

# kocircnum, 원숫자

Nova De Hi, Dohyun Kim, Yi Hoze

version 1.2 [2014/06/16]

## 요약

①, ②와 같이 동그라미 친 숫자를 식자하게 한다. 이 패키지는 Dohyun Kim의 hcrnumbers와 Yi Hoze의 hzmisc에 있는 명령 `\wrapnum`, 그리고 새로 정의된 `\tikzcircnum` 세 개를 합친 것이다. 간편하게 사용할 수 있는 `\circ`와 카운터 수식자 `\Cnum` 명령을 제공한다.

## 차례

1 소개	2
2 사용법	3
2.1 요구사항	3
2.2 패키지 사용 선언	3
2.3 <code>\hrcircnum</code>	3
2.4 <code>\hzcircnum</code>	6
2.5 <code>\tikzcircnum</code>	9
2.6 기본 명령	15
3 카운터와 함께 사용하기	16
4 <code>enumerate</code> 에서 사용하기	17
4.1 <code>\Cnum</code> 카운터 수식자를 이용하기	17
4.2 다른 방법	19
4.3 아이টে를 참조하기	20
5 섹션 타이틀에 쓰는 방법	20
6 기타	21
7 변경 사항	21
8 구현	21

9 wrapnum 패키지	34
9.1 사용법	34
9.2 기타	35
10 hcrnumbers 패키지	36
찾아보기	37

## 1 소개

동그라미 친 숫자를 식자하는 방법에 대해서 [KTUG의 Faq 페이지 “원글자”](#)<sup>1</sup>에는 다음과 같은 방법이 소개되어 있다.

- ① pifont, dingbat 이용법
- ② Combinumerals font를 이용하는 방법
- ③ 함초롬체/GSUB의 nalt 속성 이용법
- ④ tikz를 이용하여 그리기
- ⑤ 그밖의 방법

이 가운데 pifont 방법은 사용할 수 있는 숫자가 너무 적고  $ko_{\text{TeX}}$ 의 `\onum` 카운터 수식자 역시 15까지만 가능하다. “임의의 숫자”를 원숫자로 식자하는 방법이 필요해져서 위의 페이지에 제시된 여러 가지 방법이 나왔다. 이 가운데 ③번 방법과 ④번 방법을 하나의 패키지로 합친 것이 `kocircnum`이다.

이 패키지에 포함된 코드 가운데 `\hrcircnum`은 Dohyun Kim이 작성한 `hcrnumbers.sty`<sup>2</sup>를, `\wrapnum`은 이호재의 `hzmisc.sty`에 포함된 것<sup>3</sup>을 가져온 것이다. 나머지 `\tikzcircnum`의 정의와 패키지의 구성은 Nova De Hi가 한 것이다. 한편, `hzmisc`의 `\wrapnum`에 대하여, 이 패키지와는 별도로 `wrapnum`을 만들어두었는데,<sup>4</sup> `hzmisc`에 기원을 두고 있다는 점은 같으나 이 패키지가 옵션이나 명령 이름, 동작 방식을 대폭 수정한 반면, `wrapnum` 패키지의 `\wrapnum` 명령은 `hzmisc`의 것과 거의 동일하게 되어 있다.

이 패키지의 v1.2에서 `hcrnumbers`의 코드를 대폭 재구현하면서 원본 `hcrnumbers.sty`를 별도의 스타일로 포함해 두었다. 이 패키지만이 필요하다면 `hcrnumbers`를 사용하지 바란다.

<sup>1</sup><http://faq.ktug.org/faq/%BF%F8%B1%DB%CO%DA>

<sup>2</sup>Dohyun Kim, “함초롬의 nalt 속성을 이용해보자,” KTUG Team Blog, 2011/06/29, [http://www.ktug.org/xe/index.php?document\\_srl=44737](http://www.ktug.org/xe/index.php?document_srl=44737).

<sup>3</sup>yihoe, “원숫자,” KTUG Team Blog, 2014/01/02, [http://www.ktug.org/xe/index.php?document\\_srl=17885](http://www.ktug.org/xe/index.php?document_srl=17885). `hzguide` 패키지의 일부이다. <http://wiki.ktug.org/wiki/wiki.php/hzguide>.

<sup>4</sup>[http://www.ktug.org/xe/index.php?document\\_srl=183347](http://www.ktug.org/xe/index.php?document_srl=183347)

## 2 사용법

### 2.1 요구사항

- Xe<sub>La</sub>TeX, Lua<sub>La</sub>TeX 사용시 (패키지 디폴트인) `\hrcircnum`을 쓰려면 함초롬체/GSUB 글꼴<sup>5</sup>이 설치되어 있어야 한다.
- `\hrcircnum`을 위해 Xe<sub>La</sub>TeX, Lua<sub>La</sub>TeX에서 `fontspec` 패키지를 필요로 한다.
- 색상을 제어하기 위해서 `color` 또는 `xcolor` 패키지를 로드하여야 한다.

### 2.2 패키지 사용 선언

문서의 preamble에 다음과 같이 선언한다.

```
\usepackage[<option>]{kocircnum}
```

옵션으로 줄 수 있는 것은 `[hcr]`, `[hz]`, `[tikz]`, `[tikzsmall]`, `[tikzbig]`이다.

앞의 세 옵션은 `\circ`의 동작을 선택하는 것이다. 이 패키지는 `\circ`이라는 명령을 정의하는데, 이 명령의 의미가 옵션에 따라 달라진다. 옵션을 주지 않으면 `[hcr]`과 같다. 어떤 옵션을 주더라도 이 패키지가 정의한 모든 명령을 사용할 수 있으며 다만 `\circ...`의 의미만이 달라진다. 자세한 사항은 2.6 소절을 보라.

`[tikzbig]`과 `[tikzsmall]`은 기본적으로 만들어지는 원의 크기에 차이가 있고 다른 것은 모두 같다. 단순히 `[tikz]`라고만 하면 `[tikzsmall]`이 된다. 이것은 v1.1의 변경사항이다. 2.5 소절에서 설명한다.

### 2.3 \hrcircnum

함초롬체/GSUB의 `nalt` 속성을 이용하여 원숫자를 식자한다. 구체적인 방법은

```
\fontspec[Annotation=4]{HCR Batang LVT} 1
```

와 같이 `\fontspec` 명령에 `Annotation`을 주어 숫자를 찍는 것이다. 두 자리 또는 세 자리 숫자가 왔을 때 이를 적절하게 처리하기 위한 코드를 포함하는 Dohyun Kim이 작성한 `hcrnumbers`를 Nova De Hi가 재구현하였다. 원본에서는 `\circ` 22와 같이 숫자를 주고 마지막에 스페이스로 숫자의 끝을 표시하여 원숫자를 식자하게 되어 있는데 그 결과 숫자 뒤에 오는 스페이스가 사라지는 현상이 있었다. 새로 구현하면서 좀더  $\TeX$ 스럽게 `\hrcircnum{22}`와 같은 방식으로 숫자를 인자로 주도록 하였다. 그러나 근본적으로 다른 것은 아니며 `hcrnumbers`와 거의 동일하게 동작한다.

<sup>5</sup><http://faq.ktug.org/faq/%c7%d4%c3%ca%b7%d2%c3%bc/GSUB>

한편, 이 명령은 pdfTeX에서는 의미가 없다. 트루타입/오픈타입 속성을 이 엔진에서 이용할 수 없기 때문이다. pdfTeX으로 \hrcircnum 명령이 포함된 문서를 처리하면 오류가 발생하지는 않으나 실제 함초롬체를 사용하는 효과도 일어나지 않고 단지 이후 설명할 \hzcircnum이 대신 동작한다.

사용자 인터페이스 명령은 다음과 같다.

```
\hrcircnum[shape = <circle/box>, white/black,color=<color>,%
           font=<font>,reset]{number}
\hrcircnum*[ ... ]{number}
```

\hrcircnum의 옵션은 다음과 같다.

**shape** 동그라미 모양과 네모난 모양 가운데 고를 수 있다. shape=circle은 간단히 circle이라고만 해도 된다. box 대신 rectangle을 쓸 수 있다.

**white, black** 흰 바탕에 검은 숫자, 검은 바탕에 흰 숫자.

**color** color 관련 패키지를 로드해야 한다. white 상태이면 전경색이, black 상태이면 배경색이 지정한 색으로 칠해진다.

**font** 예를 들어 \rmfamily\bfseries 등을 지정할 수 있다. \rmfamily이면 함초롬 바탕체가, \sffamily이면 함초롬 돋움체가 사용된다. 사이즈 명령 등을 줄 수도 있다.

**reset** 설정을 초기화한다. 옵션 중에서 제일 먼저 쓰는 것이 좋다.

숫자는 세 자리까지 가능하고 이를 초과하면 식자되지 않는다.

