

날짜 계산 dayoftheweek.sty

Nova de Hi

2024년 5월 16일 (목요일) v0.3.2

차례

1 개요	1
2 요약	1
3 구현	2
3.1 함수	2
3.2 문서 명령	8
3.3 Change log	11

1 개요

2019년 2월 26일, KTUG 게시판에 <살아온 날은 며칠?>이라는 제목으로 올린 글에서 소개한 자작 패키지이다. 이번에 버그를 하나 수정하고 패키지 문서를 만들었는데, 1906일만이다. 이 패키지는 그후, <음력·양력 변환>을 위해서도 쓰인 바 있다. 이 글은 2020년 11월 23일자로 게시하였으니, 오늘 2024년 5월 16일 (목요일)까지 1270일이 지났다.

2 요약

패키지가 제공하는 기능은 다음과 같다.

- (1) 주어진 날짜의 요일을 계산하는 것
- (2) 주어진 날짜를 기산일(1900년 1월 1일)로부터 경과일수로 환산하는 것
- (3) 두 날짜 사이의 경과일 차이를 계산하는 것

각 명령, 함수, 변수의 용법은 3.2절 「문서 명령」을 참고하라. 여기서는 실제 문서작성에 사용할 수 있는 명령의 의미를 간략히 요약한다.

- \dayoftheweek 요일을 보여준다. 인자는 YYYY/MM/DD 형식 또는 \today이다.

<code>\dayoftheweek{\today}</code>	목요일
------------------------------------	-----

- \xtoday 오늘 날짜와 요일을 보여준다.

<code>\xtoday</code>	2024년 5월 16일 (목요일)
----------------------	--------------------

- `\followingday` 인자로 주어지는 요일에 해당하는 오늘(2024년 5월 16일) 이후의 첫 날 날짜를 보여준다.

<code>\followingday{sun}</code>	2024년 5월 19일
---------------------------------	--------------

- `\dayafter`는 주어진 날짜로부터 #2일 지난 날이 며칠인지 보인다. 예를 들어 2020년 11월 20일부터 300일이 지난 날을 다음처럼 인쇄한다. 오늘을 기준으로 하려면 `\todayafter`.

<code>\dayafter{2020/11/20}{300}\</code> <code>\todayafter{300}</code>	2021년 9월 16일 (목요일) 2025년 3월 12일 (수요일)
---	--

- `\dayspassed` 명령은 주어진 두 날짜 사이의 경과일을 계산하여 디스플레이한다. 날짜에 `\today`를 쓸 수 있다. 경과일 자체만 얻으려면 별표를 붙인다.

<code>\dayspassed{2020/11/20}{\today}\</code> 선거일까지 <code>\dayspassed*{2024/1/5}</code> → <code>{2024/4/10}</code> 일.	2020년 11월 20일부터 2024년 5월 16일 까지 1273일 선거일까지 96일.
---	--

3 구현

3절의 내용은 `dayoftheweek.sty` 파일 안에 적혀 있는 주석문을 코드와 함께 조판한 것이다(소위 문학적 프로그래밍). 이를 조판하는 데 `srcandtext` 패키지를 썼다.

주어지는 날짜에 대하여 요일을 계산하는 목적의 패키지이다. 요일은 한글로 표현하므로 `ko.TeX`을 필요로 한다. 그밖에 경과일을 계산하거나 보여주는 등의 기능이 있다.

1. 패키지 정보

```
\ProvidesExplPackage
{dayoftheweek}
{2024/01/11}
{v0.3.3}
{what day of the week today}
```

3.1 함수

2. 준비.

