `\foreach`
5. multido의 `\multido`
6. forloop의 `\forloop`

ksforloop

Expl3의 반복문

1. 정수형 반복문

   ```
   \int_do_until:(nN)nn, \int_do_while:(nN)nn
   \int_until_do:(nN)nn, \int_while_do:(nN)nn
   ```

2. 길이값형 반복문

   ```
   \dim_do_until:(nN)nn, \dim_do_while:(nN)nn
   ...
   ```

3. 실수형 반복문

   ```
   \fp_do_until:(nN)nn
   ```

4. 불린형 반복문

   ```
   \bool_do_until:Nn, \bool_do_while:Nn
   ```

ksforloop

정수형 반복문 \ksforloop을 정의

```
\ksforloop{i=1+1}{10}{%
  \ksfori,\space
}
```

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

ksforloop

정수형 반복문 \ksforloop을 정의

```
\ksforloop{i=1+1}{10}{%
  \ksfori,\space
}
```

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

주로 패키지 내부에서 사용하려는 목적으로 제작하였으며 일반 문서에서는 \multido를 권함.

ksmisc

Example

```
\NewDocumentCommand \test { m }
{
    \tl_set:Nn \l_tst { #1 }
    \ks_for_loop:nnn {}{\tl_count:N \l_tst }
    {
        \ks_pop_right:NN \l_tst \l_tmp
        \fbox{\l_tmp}
    }
}
```

```
\test{가나다라마{바사}아}
```

아	바사	마	라	다	나	가

kslinematters

space, carriage return, tab 문자를 “보이는 대로” 출력하도록 하는 명령과 환경을 제공한다. 한글 시문을 식자하거나 복사된 텍스트의 white space를 그대로 보일 때 쓸 수 있다. verbatim과 달리 행나눔과 white space를 제외하면 LaTeX 명령이 모두 동작한다.

```
\obeythem, \disobeythem
\vobeyspaces, \vobeytabs, \vobeylines
\begin{ksobeys} ... \end{ksobeys}
```

줄 끝까지를 인자로 받는 명령을 정의한다.

```
\newlinecommand
```

차 례

How to Define Macros (2012)

Hoze의 2012년 KTS Conference 발표 자료, “A Guide to How to Define Macros” 의 내용을 바탕으로 같은 일을 Expl3 또는 LaTeX3 (Expl3+xparse) 로 하면 어떻게 할 수 있는지를 보인다.

차 례

mapping functions I

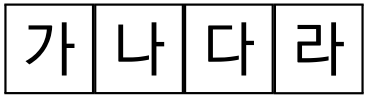
```
\seq_map_function:NN, \seq_map_inline:Nn
\clist_map_function:NN, \clist_map_inline:Nn
\tl_map_function:NN, \tl_map_inline:Nn
```

Example

mapping functions II

```
\cs_new:Npn \test_map:n #1
{
   \fbox{ #1 }
}
\NewDocumentCommand \testmap { m }
{
   \tl_set:Nn \l_tmpa_tl { #1 }
   \tl_map_function:NN \l_tmpa_tl \test_map:n
}
```

```
\testmap{가나다라}
```



Expl3의 재귀 호출: l3quark I

```
\cs_new:Npn \my_hs_letter:n #1
{
   \_my_hs_letter:n #1 \q_recursion_tail
   \q_recursion_stop
}

\cs_new:Nn \_my_hs_letter:n
{
   \quark_if_recursion_tail_stop:n { #1 }
   #1/
   \_my_hs_letter:n
}

\NewDocumentCommand \myhsletters { m }
{
```

Expl3의 재귀 호출 : l3quark II

```
    \my_hs_letter:n { #1 }
}
```

Expl3의 재귀 호출: 피보나치 수 I

```
\int_new:N \output_int

\cs_new:Npn \fibo_iter:nnn #1 #2 #3
{
    \int_compare:nTF { #1 = 0 }
    {
        \int_set:Nn \output_int { #2 }
    }
    {
        \fibo_iter:nnn { \int_eval:n { #1 - 1 } }
            { #3 } { \int_eval:n { #3 + #2 } }
    }
}

\NewDocumentCommand \fibo { m }
```

Expl3의 재귀 호출: 피보나치 수 II

```
{
    \fibo_iter:nnn { #1 } { 0 } { 1 }
    $\mathrm{F}\c_math_subscript_token {#1} =
        \int_use:N \output_int$
}
```

재귀호출: 하노이 탑 I

```
\cs_new:Npn \do_hanoi:nnnn #1 #2 #3 #4
{
    \int_compare:nT { #1 > 0 }
    {
        \do_hanoi:nnnn { #1 - 1 } { #2 } { #4 } { #3 }

        Move~disk~\int_eval:n
            {#1}~:~#2~$\longrightarrow$~#3\par

        \do_hanoi:nnnn { #1 - 1 } { #4 } { #3 } { #2 }
    }
}

\NewDocumentCommand \hanoi { m }
{
```

재귀호출: 하노이 탑 II

```
    \do_hanoi:nnnn { #1 } { A } { C } { B }
}
```

Expl3의 재귀 호출: 유클리드 호제법 I

작은나무의 코딩:

```
\newcount\m \newcount\n \newcount\t

\def\euclid#1#2{\m=#1 \n=#2
  \ifnum\n>\m \t=\m \m=\n \n=\t \fi
  \ifnum\n=0 \number\m
  \else\t=\m \divide\m\n \multiply\m\n
  \advance\t-\m \euclid\n\t \fi}
```

Expl3 코딩:

Expl3의 재귀 호출: 유클리드 호제법 II

```
\int_new:N \output_int

\cs_new:Npn \euclid_gcd:nn #1 #2
{
    \int_compare:nTF { #2 = \c_zero }
    {
        \int_set:Nn \output_int { #1 }
    }
    {
        \euclid_gcd:nn { #2 } { \int_mod:nn { #1 } { #2 } }
    }
}

\cs_new:Npn \print_gcd:nn #1 #2
{
    The~gcd~of~#1~and~#2~is~\int_use:N \output_int .\par
```

Expl3의 재귀 호출: 유클리드 호제법 III

```
}

\NewDocumentCommand \eukgcd
{ > { \SplitArgument { 1 } { ; } } m }
{
    \euclid_gcd:nn #1
    \print_gcd:nn #1
}
```

Sorting

l3sort 패키지

```
\clist_set:Nn \l_foo_clist { 3 , 01 , -2 , 5 , +1 }
\clist_sort:Nn \l_foo_clist
  {
    \int_compare:nNnTF { #1 } > { #2 }
      { \sort_reversed: }
      { \sort_ordered: }
  }
```

- 문자열 소팅 http://www.ktug.org/xe/index.php?mid=blog&document_srl=182672
- 리스트 문단 소팅 http://doeun.blogspot.kr/2015/01/blog-post_20.html

Expl3 관련 패키지 몇 개

xtemplate	document authors who are not programmers can easily change the design of their documents. xtemplate also makes it easier for LaTeX programmers to provide their own customisations on top of a pre-existing class.
xcoffin	provides user interface with expl3's *coffin* data-type.
xgalley	provides templates for the layers of *galley*s.

전혀 상관없는...

1. http://cloud.sagemath.com 클라우드 LaTeX'ing.

전혀 상관없는...

1. http://cloud.sagemath.com 클라우드 LaTeX'ing.
2. 츤데레성 질문에 대한 감상

감사합니다.