**기본 사용법** 아무런 옵션 없이 쓰면 다음과 같다.

```
\hrcircnum{0}, \hrcircnum{1}, \hrcircnum{25}, \hrcircnum{116}
```

①, ①, ㉕, ㉙

옵션으로 shape나 black/white를 일단 지정하고 나면 그 후로도 같은 모양이 지속된다. 다음 예를 보자. shape를 rectangle로 지정하면 다음과 같다. 간단히 box라고 지정할 수 있다.

```
\hrcircnum[shape=rectangle]{1},
\hrcircnum[box]{25},
\hrcircnum{116}
```

1, 25, 116

다음 보기는 네 가지 모양을 식자해본 것이다.

```

\hrcircnum[shape=circle,white]{20}, %% default
\hrcircnum[shape=circle,black]{20},
\hrcircnum[shape=rectangle,white]{20},
\hrcircnum[shape=rectangle,black]{20}

```

②①, ②①, ②①, ②①

font 설정을 바꾸면 다음과 같이 된다.

```

\hrcircnum[reset,font={\Large}]{20}
\hrcircnum[reset,black,font={\sffamily\bfseries\LARGE}]{30}

```

②① ③①

color 설정을 시험해보자. black을 함께 지정하면 배경색이 지정된 색상으로 칠해진다.

```

\hrcircnum[reset,color=blue]{15}
\hrcircnum[color=red]{15}
\hrcircnum[color=black]{15}
\hrcircnum[black,color=blue]{15}
\hrcircnum[color=red]{15}

```

①⑤ ①⑤ ①⑤ ①⑤ ①⑤

**별표붙은 명령, reset** 별표붙은 명령 `\hrcircnum*`을 써서 모든 옵션을 처음 상태로 되돌린다. 별표붙은 명령에 옵션 인자를 함께 사용하면 초기화한 후에 주어진 옵션만 효과를 발휘한다. 별표붙은 명령을 쓰는 것과 reset 옵션을 주는 것은 동일하다. 단, reset 옵션은 반드시 모든 옵션 중에 제일 처음에 있어야 한다.

```

\hrcircnum*{1},
\hrcircnum[shape=rectangle]{2},
\hrcircnum*[black]{3},
\hrcircnum{4},
\hrcircnum[reset,box]{5},
\hrcircnum[black]{6}

```

①, ②, ③, ④, ⑤, ⑥

네 종류의 모양을 식자하는 앞서 보인 코드는 이전 속성을 그 후로 이어받는다는 점과 약어 옵션을 이용하여, 다음과 같이 간단히 코딩할 수 있다.

```

\hrcircnum*{20},
\hrcircnum[black]{20},
\hrcircnum*[rectangle]{20},
\hrcircnum[black]{20}

```

⑳, ㉑, ㉒, ㉓

**선언형 옵션 명령과 리셋 명령** color와 shape의 기본값을 다음 명령으로 바꿀 수 있다. 바꾼 이후에는 지속적으로 영향을 미친다.

```
\hrcircnumsetup{shape=<shape>,color=<color>}
\restorehrcircnumsetup
```

\restorehrcircnumsetup는 원래의 기본값으로 되돌리는 명령이다. 별표붙은 명령이나 reset 옵션을 통해서 간단히 할 수 있는 일이지만 옵션 인자를 쓸 수 없는 경우 등을 위하여 별도로 마련해두었다.

\hrcircnumsetup의 사용례를 보이면 다음과 같다. \hrcircnum의 옵션과 마찬가지로 약어도 동작한다.

```
\hrcircnumsetup{shape=rectangle,black}\hrcircnum{20},
\hrcircnumsetup{shape=rectangle,white}\hrcircnum{50},
\hrcircnumsetup{circle,black}\hrcircnum{60},
\hrcircnumsetup{circle,white}\hrcircnum{70}
```

㉔, ㉕, ㉖, ㉗

## 2.4 \hzcircnum

이호재의 hzguide 패키지 묶음에 들어 있는 hzmisc.sty에는 \wrapnum이라는 명령이 정의되어 있다. 이 코드를 가져와서 Nova De Hi가 수정하였다. 코드 사용에 대한 허락은 추후에 받을 계획이다.

```
\hzcircnum[shape=<type>,
           color=<color>,
           bgcolor=<background color>,
           fgcolor=<foreground color>,
           sep=<dim>,raise=<dim>,Raise=<dim>,
           font=<font>,fontplus=<font>,
           base=<text>,reset]{number}
\hzcircnum*[ ... ]{number}
```

\hzcircnum에 주어지는 옵션 <key>는 다음과 같은 의미이다.

**shape** 숫자를 둘러쌀 모양. circle, rectangle, oval, ball 중에서 선택한다.

**color, bgcolor, fgcolor** 색상을 지정한다. 색상은 xcolor의 기정의 색상이나 자신이 `\definecolor`한 것을 모두 쓸 수 있다. **color**는 **bgcolor**와 **fgcolor**를 일괄지정하는 옵션으로서 `color=black`으로 하면 검은 바탕에 흰 숫자, `color=white`로 하면 흰 바탕에 검은색 숫자가 찍힌다. 이 이외의 색상을 주면 주어진 색상을 **bgcolor**로 하고 숫자는 `white`가 된다.

**sep** 길이값을 준다. 둘러싸는 도형과 숫자 사이의 간격. 이 값이 커지면 둘러싸는 도형 자체가 커진다.

**raise, Raise** 원숫자의 수직 이동값을 길이값으로 지정한다. 기본값은 `-0.5ex`이다. **raise** 옵션은 이미 주어진 기본값을 기준으로 이동할 길이를 정한다. 반면 기본값을 무시하고 절대 길이값을 주려면 **Raise**로 지정한다.

**font, fontplus** 숫자를 식자할 폰트. 기본값은 `{\sffamily\footnotesize}`이다. **fontplus**는 현재의 폰트 설정을 그대로 두고 인자로 주어진 값을 그 후에 추가한다. 폰트 패밀리는 그대로 두고 사이즈만 바꾼다든가 할 때 유용하다.

**base** 주어진 텍스트의 너비를 기준으로 원숫자를 그린다.

**reset** 모든 옵션 설정을 되돌린다.

편의를 위해 약어 옵션이 마련되어 있다. 각각 `circle`, `rectangle`, `oval`, `ball`만을 써도 `shape=circle` 등으로 한 것과 동일하며, `rectangle`은 `rect`로, `circle`은 `circ`로 줄여 쓸 수도 있다. 색상 관련 약어는 `white`와 `black`이 있고 의미는 `color=white` 등으로 한 것과 동일하다.

별표붙은 명령의 사용법은 앞서 `\hzcircnum`의 경우와 같다. 기본값으로 되돌리는 것이다. 별표붙은 명령을 쓰는 것과 옵션 처음에 `reset`을 붙이는 것은 동일하다.

**사용례** 이제 사용례를 보자. 아무런 옵션을 주지 않으면 다음처럼 된다.

```
\hzcircnum{5}, \hzcircnum{67}
```

⑤, ⑥⑦

**shape** `shape`는 약어로 옵션을 줄 수 있다.

```
\hzcircnum[shape=circle]{5}, \hzcircnum[shape=ball]{5},  
\hzcircnum[shape=rectangle]{5},  
\hzcircnum[shape=oval]{5},  
\hzcircnum[circ]{7},  
\hzcircnum[rect]{7}
```

⑤, ⑤, ⑤, ⑤, ⑦, ⑦

**sep** rectangle이나 oval은 sep 값을 조금 주는 편이 좋다.

```
\hzcircnum[shape=rectangle,sep=1.5pt]{5},  
\hzcircnum[shape=oval,sep=1.5pt]{5}
```

5, 5

**color** color를 바꾸어 본다. 아래의 보기들은 일단 초기화한 후에 해당 속성만 바꾼 것이다. 초기화 명령을 별도로 보이지 않았다. white와 black은 약어를 쓸 수 있다.

```
\hzcircnum[black]{5}, \hzcircnum[color=blue]{5},  
\hzcircnum[color=red]{5}, \hzcircnum[color=orange]{5},  
\hzcircnum[white]{5}
```

5, 5, 5, 5, 5

**fgcolor**와 **bgcolor**를 별도로 설정할 수 있다.

```
\hzcircnum[fgcolor=red,bgcolor=blue]{5}
```

5

**font** font를 설정할 수 있다.

```
\hzcircnum[color=blue,font={\rmfamily\itshape}]{5},  
\hzcircnum{6}
```

5, 6

**fontplus**를 써서 현재의 폰트 속성에 **\bfseries**를 추가해본다.

```
\hzcircnum[fontplus={\bfseries}]{5},  
\hzcircnum{6}
```

5, 6

**raise**와 **sep** raise와 sep 길이값을 테스트한다. 초기화를 위해 별표붙은 명령을 썼다.

```
a, b, c, 1, 2, 3, \hzcircnum*[raise=3pt,sep=3pt]{7},  
\hzcircnum[Raise=3pt]{8}
```



a, b, c, 1, 2, 3, 7, 8

여기서 `raise`와 `Raise`가 다른 이유는 기본적으로 `\hzcircnum`이 약간 내려찍는 것을 디폴트로 하고 있기 때문이다. `raise`는 이렇게 디폴트로 식자된 위치에서 주어진 값만큼 올려찍는 데 비해 `Raise`는 기본값을 무시하고 주어진 길이만큼 박스를 올려찍는다. 일반적인 목적에는 `raise`만으로 충분하다. 다음 예를 보면 의미가 분명하리라 본다.

```
1, 2, 3, \hzcircnum*{4}, \hzcircnum[raise=0pt]{5},  
\hzcircnum[Raise=0pt]{6}
```

1, 2, 3, 4, 5, 6

즉, `raise=0pt`로 하였을 때 기본적으로 식자되는 것과 위치가 동일하다는 것이다. 반면 `Raise=0pt`는 `0pt`라는 숫자에도 불구하고 위치가 꽤 올라가게 된다. 사용자 입장에서 `Raise=0pt`는 혼동스러울 수 있다.

**base** base값과 font를 바꾸어서 세 자리 숫자를 식자해보자. 다음 예에서 ‘333’을 지정한 것은 이 세 자리 숫자가 차지하는 간격만큼을 기준으로 원숫자를 그린다는 의미이다.(base 옵션을 font보다 나중에 주는 것이 좋다.)

```
\hzcircnum[font={\sffamily\tiny},base=333]{989}
```

989

## 선언형 설정 명령과 리셋 명령

```
\hzcircnumsetup{ <keyval list> }  
\restorehzcircnumsetup
```

`\hzcircnumsetup`은 옵션으로 줄 인자를 미리 선언하는 형식의 명령이다. 맨처음의 기본값으로 바꾸려면 `\restorehzcircnumsetup`을 선언한다.