이 패키지에서 내부적으로 사용할 변수를 선언하고, 각 달의 크기를 `prop` 자료형으로 미리 저장해둔다. 2월은 평년 기준 28일로 한다.

```
\tl_new:N \g_year_tl
\tl_new:N \g_month_tl
\tl_new:N \g_day_tl

\int_new:N \g_totaldays_int
```

```

\int_new:N \g_dayofweek_int

\prop_const_from_keyval:Nn \c_monthdays_prop {
  1 = 31,    2 = 28,    3 = 31,    4 = 30,
  5 = 31,    6 = 30,    7 = 31,    8 = 31,
  9 = 30,    10 = 31,   11 = 30,   12 = 31
}

\int_new:N \l_year_int
\int_new:N \l_month_int
\int_new:N \l_day_int
\int_new:N \l_daysofthismonth_int
\int_new:N \g_passeddays_int

```

3. 다음 함수는 YYYY/MM/DD 형식의 날짜 문자열을 분해하여 \g_year_tl, \g_month_tl, \g_day_tl에 각각 저장한다. 단순히 tl에 넣지 않고 int로 변환하는 단계를 거친 이유는 예컨대 2024/01/02가 2024/1/2와 동일하게 취급되게 하기 위해서이다.

```

\cs_new:Npn \dw_split_datestr_normal:w #1/#2/#3\q_stop
{
  \int_set:Nn \l_tmpa_int { #1 }
  \tl_gset:No \g_year_tl { \int_use:N \l_tmpa_int }

  \int_set:Nn \l_tmpa_int { #2 }
  \tl_gset:No \g_month_tl { \int_use:N \l_tmpa_int }

  \int_set:Nn \l_tmpa_int { #3 }
  \tl_gset:No \g_day_tl { \int_use:N \l_tmpa_int }
}

```

4. 총 지난 날짜수의 계산

핵심 프로시저이다. 주어진 날짜가 기산일(1900/01/01)로부터 며칠이나 지났는지를 계산한다.

입력되는 인자는 세 개로서 각각 연, 월, 일을 나타내는 정수들이다.

반환값이 없고 그 대신 \g_totaldays_int와 \g_dayofweek_int 두 개의 전역변수로 결과값을 되돌린다.

Routine 1 — 변수 \g_totaldays_int를 초기화한다.

Routine 2 — 전년도까지 1900년부터 매년의 날짜수를 \g_totaldays_int에 더해간다. 윤년이 들어 있으면 366을, 그렇지 않으면 365를 더한다. 이 루틴이 끝나면 직전 연도 12월 31일까지의 총 날짜수가 저장되어 있다.

Routine 3 — 주어진 달의 전월까지 매달 날짜수를 더한다. 각 달의 날짜수는 \dw_get_daymonth:nn을 불러서 계산한다. 이 루틴이 종료되면 직전 연도까지의 날짜수 + 직전 월까지의 날짜수가 저장된다.

Routine 4 — 주어진 날짜를 여기에 더한다. 여기까지 계산하면 \g_totaldays_int는 1900년 1월 1일 이후 며칠이 지났는지를 저장하고 있다.

Routine 5 — 1900년 1월 1일은 월요일이다. \g_totaldays_int에 저장된 숫자를 7로 나누어 얻은 나머지가 1이면 월요일, 0이면 일요일이 된다. 변수 \g_dayofweek_int에 요일을 나타내는 정수를 저장한다.

```

\cs_new:Npn \dw_calc_totaldays:nnn #1 #2 #3
{
  %% Routine 1 ---
  \int_zero:N \g_totaldays_int

  %% Routine 2 ---
  \int_step_inline:nnn { 1900 } { #1 - 1 }
  {
    \is_leapyear:nTF { ##1 }           %% 윤년인가?
    { %% 윤년이면 366일
      \int_gadd:Nn \g_totaldays_int { 366 }
    }
    { %% 평년이면 365일이다.
      \int_gadd:Nn \g_totaldays_int { 365 }
    }
  }

  %% Routine 3 ---
  \int_case:nnF { #2 }
  {
    %% #2가 1월이라면 0, 2월이라면 전달인 1월의 31일을 더한다.
    { 1 } { }
    { 2 } { \int_gadd:Nn \g_totaldays_int { 31 } }
  }
  {
    %% #2가 1월도 2월도 아닐 때, 전달까지 날짜수 합산
    \int_step_inline:nn { \int_eval:n { #2 - 1 } }
    {
      \exp_args:Noo \dw_get_daymonth:nn { #1 } { ##1 }
      \int_gadd:Nn \g_totaldays_int { \l_daysofthismonth_int }
    }
  }

  %% Routine 4 ---
  \int_gadd:Nn \g_totaldays_int { #3 }

  %% Routine 5 ---
  \int_gset:Nn \g_dayofweek_int { \int_mod:nn { \g_totaldays_int } {
    ↪ 7 } }
}

```

5. 윤년여부를 확인.