## 2.5 \tikzcircnum

위의 `\hzcircnum`와 마찬가지로 `tikz`를 이용하여 원숫자를 그리도록 한 것이다. 많은 부분이 `\hzcircnum`과 유사하다. 원래 이것은 Nova De Hi가 `\hzcircnum`과 별도로 만들었던 명령에 기원을 두고 있는데 하나의 패키지로 합치면서 옵션과 동작을 일치시키려다보니 거의 유사한 명령이 만들어졌다. 그러나 `\hzcircnum`과 다른 방식으로 동작하는 부분이 몇 가지 있고 기본적으로 만들어지는 원숫자의 모양도 조금 다르다.

```

\tikzcircnum[shape=<shape>,
             color=<color>,bgcolor=<background color>,
             fgcolor=<foreground color>,
             bdcolor=<border line color>,colorinv=<color>,
             round=<dim>,raise=<dim>,Raise=<dim>,
             sep=<dim>,sepplus=<dim>,
             font=<font>,fontplus=<font>,base=<text>,
             linewidth=<dim>,
             reset]{number}
\tikzcircnum*[ ... ]{number}

```

\tikzcircnum에 주어지는 옵션의 의미는 다음과 같다.

- shape에서 선택할 수 있는 것은 circle, rectangle, ball이다. 이 세 가지 shape에 대해서는 약식 옵션이 마련되어 있어서 shape=circle 대신 circle 이라고만 해도 좋다.
- color 옵션의 의미가 black, white가 아닌 일반 색상일 경우 \hzcircnum과 다른 점은 배경색이 아니라 전경색을 칠하게 한다는 것이다. 만약 \hzcircnum의 color와 같은 의미가 되게 하려면 colorinv로 지정해야 한다. bdcolor(또는 linecolor)는 외곽선을 그리는 색상을 지정한다.<sup>6</sup>
- 이 명령에만 있는 round는 rectangle을 shape로 선택했을 때만 의미가 있는 것으로 tikz의 rounded corners 값이다. 이 key가 없으면 아무런 역할도 하지 않지만 단순히 round라고만 선언해도 기본값인 2pt를 넘겨준다. 물론 길이값을 지정할 수 있다. round를 선언한 rectangle은 \hzcircnum의 oval과 거의 같다.
- 이 명령에만 있는 linewidth는 외곽선의 두께를 지정한다. 기본값은 0.4pt로서 tikz의 thin에 해당한다.
- 그밖의 것은 \hzcircnum의 옵션과 비슷하여 더 설명할 것이 없다.

**사용례** 아무 옵션도 주지 않으면 다음과 같다 ([tikzbig]).

```
\tikzcircnum{5}, \tikzcircnum{25}
```

⑤, ②⑤

<sup>6</sup>외곽선 색상은 Xe<sub>La</sub>TeX에서만 정상 동작한다.

**tikzsmall** 패키지 옵션으로 [tikzsmall]을 주면 \tikzcircnum(\circnum)으로 찍히는 원숫자의 모양이 달라진다. 다음은 그 모양을 보인 것이다.

```
\tikzcircnum{5}, \tikzcircnum{25}, \tikzcircnum{99}
```

⑤, ②⑤, ⑨⑨

**shape** shape를 rectangle로. 이 속성은 이후로 이어받는다. 약어 옵션을 쓸 수 있다. round를 지정하되 값을 정해주지 않으면 기본값인 2pt가 쓰인다.

```
\tikzcircnum[shape=rectangle]{5},
\tikzcircnum{25},
\tikzcircnum[round]{10},
\tikzcircnum[round=5pt]{15},
\tikzcircnum[ball]{8}
```

5, 25, 10, 15, 8

**sep** sep을 조금 주어보자. 이미 round 값을 주었으므로 shape만 rectangle로 하고 sep값을 변화시킨다.

```
\tikzcircnum[rectangle,sep=5pt]{5},
\tikzcircnum[sep=5pt]{25}
```

5, 25

초기화하고 sep과 sepplus를 비교해본다. sepplus는 현재 설정되어 있는 sep 값에 주어진 길이값을 계속 더해가는 것이다.

```
\tikzcircnum[reset,sep=0pt]{5},
\tikzcircnum[sepplus=1pt]{6},
\tikzcircnum[sepplus=2pt]{7},
\tikzcircnum[sep=1pt]{8}
```

⑤, ⑥, ⑦, ⑧

\tikzcircnum은 기본적으로 sep 값에 약간의 여유값을 준다. sep 길이를 0pt로 하여도 이 여유값은 작용하므로 실제 0pt를 sep으로 하려면 이 값(0.25pt)만큼을 줄여주어야 한다. 이것은 \hzcircnum과 다른 점이다.

```
\tikzcircnum[reset,sep=0pt]{9},
\tikzcircnum[sep=-0.25pt]{9}
```

⑨, ⑨

**color** 초기화하고 color를 바꾼다. black과 white는 약어 옵션으로 쓸 수 있다.

```
\tikzcircnum*[black]{9}
\tikzcircnum[white]{10}
```

9 10

```
\tikzcircnum*[color=blue]{5}
\tikzcircnum[color=green]{25}
```

5 25

colorinv는 black, white일 때는 color와 같다. 그러나 일반 색상이면 \hzcircnum에서의 color 효과처럼 배경에 지정된 색상을 칠하고 전경을 흰색으로 한다.

```
\tikzcircnum*[colorinv=blue]{5}
\tikzcircnum[colorinv=green]{25}
\tikzcircnum[colorinv=white]{6}
\tikzcircnum[colorinv=black]{16}
```

5 25 6 16

초기화하고 배경색과 전경색을 각각 바꾸어본다.

```
\tikzcircnum*[bgcolor=black,fgcolor=green]{5}
\tikzcircnum[bgcolor=orange,fgcolor=white]{25}
```

5 25

초기화하고 외곽선 색상을 바꾼다. 외곽선의 디폴트는 gray이다.(단, [tikzsmall]이면 black).

```
\tikzcircnum*[bdcolor=blue]{5},
\tikzcircnum[bdcolor=red]{25},
\tikzcircnum[reset]{6}
```

5, 25, 6

**font** 초기화하고 font를 바꾼다.

```
\tikzcircnum*[font={\rmfamily\bfseries}]{5},
\tikzcircnum{6}
```

5, 6

fontplus로 \itshape를 추가해보자.

```
\tikzcircnum[fontplus={\itshape}]{5}
```

5

**raise, Raise** raise를 바꾼다. raise와 Raise의 차이는 \hzcircnum의 경우와 같다. 초기화하지 않으므로 직전에 테스트한 폰트 속성을 이어받는다.

```
1,2,3,4,\tikzcircnum[raise=0pt]{5},  
\tikzcircnum[Raise=0pt]{6},7
```

1,2,3,4,⑤,⑥,7

**base** \hzcircnum과 같이 base로 크기를 재설정할 수 있다. 폰트를 바꾸면 크기 설정에 영향을 미친다. 따라서 font를 base보다 앞서 선언하는 것이 좋다.

```
\tikzcircnum[reset,base=999]{16},  
\tikzcircnum[reset,font={\sffamily\large},base=000]{16},  
\tikzcircnum[reset,base=999]{999}
```

①⑥, ①⑥, ①⑥⑨⑨⑨

**linewidth** 외곽선의 두께를 조절한다.

```
\tikzcircnum[reset,linewidth=1pt]{5},  
\tikzcircnum[reset,linewidth=2pt]{6},  
\tikzcircnum[reset,bdcolor=blue,linewidth=1.5pt]{7}
```

⑤, ⑥, ⑦

참고로, linewidth 대신 bdwidth라고 할 수 있으며, 마찬가지로 bdcolor 대신 linecolor라고 해도 된다.

### 선언형 옵션 명령과 리셋 명령

```
\tikzcircnumsetup{ <keyval list> }  
\restoretikzcircnumsetup
```

\tikzcircnumsetup은 옵션값을 설정하는 명령이다. \restoretikzcircnumsetup을 선언하면 모든 설정값을 원래의 기본값으로 초기화한다. [tikz], [tikzsmall], [tikzbig] 옵션을 선언한 상태라면 \circnumsetup과 \restorecircnumsetup 명령을 쓸 수 있다. \restoretikzcircnumsetup 명령은 reset하거나 별표 붙인 \tikzcircnum을 쓴 것과 효과가 동일하다.

## 한 번만 적용되는 설정

```
\tikzcircnumonce[ <keyval list> ]{number}
```

다른 명령군에는 없고 `\tikz...`에만 있는 명령이다. 이것은 주어진 옵션의 설정으로 단 한 번만 식자하게 한다. 그 뒤로는 설정이 이어지지 않고 이 명령이 주어지기 직전 설정을 따른다.

```
\tikzcircnum[reset]{1},  
\tikzcircnum[color=black]{2},  
\tikzcircnumonce[rect,colorinv=red,round]{3},  
\tikzcircnum{4}
```

①, ②, ③, ④

위의 예에서 3을 식자한 뒤의 4는 그 이전 2를 식자할 때의 설정을 그대로 보존하고 있다. 이 명령은 예를 들면 장절 표제의 번호를 원숫자로 식자하고자 할 때 필요하다. 왜냐하면 절표제를 식자하면서 사용했던 설정이 본문에 계속 영향을 미쳐서는 곤란하기 때문이다. 5절에서 그 사용례를 볼 수 있다.

[tikz]나 [tikzsmall] 옵션을 준 경우에는 `\circnumonce` 명령을 쓸 수 있고 `\tikzcircnumonce`와 같다.

## 디폴트 옵션의 사용자화

```
\settikzcircnumsetup{ <keyval list> }
```

다른 명령군에는 없고 `\tikz...`에만 있는 명령이다. `\settikzcircnumsetup`은 문서 전체에 대해서 `\tikzcircnum`의 디폴트 값을 일괄 변경하고자 할 때 사용한다. 즉 자신이 설정한 어떤 모양을 `\tikzcircnum`의 기본 모양으로 선언하는 것이다. 이 명령이 주어진 다음에는 별표붙은 명령이나 `\restoretikzcircnumsetup`을 주면 원래의 기본값으로 돌아가는 것이 아니라 이 명령의 인자로 주어진 설정으로 되돌아간다. 단, 이 명령의 인자에서 지정되지 않은 key 옵션은 복원되지 않는다. 문서의 앞부분에 선언한 다음 마치 기본값처럼 사용할 수 있다. preamble과 문서 중간 어디서나 가능하다.