입력 인자는 연도를 나타내는 정수. 이 함수는 conditional이기 때문에 :nTF나 :nT 꼴로 쓸 수 있다. 정의상 \is_leapyear_p:n으로도 쓸 수 있다.

\ExplSyntaxOn

1900년은~윤년\is_leapyear:nTF{1900}{이}

↪ {이~아니}다.

\ExplSyntaxOff

1900년은 윤년이 아니다.

연도를 나타내는 정수가 4로 나누어지면서 100으로는 나누어지지 않을 때, 또는 400으로 나누어질 때 윤년임을 이용한다.

```
\prg_new_conditional:Npnn \is_leapyear:n #1 { p, T, TF }
{
  \bool_if:nTF {
    \int_compare_p:n { \int_mod:nn { #1 } { 4 } == 0 } &&
    ! \int_compare_p:n { \int_mod:nn { #1 } { 100 } == 0 } ||
    \int_compare_p:n { \int_mod:nn { #1 } { 400 } == 0 }
  }
  {
    \prg_return_true:
  }
  {
    \prg_return_false:
  }
}
```

6. 주어진 달의 날짜수 구하기

인자는 연과 월을 정수 형식으로 입력받는다. \c_monthdays_prop으로부터 월을 key로 하여 그 달의 날짜수를 구한다. 만약 #2가 2월이면 #1이 윤년이면 1 증가시켜 29일로 한다.

구해진 날짜수를 \l_daysofthismonth_int에 저장한다.

```
\cs_new:Npn \dw_get_daymonth:nn #1 #2
{
  \prop_get:NnN \c_monthdays_prop {#2} \l_tmpa_tl
  \int_gset:Nn \l_daysofthismonth_int { \l_tmpa_tl }

  \bool_if:nT
  { \int_compare_p:n { #2 == 2 } && \is_leapyear_p:n { #1 } }
  { \int_gincr:N \l_daysofthismonth_int }
}
```

7. 경과일을 주고 날짜를 구한다.

주어진 날짜에서 n 일 후가 몇년 몇월 며칠인지를 구한다. 문서 명령인 \dayafter 등에서 사용된다. 입력 인자는 네 개로서, 각각 연도, 월, 일과 경과일수이다. 즉 2024 01 02 40이면 2024년 1월 2일부터 40일후의 날짜를 구하여, \g_year_tl, \g_month_tl, \g_day_tl에 저장하여 돌려준다.

먼저, 입력된 세 번째 인자 날짜에 네 번째 경과일수를 더하여 2024 01 42의 꼴로 만든 다음, 이것을 연/월/일로 환산하는 보조 함수 \dw_cals_date_to_date:nnn을 호출한다.

```
\cs_new:Npn \dw_calc_date_to:nnnn #1 #2 #3 #4
{
  \dw_calc_date_to_date:nnn {#1} {#2} {#3+#4}
}
```

내부적으로 매크로를 인자로 줄 때가 있어서 variant를 하나 만들어둔다.

```
\cs_generate_variant:Nn \dw_calc_date_to:nnnn { nnnV }
```

\dw_calc_date_to:nnnn의 실제 계산은 \dw_calc_date_to_date:nnn에서 이루어진다.

앞서 예시한 2024년 1월 2일부터 60일후인 2024 1 2 60을 생각하자. 그러면 2024 1 62가 인자로 넘어올 것이다.

Routine 1 — 이 달이 며칠까지 있는지를 \dw_get_daymonth:nn로 확인한다.

Routine 2 — 이 달 날짜수보다 현재 주어진 날짜가 크다면 달을 넘어가는 것이다. \l_month_int를 1 증가시키고 이달 날짜수만큼을 뺀다. 만약 12월이면 연도를 1 증가시키고 월을 1로 만들어야 한다. 이 계산 후에, 달이 바뀌었으므로 ‘이번 달 날짜수’를 다시 계산하기 위하여 Routine 1을 한번 더 호출한다.

Routine 3 — 만약 아직도 이 달 날짜수보다 날짜가 더 크다면 자신을 재귀호출한다. 그렇지 않으면 결과를 \g_year_tl, \g_month_tl, \g_day_tl에 넣어 되돌리고 종료. 예: 31이 29보다 크므로, 이 함수 자신을 2024 2 31을 인자로 하여 다시 호출한다. 그러면 2월 -> 3월이 되고 31 -> 2가 된다. 이 결과를 돌려준다.

```
\cs_new:Npn \dw_calc_date_to_date:nnn #1 #2 #3
{
  \int_set:Nn \l_year_int { #1 }
  \int_set:Nn \l_month_int { #2 }
  \int_set:Nn \l_day_int { #3 }

  %% Routine 1 ---
  \exp_args:Noo \dw_get_daymonth:nn { \int_use:N \l_year_int } {
    ↪ \int_use:N \l_month_int }

  %% Routine 2 ---
  \int_compare:nT { \l_day_int > \l_daysofthismonth_int }
  {
    %% 12월이면 해도 넘어감
    \int_compare:nTF { \l_month_int == 12 }
    {
      \int_incr:N \l_year_int
      \int_set:Nn \l_month_int { 1 }
      \int_sub:Nn \l_day_int { \l_daysofthismonth_int }
    }
    {
      \int_incr:N \l_month_int
      \int_sub:Nn \l_day_int { \l_daysofthismonth_int }
    }
  }
}
```

```

    }

%% Routine 1 ---
%% 달이 넘어갔으므로 이달 날짜수를 다시 구한다.
\exp_args:Noo \dw_get_daymonth:nn { \int_use:N \l_year_int } {
  ↪ \int_use:N \l_month_int }

%% Routine 3 ---
\int_compare:nTF { \l_day_int > \l_daysofthismonth_int }
{
  \dw_calc_date_to_date:VVV \l_year_int \l_month_int \l_day_int
}
{
  \tl_gset:No \g_year_tl { \int_use:N \l_year_int }
  \tl_gset:No \g_month_tl { \int_use:N \l_month_int }
  \tl_gset:No \g_day_tl { \int_use:N \l_day_int }
}
}

```

```
\cs_generate_variant:Nn \dw_calc_date_to_date:nnn {VVV}
```

8. 요일 정수를 요일 이름으로 바꾸어 출력

\g_dayofweek_int는 요일을 나타내는 정수를 보관하고 있다. 이것을 출력하려면 ‘글자’로 바꾸어주어야 한다.

```

\cs_new:Npn \dw_print_dayoftheweek:n #1
{
  \int_case:nn { #1 }
  {
    {0} {일}
    {1} {월}
    {2} {화}
    {3} {수}
    {4} {목}
    {5} {금}
    {6} {토}
  }
}