다음 보기는 `\settikzcircnumsetup`을 테스트하기 위하여 조금 큰 크기의 원숫자를 만들어서 기본값처럼 사용하는 예를 보이겠다. `reset`을 위하여 크기만이 아니라 다른 옵션도 미리 설정해두면 좋다. `\settikzcircnumsetup`을 선언한 후에 `reset`하면 새로이 선언된 설정으로 되돌아가는 것을 주의하여 보라.

```

\settikzcircnumsetup{shape=circle,color=white,sep=0.5pt,
    raise=-0.2ex,bdcolor=gray,linewidth=0.4pt,
    font={\sffamily\Large},base=00}
\tikzcircnum{2},
\tikzcircnum[ball]{3},
\tikzcircnum[reset]{4}

```

②, ③, ④

## 복원 명령

```

\declaretikzcircnumsmallsetup
\declaretikzcircnumbigsetup

```

\declaretikzcircnumsmallsetup과 \declaretikzcircnumbigsetup은 어떤 상황에서도 [tikzsmall]과 [tikzbig]의 초기 설정으로 강제로 되돌리는 명령이다.

[tikzbig]이나 [tikzsmall] 옵션이 주어지면 \tikzcircnum은 원숫자를 크거나 작은 모양으로 식자하는데, 중간에 \settikzcircnumsetup을 이용한다든가 하여 바뀐 모양을 쓰다가 다시 원래의 모양으로 돌아오게 하기 위한 명령이다.

다음은 이 과정 전체를 시험해본 것이다.

```

%% tikzsmall default
\declaretikzcircnumsmallsetup
\circenum{2}
\circenumsetup{shape=ball}\circenum{3}
\circenum*{4}
\circenumsetup{colorinv=blue}\circenum{5}
\restorecircnumsetup
\circenum{6}
\declaretikzcircnumbigsetup
\circenum{7}
\circenum[ball]{8}
\circenum[reset,colorinv=blue]{9}
\declaretikzcircnumsmallsetup
\circenum{10}

```

② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

## 2.6 기본 명령

이 세 종류의 원숫자 명령 가운데 사용자는 사실상 한 종류만 쓰면 된다. 그것을 \circenum으로 명명해두었다.

표 1: 이 패키지가 제공하는 명령

	원숫자의 식자	옵션 사전 설정	옵션 초기화
대표명령	<code>\circ</code>	<code>\circsetup</code>	<code>\restorecircsetup</code>
<code>[hcr]</code> *	<code>\hrcirc</code>	<code>\hrcircsetup</code>	<code>\restorehrcircsetup</code>
<code>[hz]</code>	<code>\hzcirc</code>	<code>\hzcircsetup</code>	<code>\restorehzcircsetup</code>
<code>[tikz]</code>	<code>\tikzcirc</code>	<code>\tikzcircsetup</code>	<code>\restoretikzcircsetup</code>
카운터	<code>\Cnum</code>		

```
\circ[ <keyval list> ]{number}
\circ*[ ... ]{number}
```

패키지 옵션이 없으면 `\hrcirc`이 `\circ` 명령으로 된다. 패키지 옵션으로 `[hz]`를 선언하면 `\circ=\hzcirc`이 되고, 마찬가지로 `[tikz]`를 선언하면 `\circ=\tikzcirc`이 된다.

```
\circsetup{ <keyval list> }
\restorecircsetup
```

각 기본 명령의 디폴트 값을 바꾸거나 복원하는 명령이다. 사용되는 옵션값들은 기본 명령이 무엇으로 정의되었느냐에 따라 다르다. 표 2를 참고하라.

### 3 카운터와 함께 사용하기

ko<sub>TeX</sub>의 카운터 수식자 `\onum`은 인자로 주어진 카운터 변수가 1부터 15까지일 때 원숫자를 만들어준다.<sup>7</sup> 다음과 같이 사용한다. 임의의 카운터를 `foo`라 하면,

```
\newcounter{foo}
\stepcounter{foo} \onum{foo}
\addtocounter{foo}{3} \onum{foo}
\setcounter{foo}{9} \onum{foo}
```

- ① ④ ⑨

```
\Cnum{ <counter> }
```

---

<sup>7</sup>kotexdoc 문서를 참고하라.



표 2: 명령군별로 사용가능한 keyval 옵션 (default=\*)

	<code>\hrcircnum</code>	<code>\hzcircnum</code>	<code>\tikzcircnum</code>
shape	circle* rectangle — —	circle* rectangle oval ball	circle* rectangle (rectangle+round) ball
color	<color>, white*	<color>, white*	<color>, white*
bgcolor	—	<color>, white*	<color>, white*
fgcolor	—	<color>, black*	<color>, black*
bdcolor	—	—	<color>
sep	—	<length>, 0.25pt*	<length>, 0.25pt*
sepplus	—	—	<length>
raise	—	<length>	<length>
Raise	—	<length>, -0.5ex*	<length>, -0.6ex*
font	<font>	<font>	<font>
fontplus	—	<font>	<font>
base	—	<text>	<text>
round	—	—	<length>, 2pt*
bdwidth	—	—	<length>, 0.4pt*

이런 식의 카운터 수식자로 쓸 수 있도록 `\Cnum` 명령을 제공한다. `\circ`가 보통 숫자를 인수로 취하는 것과 달리 이 명령은 반드시 카운터 변수가 와야 한다. 원숫자를 만드는 것은 현재의 `\circ`를 이용하므로 패키지 옵션에 따라 동작이 달라질 것이다. 원숫자의 모양은 `\dots` 명령을 이용하면 된다.

```
\setcounter{foo}{5} \Cnum{foo}
\addtocounter{foo}{15} \Cnum{foo}
\circsetup{color=red} \stepcounter{foo} \Cnum{foo}
```

⑤ ②① ②①

## 4 enumerate에서 사용하기

### 4.1 \Cnum 카운터 수식자를 이용하기

enumerate 패키지를 로드하였거나 `oblivoir` 문서라면 enumerate 환경에 옵션 인자를 사용할 수 있다. 이를 이용하여 원숫자를 enumerate의 글머리에 둘 수 있다. 카운터 수식자 `\Cnum`을 이용한다.<sup>8</sup>

<sup>8</sup>이 방식을 쓰면 참조(ref)에 아이템 라벨이 쓰일 수 없다는 경고가 나올 수 있다. ‘숫자’를 지정하지 않았기 때문에 발생하는 일이다. 4.3 소절을 보라.

```

\begin{enumerate}[\Cnum{enumi}]
\item 첫번째 아이템
\item 두번째 아이템
\item 세번째 아이템
\end{enumerate}

```

- ① 첫번째 아이템
- ② 두번째 아이템
- ③ 세번째 아이템

한편, enumitem 패키지를 이용하는 경우에는 preamble에

```

\usepackage{enumitem}
\AddEnumerateCounter{\Cnum}{\Cnum}{aa}

```

라고 선언한 후에,

```

\begin{enumerate}[label={\Cnum*}]
\item 첫번째 아이템
\item 두번째 아이템
\item 세번째 아이템
\end{enumerate}

```

처럼 하면 같은 결과를 얻을 수 있다. 단 enumerate 방식과 enumitem 방식을 동시에 쓸 수는 없다.

항목 머리 원숫자의 모양을 바꾸어야 할 경우, 옵션 변경 명령을 사용한다. 다음 예시는 `\circenum=\tikzcircnum`인 경우이다.

```

\circenumsetup{sep=2pt,fgcolor=white,shape=rectangle,
               bgcolor=blue!80,round,raise=-.2pt}
\begin{enumerate}[\Cnum{enumi}]
\item 첫번째 아이템
\item 두번째 아이템
\item 세번째 아이템
\end{enumerate}

```

- 1 첫번째 아이템
- 2 두번째 아이템
- 3 세번째 아이템

## 4.2 다른 방법

oblivoir의 enumerate 방식을 쓸 때, 카운터 수식자 \Cnum을 이용하지 않고 숫자에 대하여 다음처럼 하는 방법도 있다.

```
\begin{enumerate}[\expandafter\circenum1]
\item 첫번째 아이TEM
\item 두번째 아이TEM
\item 세번째 아이TEM
\end{enumerate}
```

- ① 첫번째 아이TEM
- ② 두번째 아이TEM
- ③ 세번째 아이TEM

옵션을 지정하려면 다음과 같이 할 수 있다.

```
% tikzsmall-default
\begin{enumerate}[%
\circnumsetup{shape=circle,fgcolor=orange!10,bgcolor=blue}%
\expandafter\circnum1]
\item 첫번째 아이TEM
\item 두번째 아이TEM
\item 세번째 아이TEM
\end{enumerate}
```

- ① 첫번째 아이TEM
- ② 두번째 아이TEM
- ③ 세번째 아이TEM

명령군 각각에 대하여 비슷한 방식으로 할 수 있다.

```
\begin{enumerate}[%
\hzcircnumsetup{shape=oval,color=red,sep=1.5pt}%
\expandafter\hzcircnum1]
\item 첫번째 아이TEM
\item 두번째 아이TEM
\item 세번째 아이TEM
\end{enumerate}
```

- ① 첫번째 아이TEM
- ② 두번째 아이TEM
- ③ 세번째 아이TEM

### 4.3 아이টে을 참조하기

아이টে에 label을 달고 나중에 그것을 참조하려 할 때가 있다. 피참조자를 원숫자로 표현하고자할 때, 앞서 보인 `\Cnum{enumi}`를 쓰면 아이টে에서 카운트되지 않기 때문에 쓸 수가 없다. 그러나 `\circenum1` 방식을 쓰면 이것을 가능하게 할 수 있다.

`refcount` 패키지를 이용하여, `preamble`에서 다음과 같이 `\cnumref` 명령을 정의한다.

```
\usepackage{refcount}
\newcounter{cnumrefcnt}
\newcommand*\cnumref[1]{%
  \setcounterref{cnumrefcnt}{#1}\Cnum{cnumrefcnt}%
}
```

이제 다음 코드를 보자.

```
\begin{enumerate}[\expandafter\circenum1]
\item item 1
\item \label{itm:test} item 2
\item item 3
\end{enumerate}
now, it's \cnumref{itm:test}.
```

- ① item 1
- ② item 2
- ③ item 3

now, it's ②.