```

```
\cs_generate_variant:Nn \dw_print_dayoftheweek:n { o, V }
```

9. 다음 함수는 문서 명령 \dayspassed에서 부르는 내부 루틴이다. 두 개의 인자를 받는데 인자는 YYYY/MM/DD 형식이고, 첫 인자의 날짜와 둘째 인자의 날짜 사이의 경과 일수를 계산하여 \g_passeddays_int 전역변수에 저장한다.

```

\cs_new:Npn \dw_calc_days_passed:nn #1 #2
{
  \str_if_eq:nnTF { \today } { #1 }
  {
    \dw_split_datestr_normal:w \year/\month/\day \q_stop
  }
  {
    \dw_split_datestr_normal:w #1 \q_stop
  }
  \tl_set:Nx \l_dateone_tl { \g_year_tl 년~\g_month_tl 월~\g_day_tl
    ↪ 일}
  \dw_calc_totaldays:nnn { \g_year_tl } { \g_month_tl } { \g_day_tl }
  \int_set_eq:NN \g_tmpa_int \g_totaldays_int

  \str_if_eq:nnTF { \today } { #2 }
  {
    \dw_split_datestr_normal:w \year/\month/\day \q_stop
  }
  {
    \dw_split_datestr_normal:w #2 \q_stop
  }
  \tl_set:Nx \l_datetwo_tl { \g_year_tl 년~\g_month_tl 월~\g_day_tl
    ↪ 일}
  \dw_calc_totaldays:nnn { \g_year_tl } { \g_month_tl } { \g_day_tl }
  ↪ }
  \int_set_eq:NN \g_tmpb_int \g_totaldays_int

  \int_gset:Nn \g_passeddays_int { \int_abs:n { \g_tmpa_int -
    ↪ \g_tmpb_int } }
}

```

3.2 문서 명령

10. 이 패키지의 핵심 명령은 `\dayoftheweek`이다. 인자는 YYYY/MM/DD 형식이다. 다음 예를 보라.

```
\dayoftheweek{2024/05/20}
```

월요일

인자가 `\today`일 때는 시스템 날짜로 바뀌어서 처리한다.

```

\NewDocumentCommand \dayoftheweek {m}
{
  \str_if_eq:nnTF { #1 } { \today }
  {
    \dw_split_datestr_normal:w \year/\month/\day \q_stop
  }
}

```



```

    }
    {
        \dw_split_datestr_normal:w #1 \q_stop
    }

    \dw_calc_totaldays:nnn {\g_year_tl} {\g_month_tl} {\g_day_tl}
    \dw_print_dayoftheweek:V \g_dayofweek_int
    요일
}

```

11. \xtoday는 오늘 날짜에 요일을 괄호 안에 넣어 출력한다. 옵션 인자로 [p] 또는 [t]를 줄 수 있는데, [p]가 디폴트이다. p가 주어지면 요일에 괄호를 치고 t가 주어지면 괄호 없이 요일만을 식자한다.

```

\NewDocumentCommand \xtoday {O{p}}
{
    \str_case:nnF { #1 }
    {
        { p } { \bool_set_true:N \l_tmpa_bool }
        { t } { \bool_set_false:N \l_tmpa_bool }
    }
    {
        \bool_set_false:N \l_tmpa_bool
    }
    \dw_calc_totaldays:nnn {\year} {\month} {\day}
    \the\year 년~\the\month 월~\the\day 일~
    \bool_if:NT \l_tmpa_bool { ( }
        \dw_print_dayoftheweek:V \g_dayofweek_int 요일
    \bool_if:NT \l_tmpa_bool { ) }
}

```

12. \followingday는 인자로 sun, mon, tue, wed, thu, fri, sat 가운데 하나를 취한다. 그러면, 오늘을 기준으로 처음 오는 해당 요일의 날짜를 출력한다.

`\followingday{sun}`

2024년 5월 19일

위의 예는 오늘(2024년 1월 3일) 이후 첫 일요일 날짜를 보여준다. 달이나 해가(12월 마지막 주일 때) 넘어가는 것을 체크해야 하기 때문에 \dw_calc_date_to:nnnV를 호출한다.

```

\NewDocumentCommand \followingday {m}
{
    \str_case:nn { #1 }
    {
        {sun} {\int_set:Nn \l_tmpa_int {0} }
        {mon} {\int_set:Nn \l_tmpa_int {1} }
    }
}