당연히, `\cnumref` 결과는 두번째로 컴파일할 때 나타난다.

## 5 섹션 타이틀에 쓰는 방법

이렇게까지 해야 하는가 생각이 든다. 결코 좋은 발상은 아니지만 굳이 해보겠다면..., memoir나 oblvioir에서는 다음과 같이 간단하게 할 수 있다. `\circnumonce`는 [tikz] 또는 [tikzsmall] 패키지 옵션을 주었을 때 사용할 수 있는 명령이다.

```
\setsechook{%
  \setsecnumformat{%
    \circnumonce[colorinv=blue,font={\bfseries\Large},base={00},raise=-1pt]%
    \csname the##1\endcsname\quad}}
\section{test}
```

이 문서의 다음 절이 이 모양으로 나타날 것이다.

## 6 기타

이 패키지와 문서는 public domain이다. 오류 지적, 개선에 대한 토론은 [KTUG 게시판](#)을 통해서 하기를 원한다.

## 7 변경 사항

2014/06/16 ver1.2: `\hcr...`에 변경사항. `\hrcircnum`의 기본 코드를 재구현.

2014/06/15 ver1.1: `\tikz...`에 변경사항. `[tikz]`를 `[tikzsmall]`과 같은 의미로 하고 `[tikzbig]`을 새로 만들. `linewidth` 옵션 추가.

2014/06/09 ver1.0: `\tikz...`에 변경사항. 기본명령을 `\circ`와 같이 소문자로 바꿈.

2014/06/08 ver0.8: `[tikzsmall]` 추가. `\declaretikzcircnumsmallsetup`.

2014/06/07 ver0.7: `\tikz...`에 `\settikzcircnumsetup` 추가.

2014/06/07 ver0.6: `\tikz...`에 `round` 옵션과 `ball` 추가.

2014/06/07 ver0.5: `\hz...` 코드 개편. 별표붙은 명령 도입.

2014/06/06 ver0.4: `\tikzcircnum`이 두 자릿수 이상이 될 때 크기를 미세조정함.

2014/06/06 ver0.3: `\hzcircnum`의 옵션 `<key>`를 다른 명령과 일치하도록 수정.

2014/06/06 ver0.2: 버그 수정.

## 8 구현

`hcrnumbers`의 정의를 흉내내어 다시 구현한다. 핵심은 숫자를 분석하여 자릿수와 색상, 모양에 따라 적절한 Annotation 번호를 얻어내는 것이다. `hcrnumbers`에서는 예컨대 `\circ 234`와 같은 선언에서 이 숫자를 세 개의 토큰으로 보고 `\XXXcirc 234`로 처리하는데 여기서는 그렇게 하지 않고 숫자를 분석해서 자릿수를 각각 얻어내는 방법을 취했다. 확실히 위의 방법에 비하여 계산이 복잡해지는 감이 있기는 하나 Expl3의 `\int_div_truncate:nn`의 도움으로 코딩 자체는 매우 쉽다.

```
1 \cs_new:Npn \check_batang_dotum:
2 {
3   \tl_if_eq:NNTF \f@family \rmdefault
4   {
5     \tl_set:Nn \this_hcr_font { HCR Batang LVT }
6   }
7   {
8     \tl_set:Nn \this_hcr_font { HCR Dotum LVT }
9   }
10 }
```

```

11
12 \cs_new:Npn \fontspec_with_annotation:n #1
13 {
14   \fontspec [ Annotation = #1 ] { \this_hcr_font }
15 }
16
17 \cs_new:Npn \get_anno_mark:nnn #1 #2 #3
18 {
19   \tl_clear:N \treat_option_str
20   \str_case:nnF { #1 }
21   {
22     { one   } { \tl_put_right:Nn \treat_option_str { #1 } }
23     { two   } { \tl_put_right:Nn \treat_option_str { #1 } }
24     { three } { \tl_put_right:Nn \treat_option_str { #1 } }
25   }
26   {
27     \tl_set:Nn \treat_option_str { unknown }
28   }
29
30   \str_case:nnF { #2 }
31   {
32     { circle } { \tl_put_right:Nn \treat_option_str { #2 } }
33     { box    } { \tl_put_right:Nn \treat_option_str { #2 } }
34     { rectangle } { \tl_put_right:Nn \treat_option_str { box } }
35   }
36   {
37     \tl_set:Nn \treat_option_str { unknown }
38   }
39
40   \str_case:nnF { #3 }
41   {
42     { white } { \tl_put_right:Nn \treat_option_str { #3 } }
43     { black } { \tl_put_right:Nn \treat_option_str { #3 } }
44   }
45   {
46     \tl_set:Nn \treat_option_str { unknown }
47   }
48
49   \str_case_x:nn \treat_option_str
50   {
51     { onecirclewhite } { \tl_set:Nn \anno_ind { 0 } }
52     { onecircleblack } { \tl_set:Nn \anno_ind { 4 } }
53     { oneboxwhite    } { \tl_set:Nn \anno_ind { 14 } }
54     { oneboxblack    } { \tl_set:Nn \anno_ind { 17 } }
55     { twocirclewhite } { \tl_set:Nn \anno_left { 10 }
56                       \tl_set:Nn \anno_right { 11 } }
57     { twocircleblack } { \tl_set:Nn \anno_left { 12 }
58                       \tl_set:Nn \anno_right { 13 } }
59     { twoboxwhite    } { \tl_set:Nn \anno_left { 15 }
60                       \tl_set:Nn \anno_right { 16 } }
61     { twoboxblack    } { \tl_set:Nn \anno_left { 18 }

```

```

62         \tl_set:Nn \anno_right { 19 } }
63     { threecirclewhite }
64     { \tl_set:Nn \anno_left { 20 }
65       \tl_set:Nn \anno_mid { 21 }
66       \tl_set:Nn \anno_right { 22 } }
67     { threecircleblack }
68     { \tl_set:Nn \anno_left { 23 }
69       \tl_set:Nn \anno_mid { 24 }
70       \tl_set:Nn \anno_right { 25 } }
71     { threeboxwhite }
72     { \tl_set:Nn \anno_left { 26 }
73       \tl_set:Nn \anno_mid { 21 }
74       \tl_set:Nn \anno_right { 22 } }
75     { threeboxblack }
76     { \tl_set:Nn \anno_left { 27 }
77       \tl_set:Nn \anno_mid { 28 }
78       \tl_set:Nn \anno_right { 29 } }
79   }
80 }
81
82 \cs_new:Npn \number_into_digits:n #1
83 {
84   \bool_if:nT
85     { \int_compare_p:n { #1 >= 0 } && \int_compare_p:n { #1 < 10 } }
86     { \tl_set:Nn \hcrnum_digit_one { #1 } }
87
88   \bool_if:nT
89     { \int_compare_p:n { #1 >= 10 } && \int_compare_p:n { #1 < 100 } }
90     {
91       \tl_set:Nn \hcrnum_digit_two_left { \int_div_truncate:nn { #1 } { 10 } }
92       \tl_set:Nn \hcrnum_digit_two_right
93         { \int_eval:n { #1 - 10 * \int_div_truncate:nn { #1 } { 10 } } }
94     }
95
96   \bool_if:nT
97     { \int_compare_p:n { #1 >= 100 } && \int_compare_p:n { #1 < 1000 } }
98     {
99       \tl_set:Nn \hcrnum_digit_three_left
100        { \int_div_truncate:nn { #1 } { 100 } }
101       \tl_set:Nn \hcrnum_digit_three_mid
102        { \int_eval:n
103          { \int_div_truncate:nn
104            { #1 - 100 * \int_div_truncate:nn { #1 } { 100 } } { 10 } } }
105       \tl_set:Nn \hcrnum_digit_three_right
106        { \int_eval:n { #1 - 10 * \int_div_truncate:nn { #1 } { 10 } } }
107     }
108 }
109
110 \DeclareDocumentCommand \hcrannonum { m m m }
111 {
112   \if_mode_vertical: \leavevmode \fi:

```

```

113 \number_into_digits:n { #3 }
114
115 \bool_if:nT
116 { \int_compare_p:n { #3 >= 0 } && \int_compare_p:n { #3 < 10 } }
117 {
118   \get_anno_mark:nnn { one } { #1 } { #2 }
119   \group_begin:
120   \l_tmpa_circnum_font \check_batang_dotum:
121   \str_if_eq_x:nnF { nocolor } \l_tmpa_circnum_color
122     { \color{ \l_tmpa_circnum_color } }
123   \fontspec_with_annotation:n \anno_ind \hcrnum_digit_one
124   \group_end:
125 }
126
127 \bool_if:nT
128 { \int_compare_p:n { #3 >= 10 } && \int_compare_p:n { #3 < 100 } }
129 {
130   \get_anno_mark:nnn { two } { #1 } { #2 }
131   \group_begin:
132   \l_tmpa_circnum_font \check_batang_dotum:
133   \str_if_eq_x:nnF { nocolor } \l_tmpa_circnum_color
134     { \color{ \l_tmpa_circnum_color } }
135   \fontspec_with_annotation:n \anno_left \hcrnum_digit_two_left
136   \hbox_overlap_left:n {
137     \fontspec_with_annotation:n \anno_right \hcrnum_digit_two_right
138   }
139   \group_end:
140 }
141
142 \bool_if:nT
143 { \int_compare_p:n { #3 >= 100 } && \int_compare_p:n { #3 < 1000 } }
144 {
145   \get_anno_mark:nnn { three } { #1 } { #2 }
146   \group_begin:
147   \l_tmpa_circnum_font \check_batang_dotum:
148   \str_if_eq_x:nnF { nocolor } \l_tmpa_circnum_color
149     { \color{ \l_tmpa_circnum_color } }
150   \fontspec_with_annotation:n \anno_left \hcrnum_digit_three_left
151   \hbox_overlap_left:n {
152     \fontspec_with_annotation:n \anno_mid \hcrnum_digit_three_mid
153   }
154   \hbox_overlap_left:n {
155     \fontspec_with_annotation:n \anno_right \hcrnum_digit_three_right
156   }
157   \group_end:
158 }
159 }