```

```

    {tue} {\int_set:Nn \l_tmpa_int {2} }
    {wed} {\int_set:Nn \l_tmpa_int {3} }
    {thu} {\int_set:Nn \l_tmpa_int {4} }
    {fri} {\int_set:Nn \l_tmpa_int {5} }
    {sat} {\int_set:Nn \l_tmpa_int {6} }
  }

  \dw_calc_totaldays:nnn {\year} {\month} {\day}
  \int_compare:nTF { \g_dayofweek_int < \l_tmpa_int }
  {
    \int_set:Nn \l_tmpb_int { \l_tmpa_int - \g_dayofweek_int }
  }
  {
    \int_set:Nn \l_tmpb_int { \l_tmpa_int + 7 - \g_dayofweek_int }
  }

  \dw_calc_date_to:nnnV {\year} {\month} {\day} \l_tmpb_int

  \g_year_tl 년~\g_month_tl 월~\g_day_tl 일
}

```

13. \dayafter는 두 개의 인자를 취하며, 첫 인자는 YYYY/MM/DD 형식이고 두번째 인자는 정수이다. 주어진 날로부터 #2만큼 지난 날이 언제인지를 출력한다.

\todayafter는 \dayafter의 첫 인자를 시스템 날짜(오늘)로 하는 명령이다. 따라서 하나의 인자만을 취한다.

\dayafter{2024/1/3}{100} \\	2024년 4월 12일 (금요일)
\todayafter{100} \\	2024년 8월 24일 (토요일)
\todayafter{1}	2024년 5월 17일 (금요일)

내일(다음 날)은 1일 후이다. 그러므로 n 일째 되는 날을 알고자 한다면 $n - 1$ 을 인자로 주어야 한다.

```

\NewDocumentCommand \todayafter {m}
{
  \dw_calc_date_to:nnnn { \year } { \month } { \day } { #1 }
  \g_year_tl 년~\g_month_tl 월~\g_day_tl 일~(
  \dw_calc_totaldays:nnn {\g_year_tl} {\g_month_tl} {\g_day_tl}
  \dw_print_dayoftheweek:V \g_dayofweek_int 요일)
}

```

```

\NewDocumentCommand \dayafter {mm}
{
  \dw_split_datestr_normal:w #1 \q_stop
}

```

```

\dw_calc_date_to:nnnn {\g_year_t1} {\g_month_t1} {\g_day_t1} {#2}
\g_year_t1 년~\g_month_t1 월~\g_day_t1 일~(
\dw_calc_totaldays:nnn {\g_year_t1} {\g_month_t1} {\g_day_t1}
\dw_print_dayoftheweek:V \g_dayofweek_int 요일)
}

```

14. \dayspassed는 두 개의 인자를 취한다. 각 인자는 \today 또는 YYYY/MM/DD 형식이고, 두 날짜 사이의 경과일을 출력한다. 별표붙은 명령 형식으로 쓰면 단순히 경과일수만 출력하고 그렇지 않으면 다음 예제와 같은 모양이 된다.

```

\dayspassed{\today}{1995/3/5}\
\dayspassed*{1995/3/5}{2024/1/3}

```

1995년 3월 5일부터 2024년 5월 16일까지

10665 일

10531

인자의 순서는 뒤바뀌어도 결과가 같다.

```

\NewDocumentCommand \dayspassed { s m m }
{
  \dw_calc_days_passed:nn { #2 } { #3 }

  \IfBooleanTF { #1 }
  {
    \int_use:N \g_passeddays_int
  }
  {
    \int_compare:nTF { \g_tmpa_int > \g_tmpb_int }
    {
      \l_datetwo_t1 부터~\l_dateone_t1 까지~
    }
    {
      \l_dateone_t1 부터~\l_datetwo_t1 까지~
    }
    \fbox{ \int_use:N \g_passeddays_int } 일
  }
}

```

3.3 Change log

- 2019/2/26: released 0.1
- 2019/2/27: v0.2 : \g_passeddays_int introduced
- 2024/1/2 : v0.3 : \dw_calc_date_to_date:nnn bugfix. srcandtext.
- 2024/1/5 : v0.3.2: \dw_get_daymonth:nn introduced.
- 2024/1/11: v0.3.3: \is_leapyear:nTF