```

사용자 인터페이스 명령 `\hrcircnum`은 `shape`에 주어진 `value`와 `color`에 주어진 `value`를 조합하여 `\hcrnumber<shapevalue><colorvalue>`로 cs 이름을 합친 다음 호출한다.



```

1 \keys_define:nn { hcrsetup }
2 {
3   shape .tl_set:N = \l_tmpa_circnum_shape,
4   shape .default:n = circle,
5   hrcolor .tl_set:N = \l_tmpa_circnum_hrcolor,
6   hrcolor .default:n = white,
7   circle .meta:n = { shape = circle },
8   box .meta:n = { shape = rectangle },
9   rectangle .meta:n = { shape = rectangle },
10  white .meta:n = { hrcolor = white },
11  black .meta:n = { hrcolor = black },
12  reset .code:n = { \restorehrcircnumsetup },
13  font .tl_set:N = \l_tmpa_circnum_font,
14  color .code:n =
15    { \str_case:nnTF { #1 }
16      { { black } { \tl_set:Nn \l_tmpa_circnum_hrcolor { black } }
17        { white } { \tl_set:Nn \l_tmpa_circnum_hrcolor { white } } }
18      { \tl_set:Nn \l_tmpa_circnum_color { nocolor } }
19      { \tl_set:Nn \l_tmpa_circnum_color { #1 } }
20    },
21 }
22
23 \DeclareDocumentCommand \hrcircnum { s o m }
24 {
25   \IfBooleanT { #1 }
26   {
27     \keys_set:nn { hcrsetup }
28     {
29       shape = circle,
30       color = white,
31       font = {\normalfont},
32     }
33   }
34   \IfValueT { #2 }
35   {
36     \keys_set:nn { hcrsetup } { #2 }
37   }
38   \cs:w
39     hcrnumber
40     \l_tmpa_circnum_shape
41     \l_tmpa_circnum_hrcolor
42   \cs_end: { #3 }
43 }
44
45 \DeclareDocumentCommand \hrcircnumsetup { m }
46 { \keys_set:nn { hcrsetup } { #1 } }
47
48 \DeclareDocumentCommand \restorehrcircnumsetup { }
49 {
50   \keys_set:nn { hcrsetup }
51   {

```

```

52     shape = circle,
53     color = white,
54     font = {\normalfont},
55   }
56 }

```

hzmisc에 정의된 \wrapnum의 실제 구현 명령인 \ovalnum, \rectnum, \cirnum, \ballnum의 코드를 가져와서 일곱 개의 인자를 받아들이는 명령으로 수정한다.

```

1  \ExplSyntaxOff
2  \newcommand*\hzovalnum[7]{%
3    \raisebox{#1}{\abnormalparskip{0pt}%
4      \tikz\node[%
5        rectangle,draw=#2,thin,inner sep=#3+1pt,%
6        rounded corners=3pt,%
7        top color=#4,bottom color=#4,%
8        text width=#5,%
9        font=#6,text badly centered,#2%
10     ]{#7};}}
11 \newcommand*\hzrectanglenum[7]{%
12   \raisebox{#1}{\abnormalparskip{0pt}%
13     \tikz\node[%
14       rectangle,draw=#2,thin,inner sep=#3+1pt,%
15       top color=#4,bottom color=#4,%
16       text width=#5,%
17       font=#6,text badly centered,#2%
18     ]{#7};}}
19 \newcommand*\hzcircnum[7]{%
20   \raisebox{#1}{\abnormalparskip{0pt}%
21     \tikz\node[%
22       circle,draw=#2,thin,inner sep=#3,%
23       top color=#4,bottom color=#4,%
24       text width=#5,%
25       font=#6,text badly centered,#2%
26     ]{#7};}}
27 \newcommand*\hzballnum[7]{%
28   \raisebox{#1}{\abnormalparskip{0pt}%
29     \tikz\node[%
30       circle,shade,draw=#4,thin,inner sep=#3,%
31       ball color=#4,%
32       text width=#5,%
33       font=#6,text badly centered,#2%
34     ]{#7};}}

```

이제 \hzcircnum을 정의하여 key-val list를 처리한 후 이들 일곱 개의 인자에 해당하는 값을 얻어내어 shape의 경우에 따라 위의 명령을 호출한다.

```

1  \dim_new:N \l_hz_sep_dim
2  \dim_new:N \l_hz_raise_dim
3  \dim_new:N \l_hz_raise_default_dim
4  \dim_set:Nn \l_hz_raise_default_dim { -0.5ex }
5  \dim_new:N \l_hz_width_dim
6

```

```

7 \cs_new:Npn \hz_treat_color_option:n #1
8 {
9   \str_case:nnF { #1 }
10  {
11    { white } { \tl_set:Nn \l_hz_bgcolor { white }
12               \tl_set:Nn \l_hz_fgcolor { black } }
13    { black } { \tl_set:Nn \l_hz_bgcolor { black }
14               \tl_set:Nn \l_hz_fgcolor { white } }
15  }
16  {
17    \tl_set:Nn \l_hz_bgcolor { #1 }
18    \tl_set:Nn \l_hz_fgcolor { white }
19  }
20 }
21
22 \cs_new:Npn \hz_treat_base_option:n #1
23 {
24   \settowidth \l_hz_width_dim { \l_hz_font #1 }
25 }
26
27 \keys_define:nn { hzsetup }
28 {
29   shape .tl_gset:N = \l_hz_shape,
30   color .code:n = { \hz_treat_color_option:n { #1 } },
31   bgcolor .tl_set:N = \l_hz_bgcolor,
32   fgcolor .tl_set:N = \l_hz_fgcolor,
33   font .tl_set:N = \l_hz_font,
34   fontplus .code:n = { \tl_put_right:Nn \l_hz_font { #1 } },
35   sep .dim_set:N = \l_hz_sep_dim,
36   Raise .dim_set:N = \l_hz_raise_dim,
37   raise .code:n = { \dim_set:Nn \l_hz_raise_dim { #1 + \l_hz_raise_default_dim } },
38   base .code:n = { \hz_treat_base_option:n { #1 } },
39   circle .meta:n = { shape = circle },
40   circ .meta:n = { shape = circle },
41   rectangle .meta:n = { shape = rectangle },
42   rect .meta:n = { shape = rectangle },
43   box .meta:n = { shape = rectangle },
44   oval .meta:n = { shape = oval },
45   ball .meta:n = { shape = ball },
46   white .meta:n = { color = white },
47   black .meta:n = { color = black },
48   reset .code:n = { \restorehzcircumsetup },
49 }
50
51 \cs_new:Npn \draw_hz_circ_num:nn #1 #2
52 {
53   \cs:w hz #1 num\cs_end:
54     { \l_hz_raise_dim } % raise
55     { \l_hz_fgcolor } % fgcolor
56     { \l_hz_sep_dim } % sep
57     { \l_hz_bgcolor } % bgcolor

```

```

58     { \l_hz_width_dim } % width
59     { \l_hz_font       } % font
60     { #2 }
61 }
62
63 \DeclareDocumentCommand \hzcircnum { s o m }
64 {
65     \IfBooleanT { #1 }
66     {
67         \restorehzcircnumsetup
68     }
69     \IfValueT { #2 }
70     {
71         \keys_set:nn { hzsetup } { #2 }
72     }
73
74     \str_case:x:nnF { \l_hz_shape }
75     {
76         { circle } { }
77         { circ   } { \tl_set:Nn \l_hz_shape { circle } }
78         { rectangle } { }
79         { rect    } { \tl_set:Nn \l_hz_shape { rectangle } }
80         { oval    } { }
81         { ball    } { }
82     }
83     {
84         \tl_set:Nn \l_hz_shape { circle }
85     }
86     \draw_hz_circ_num:nn { \l_hz_shape } { #3 }
87 }
88
89 \DeclareDocumentCommand \hzcircnumsetup { m }
90 {
91     \keys_set:nn { hzsetup } { #1 }
92 }
93
94 \DeclareDocumentCommand \restorehzcircnumsetup { }
95 {
96     \hzcircnumsetup {
97         shape = circle,
98         color  = white,
99         font   = { \sffamily\footnotesize },
100        sep    = { 0.25pt },
101        Raise  = { \l_hz_raise_default_dim },
102        base   = { 99 }
103    }
104 }
105
106 %% initialize once
107 \restorehzcircnumsetup

```

\tikzcircnum은 \hzcircnum과 거의 같은 방식이지만 하위 명령을 내부명령으로 하고 분기에 따른 호출 방식이 조금 다르다.

```

1 \dim_new:N \tcn_raise_dim
2 \dim_new:N \tcn_raise_default_dim
3 \dim_set:Nn \tcn_raise_default_dim { -0.6ex }
4 \dim_new:N \tcn_innersep_dim
5 \dim_new:N \tcn_width_dim
6 \dim_new:N \tcn_width_fix_dim
7 \dim_new:N \tcn_round_dim
8 \dim_new:N \tcn_linewidth_dim
9
10 %%% default values
11 \tl_set:Nn \tcn_shape { circle }
12 \tl_set:Nn \tcn_bgcolor { white }
13 \tl_set:Nn \tcn_fgcolor { black }
14 \tl_set:Nn \tcn_bdcolor { gray }
15 \tl_set:Nn \tcn_font { \sffamily\small }
16
17 \cs_new:Npn \tcn_set_width:nn #1 #2
18 {
19   \settowidth \tcn_width_dim { #1 #2 }
20   \dim_sub:Nn \tcn_width_dim { 1pt - \tcn_width_fix_dim }
21 }
22
23 \cs_new:Npn \tcn_color_option:n #1
24 {
25   \str_case:nnF { #1 }
26   {
27     { white } { \tl_set:Nn \tcn_fgcolor { black }
28                 \tl_set:Nn \tcn_bgcolor { white }
29                 \tl_set:Nn \tcn_bdcolor { black } }
30     { black } { \tl_set:Nn \tcn_fgcolor { white }
31                 \tl_set:Nn \tcn_bgcolor { black }
32                 \tl_set:Nn \tcn_bdcolor { black } }
33   }
34   {
35     \tl_set:Nn \tcn_fgcolor { #1 }
36     \tl_set:Nn \tcn_bgcolor { white }
37     \tl_set:Nn \tcn_bdcolor { #1 }
38   }
39 }
40
41 \cs_new:Npn \tcn_inverse_color_option:n #1
42 {
43   \str_case:nnF { #1 }
44   {
45     { white } { \tl_set:Nn \tcn_fgcolor { black }
46                 \tl_set:Nn \tcn_bgcolor { white }
47                 \tl_set:Nn \tcn_bdcolor { black } }
48     { black } { \tl_set:Nn \tcn_fgcolor { white }
49                 \tl_set:Nn \tcn_bgcolor { black } }

```

```

50         \tl_set:Nn \tcn_bdcolor { black } }
51     }
52     {
53         \tl_set:Nn \tcn_fgcolor { white }
54         \tl_set:Nn \tcn_bgcolor { #1 }
55         \tl_set:Nn \tcn_bdcolor { #1 }
56     }
57 }
58
59
60 \keys_define:nn { tccnm }
61 {
62     shape      .tl_set:N      = \tcn_shape,
63     fgcolor    .tl_set:N      = \tcn_fgcolor,
64     bgcolor    .tl_set:N      = \tcn_bgcolor,
65     bdcolor    .tl_set:N      = \tcn_bdcolor,
66     linecolor  .meta:n        = { bdcolor = #1 },
67     color      .code:n        = { \tcn_color_option:n { #1 } },
68     colorinv   .code:n        = { \tcn_inverse_color_option:n { #1 } },
69     font       .tl_set:N      = \tcn_font,
70     font       .default:n     = { \sffamily\small },
71     fontplus   .code:n        = { \tl_put_right:Nn \tcn_font { #1 } },
72     Raise      .dim_set:N     = \tcn_raise_dim,
73     raise      .code:n        =
74         { \dim_set:Nn \tcn_raise_dim { #1 + \tcn_raise_default_dim } },
75     sep        .code:n        = { \dim_set:Nn \tcn_innersep_dim { #1 + 0.25pt } },
76     sep        .default:n     = { 0.25pt },
77     sepplus    .code:n        =
78         { \dim_set:Nn \tcn_innersep_dim { #1 + 0.25pt + \tcn_innersep_dim } },
79     round      .dim_set:N     = \tcn_round_dim,
80     round      .default:n     = { 2pt },
81     linewidth  .dim_set:N     = \tcn_linewidth_dim,
82     linewidth  .default:n     = { 0.4pt },
83     bdwidth    .meta:n        = { linewidth = #1 },
84     circle     .meta:n        = { shape = circle },
85     circ       .meta:n        = { shape = circle },
86     rectangle  .meta:n        = { shape = rectangle },
87     rect       .meta:n        = { shape = rectangle },
88     box        .meta:n        = { shape = rectangle },
89     ball       .meta:n        = { shape = ball },
90     white      .meta:n        = { color = white },
91     black      .meta:n        = { color = black },
92     width      .dim_set:N     = \tcn_width_dim,
93     base       .code:n        = { \tcn_set_width:nn { \tcn_font } { #1 } },
94     reset      .code:n        = { \restoretikzcircnumsetup },
95     widthfix   .dim_set:N     = \tcn_width_fix_dim,
96 }
97
98 \DeclareDocumentCommand \tikzcircnumsetup { m }
99 {
100     \keys_set:nn { tccnm } { #1 }

```

```

101 }
102
103 \DeclareDocumentCommand \restoretikzcircnumsetup { }
104 {
105   \keys_set:nn { tccnm }
106   {
107     shape = circle,
108     bgcolor = white,
109     fgcolor = black,
110     bdcolor = gray,
111     font = { \sffamily\small },
112     Raise = { \tcn_raise_default_dim },
113     sep = { 0.25pt },
114     round = { 0pt },
115     base = { 99 },
116     widthfix = { 0pt },
117     linewidth = { 0.4pt },
118   }
119 }
120
121 %% now, initialize once
122 \restoretikzcircnumsetup
123
124 \cs_new:Npn \draw_tikz_circnum:n #1
125 {
126   \if_mode_vertical: \leavevmode \fi:
127   \raisebox{\tcn_raise_dim}
128   {
129     \tikz\node
130     [
131       \tcn_shape,
132       draw=\tcn_bdcolor,
133       line~width=\tcn_linewidth_dim,
134       inner~sep=\tcn_innersep_dim,
135       top~color=\tcn_bgcolor,
136       bottom~color=\tcn_bgcolor,
137       text~width=\tcn_width_dim,
138       text~badly~centered,
139       \tcn_fgcolor,
140       font={\tcn_font}
141     ] { #1 };
142   }
143 }
144
145 \cs_new:Npn \draw_tikz_rectnum:n #1
146 {
147   \if_mode_vertical: \leavevmode \fi:
148   \raisebox{\tcn_raise_dim}
149   {
150     \tikz\node
151     [

```

```

152         rectangle,
153         draw=\tcn_bdcolor,
154         line~width=\tcn_linewidth_dim,
155         rounded~corners=\tcn_round_dim,
156         inner~sep=\tcn_innerep_dim+1pt,
157         top~color=\tcn_bgcolor,
158         bottom~color=\tcn_bgcolor,
159         text~width=\tcn_width_dim,
160         text~badly~centered,
161         \tcn_fgcolor,
162         font={\tcn_font}
163     ] { #1 };
164 }
165 }
166
167 \cs_new:Npn \draw_tikz_ballnum:n #1
168 {
169     \if_mode_vertical: \leavevmode \fi:
170     \raisebox{\tcn_raise_dim}
171     {
172         \tikz\node
173         [
174             circle, shade,
175             draw=\tcn_bdcolor,
176             line~width=\tcn_linewidth_dim,
177             ball~color=\tcn_bgcolor,
178             inner~sep=\tcn_innerep_dim,
179             text~width=\tcn_width_dim,
180             text~badly~centered,
181             \tcn_fgcolor,
182             font={\tcn_font}
183         ] { #1 };
184     }
185 }
186
187 \DeclareDocumentCommand \tikzcircnum { s o m }
188 {
189     \IfBooleanT { #1 }
190     {
191         \restoretikzcircnumsetup
192     }
193
194     \IfValueT { #2 }
195     {
196         \keys_set:nn { tccnm } { #2 }
197     }
198
199     \str_case_x:nnF { \tcn_shape }
200     {
201         { circle } { \draw_tikz_circnum:n { #3 } }
202         { rectangle } { \draw_tikz_rectnum:n { #3 } }

```



```

203     { ball      } { \draw_tikz_ballnum:n { #3 } }
204   }
205   {
206     \draw_tikz_circ_num:n { #3 }
207   }
208 }

```

\tikzcircnumonce는 현재 설정을 저장하고 실행한 후 다시 복원하는 것이다.

```

1  \tl_new:N \l_tikz_tmp_current_shape
2  \tl_new:N \l_tikz_tmp_current_bgcolor
3  \tl_new:N \l_tikz_tmp_current_fgcolor
4  \tl_new:N \l_tikz_tmp_current_bdcolor
5  \tl_new:N \l_tikz_tmp_current_font
6  \dim_new:N \l_tikz_tmp_current_raise
7  \dim_new:N \l_tikz_tmp_current_innersep
8  \dim_new:N \l_tikz_tmp_current_width
9  \dim_new:N \l_tikz_tmp_current_round
10 \dim_new:N \l_tikz_tmp_current_linewidth
11
12 \cs_new:Npn \preserve_current_tikzcircnum_settings:
13 {
14   \tl_set_eq:NN \l_tikz_tmp_current_shape \tcn_shape
15   \tl_set_eq:NN \l_tikz_tmp_current_bgcolor \tcn_bgcolor
16   \tl_set_eq:NN \l_tikz_tmp_current_fgcolor \tcn_fgcolor
17   \tl_set_eq:NN \l_tikz_tmp_current_bdcolor \tcn_bdcolor
18   \tl_set_eq:NN \l_tikz_tmp_current_font \tcn_font
19   \dim_set_eq:NN \l_tikz_tmp_current_raise \tcn_raise_dim
20   \dim_set_eq:NN \l_tikz_tmp_current_innersep \tcn_innersep_dim
21   \dim_set_eq:NN \l_tikz_tmp_current_width \tcn_width_dim
22   \dim_set_eq:NN \l_tikz_tmp_current_round \tcn_round_dim
23   \dim_set_eq:NN \l_tikz_tmp_current_linewidth \tcn_linewidth_dim
24 }
25
26 \cs_new:Npn \restore_current_tikzcircnum_settings:
27 {
28   \tl_set_eq:NN \tcn_shape \l_tikz_tmp_current_shape
29   \tl_set_eq:NN \tcn_bgcolor \l_tikz_tmp_current_bgcolor
30   \tl_set_eq:NN \tcn_fgcolor \l_tikz_tmp_current_fgcolor
31   \tl_set_eq:NN \tcn_bdcolor \l_tikz_tmp_current_bdcolor
32   \tl_set_eq:NN \tcn_font \l_tikz_tmp_current_font
33   \dim_set_eq:NN \tcn_raise_dim \l_tikz_tmp_current_raise
34   \dim_set_eq:NN \tcn_innersep_dim \l_tikz_tmp_current_innersep
35   \dim_set_eq:NN \tcn_width_dim \l_tikz_tmp_current_width
36   \dim_set_eq:NN \tcn_round_dim \l_tikz_tmp_current_round
37   \dim_set_eq:NN \tcn_linewidth_dim \l_tikz_tmp_current_linewidth
38 }
39
40 \DeclareDocumentCommand \tikzcircnumonce { o m }
41 {
42   \IfNoValueTF { #1 }
43   {
44     \tikzcircnum { #2 }

```

```

45     }
46     {
47     \preserve_current_tikzcircnum_settings:
48     \tikzcircnumsetup { #1 }
49     \tikzcircnum { #2 }
50     \restore_current_tikzcircnum_settings:
51     }
52 }

```

나머지 부분은 옵션을 처리하기 위한 부수적인 코드들이므로 생략한다.

## 9 wrapnum 패키지

이호재 님이 [http://www.ktug.org/xe/index.php?document\\_srl=178885](http://www.ktug.org/xe/index.php?document_srl=178885)에서 이미 보여주신 wrapnum으로 충분하다고 생각하실 분도 분명 계실 것이기에, wrapnum.sty 라는 것에 대해 여기 메모를 남겨둔다.

kocircnum에서 \wrapnum을 활용하기는 하였지만 거기서는 \wrapnum의 원래 모양과는 조금 다른 모양이 되어 버렸다. Myriad Pro 글꼴 부분을 없애버렸고 옵션 이름도 완전히 바뀌었기 때문에 종래 \wrapnum을 즐겨쓰시던 분들께는 아마 완전히 다른 명령이라는 생각이 들 수도 있다.

여기 올려두는 wrapnum.sty는 원래의 hzmisc에서 식자되는 모양과 동일하고, 옵션 명칭도 똑같이 유지했다. 본질적으로 이 패키지는 hzmisc의 \wrapnum 부분을 재구현한 것이다. 패키지를 작성한 로직은 kocircnum의 것에 가깝지만.

hzmisc 대응으로 이것을 쓰시면 될 것이다. kocircnum과 wrapnum을 함께 사용하는 것은 그렇게 할 필요가 없다고 본다. 정상 동작하지 않을 것이다.

### 9.1 사용법

```

\wrapnum[ <key val> ]{number}
\setwrapnum{ <key val> }

```

- ① Xe<sub>La</sub>TeX, Lua<sub>La</sub>TeX에서 Myriad Pro 글꼴을 요구한다. pdf<sub>La</sub>TeX이면 그냥 \sffamily
- ② \wrapnum 옵션의 key-value는 다음과 같다.

**type**=<circle/rectangle/oval/ball> 원, 사각형, 둥근 사각형, 입체감을 살린 원.

**background**=(color)

**foreground**=(color)

`color= <white/black/(color)>` 이 값이 black이나 white가 아니면 배경이 color가 되고 전경은 흰색으로 된다.

`font= (font)`

`base= (text)` 원의 크기를 결정한다. 예를 들어 `\base={999}`라고 하면 999라는 글자가 차지하는 폭을 기준으로 동그라미를 그린다.

`wrapspace= (length)` 숫자와 원 사이의 간격.

`raise= (length)` hzmisc의 raise와 용법이 같지만 동작이 조금 다르다. 패키지가 설정한 기본값을 기준으로 raise 길이를 추가한다.

③ 이 패키지에서 추가된 옵션 key-value는 다음과 같다.

`linecolor= (color)` 외곽선 색상을 별도로 지정한다.

`fontplus= (font)` 기존의 font 정의를 그대로 두고 지정된 값을 추가한다. font family는 그대로 두고 사이즈만 바꾼다든가 할 때 쓴다.

`Raise= (dim)` 이것은 hzmisc의 `\wrapnum`에서 raise라고 한 것이다. 이 패키지의 raise는 패키지가 설정한 기본 raise값에서 더하거나 빼기 때문에 `raisedim`의 절댓값을 주려 할 때 대문자로 시작하는 Raise를 지정한다.

`round= (dim)` oval에서만 의미가 있다. rounded corners 값을 설정한다.

`reset` 지금까지 설정된 모든 속성값을 전부 처음으로 되돌린다. 따라서 `wrapnum[reset,color=red]{number}`와 같이 하면 color 이외의 모든 속성값이 초기화된다. 옵션의 맨처음에 쓰는 것이 좋다.

④ 이 패키지에서 추가한 약식 옵션

`type=<circ/rect>` circle의 단축형 circ, rectangle의 단축형 rect를 인식한다.

circle, rectangle, oval, ball, circ, rect 일종의 약어이다. type= 으로 주지 않고 이것만 선언해도 된다.

white, black color=white, color=black으로 선언하지 않아도 된다.

## 9.2 기타

- `\setwrapnum` 명령은 hzmisc와 동일하게 동작한다. 단, `\setwrapnum{}` 과 같이 인자를 비워서 건네주면 모든 설정값을 초기화한다.

- font와 base를 지정할 때는 font를 base보다 먼저 옵션으로 주어야 한다.

```
[font={\tiny},base={333}]
```

- 원래의 hzmisc는 memoir에서만 동작한다. 이 wrapnum은 일반 클래스에서 오류가 발생하지 않도록 해두었으나 memoir/oblivoir를 권장한다.

## 10 hcrnumbers 패키지

Dohyun Kim의 hcrnumbers 원본이다. 역시 kocircnum과 함께 사용하려 할 필요가 없을 것이고 간단히 원숫자를 식자하기 위한 목적으로 이 패키지만 사용할 수 있다.

`\circlednumber`, `\Bcirclednumber`, `\boxednumber`, `\Bboxednumber` 네 개의 명령이 제공된다.

```
\circlednumber number
\Bcirclednumber number
\boxednumber number
\Bboxednumber number
```

숫자는 증괄호로 둘러싸지 말고 그냥 명령 뒤에 잇대어 쓴다.

```
\circlednumber 2
\Bcirclednumber 15
\boxednumber 3
\Bboxednumber 256
```

## 찾아보기

### A

Annotation, 3

### C

#### Commands

`\ballnum`, 26  
`\Bboxnumber`, 36  
`\Bcircnumber`, 36  
`\bfseries`, 4, 8  
`\boxnumber`, 36  
`\circlednumber`, 3, 21, 36  
`\circlednum`, 1, 3, 11, 15–18, 20, 21  
`\circlednum...`, 3  
`\circlednumonce`, 14, 20  
`\circlednumsetup`, 13, 16  
`\circlednum`, 26  
`\Cnum`, 1, 16, 17, 19, 20  
`\cnumref`, 20  
`\declaretikzcircnumbigsetup`,  
15  
`\declaretikzcircnumsmallsetup`,  
15, 21  
`\definecolor`, 7  
`\ding`, 39  
`\fontspec`, 3  
`\footnotesize`, 7  
`\hcr...`, 21  
`\hrcircnum`, 2–7, 16, 17, 21, 24,  
39  
`\hrcircnumsetup`, 6, 16  
`\hcrnumber`, 24  
`\hz...`, 21  
`\hzcircnum`, 4, 6, 9–13, 16, 17, 21,  
26, 29, 39  
`\hzcircnumsetup`, 9, 16  
`\itshape`, 12  
`\onum`, 2, 16, 39  
`\ovalnum`, 26  
`\recnum`, 26  
`\restorecircnumsetup`, 13, 16  
`\restorehrcircnumsetup`, 6, 16  
`\restorehzcircnumsetup`, 9, 16  
`\restoretikzcircnumsetup`, 13,  
14, 16  
`\rmfamily`, 4  
`\settikzcircnumsetup`, 14, 15, 21  
`...setup`, 17  
`\setwrapnum`, 35  
`\sffamily`, 4, 7, 34  
`\tikz...`, 14, 21  
`\tikzcircnum`, 1, 2, 10, 11, 13–18,  
21, 29  
`\tikzcircnum (big)`, 39  
`\tikzcircnum (small)`, 39  
`\tikzcircnumonce`, 14, 33  
`\tikzcircnumsetup`, 13, 16  
`\wrapnum`, 1, 2, 6, 26, 34, 35

### E

#### Engines

`koTeX`, 2, 16, 37  
`Lua®TeX`, 3  
`LuaTeX`, 34  
`pdfTeX`, 4, 34

X<sub>Y</sub>TeX, 3, 10

X<sub>Y</sub>TeX, 34

## K

### Keys

background, 34

ball, 7, 10, 35

base, 7, 13, 17, 35

bdcolor, 10, 13, 17

bdwidth, 13, 17

bgcolor, 7, 17

black, 4, 7, 35

box, 4

circ, 7, 35

circle, 4, 7, 10, 35

color, 4, 7, 10, 12, 17, 35

colorinv, 10, 12

fgcolor, 7, 17

font, 4, 7, 17, 35

fontplus, 7, 17, 35

foreground, 34

linecolor, 10, 13, 35

linewidth, 10, 13, 21

oval, 7, 10, 35

Raise, 7, 9, 13, 17, 35

raise, 7, 9, 13, 17, 35

rect, 7, 35

rectangle, 4, 7, 10, 35

reset, 4–7, 13, 14, 35

round, 10, 17, 21, 35

rounded corners, 10

sep, 7, 8, 11, 17

sepplus, 11, 17

shape, 4, 6, 7, 10, 17

type, 34, 35

white, 4, 7, 35

wrapspace, 35

## O

### Options

hcr, 3, 16

hz, 3, 16

tikz, 3, 13, 14, 16, 20, 21

tikzbig, 3, 10, 13, 15, 21

tikzsmall, 3, 11–15, 20, 21

## P

### Packages

color, 3

enumerate, 17–19

enumitem, 18

fontspec, 3

hcrnumbers, 1–3, 21, 36

hzguide, 2, 6

hzmisc, 1, 2, 6, 26, 34–36

kocircnum, 2, 34, 36

kotexdoc, 16

memoir, 20, 36

oblivoir, 17, 19, 20, 36

pifont, 2, 39

refcount, 20

tikz, 2, 9, 10

wrapnum, 2, 34, 36

xcolor, 3, 7

## T

thin, 10

## ㅎ

함초롬체 LVT, 2–4

## 테스트

1부터 10까지 원숫자의 모양 비교.

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

(Combinumerals)

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

`pifont \ding{192}+`

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

`\onum`

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

`\tikzcircnum (small)`

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

`\hrcircnum`

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

`\hzcircnum`

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

`\tikzcircnum (big)`