

# 스타일 작성을 위한 명령어 모음

2008 년 공주대 워크숍 강좌팀

2008 년 5 월 24 일

## 1 새 명령어의 정의와 기존 명령어의 재정의

```
\newcommand{\명령어이름}[인자개수]{명령어정의}  
\renewcommand{\명령어이름}[인자개수]{명령어정의}
```

새 명령어를 정의할 때에는 `\newcommand`를, 기존 명령어를 재정의할 때에는 `\renewcommand`를 사용한다. 명령어 정의에서 인자는 #1, #2와 같이 번호를 붙여 사용한다.

```
\newcommand{\명령어이름}[인자개수][기본값]{명령어정의}  
\renewcommand{\명령어이름}[인자개수][기본값]{명령어정의}
```

인자의 개수가 한 개 이상인 명령어를 정의/재정의할 때에 첫 번째 인자가 생략되었을 때 사용되는 기본값을 다음과 같이 정할 수 있다.

```
\providecommand{\명령어이름}[인자개수][기본값]{명령어정의}
```

`\providecommand`는 `\newcommand`와 비슷하지만, 같은 이름의 명령어가 이미 존재할 때에도 이를 무시하고 그 이름의 명령어를 새로 정의한다.

## 2 새 환경의 정의와 기존 환경의 재정의

```
\newenvironment{환경이름}[인수의 개수]{시작 부분}{끝 부분}  
\renewenvironment{환경이름}[인수의 개수]{시작 부분}{끝 부분}
```

새 환경을 정의할 때에는 `\newenvironment`를, 기존 명령어를 재정의할 때에는 `\renewenvironment`를 사용한다.

```
\newenvironment{환경이름}[인수의 개수][기본값]{시작 부분}{끝 부분}  
\renewenvironment{환경이름}[인수의 개수][기본값]{시작 부분}{끝 부분}
```

정의/재정의되는 환경의 인수가 한 개 이상일 때, 첫 번째 인수가 생략되었을 때 사용될 기본값을 지정할 수 있다.

```
\provideenvironment{환경이름}[인수의 개수][기본값]{시작 부분}{끝 부분}
```

`\provideenvironment`는 같은 이름의 환경이 이미 존재해도 무시하고 그 환경을 새로 정의한다.

### 3 조건문

```
\newif\조건변수이름
\iftrue
\iffalse
\if
\else
\fi
```

`\newif`는 새로운 조건 변수를 설정하며, 설정된 조건 변수가 `foo`였으며, `\footrue`, `\foofalse`, `\iffoo`도 함께 정의된다. `\iftrue`와 `\iffalse`는 각각 항상 참, 거짓인 조건이다. `\if`, `\else`, 그리고 `\fi`는 조건문의 구성에 사용된다.

```
\ifcase<숫자><분기 0>
\or ...
\else<다른 분기들>
\fi
```

`\ifcase`는 조건 분기문에서 사용된다.

### 4 길이 변수 및 상수 조작

```
\newlength{\변수이름}
\setlength{\변수이름}{길이}
\addtolength{\변수이름}{길이}
\settowidth{\변수이름}{텍스트}
```

`\newlength`는 새로운 길이 변수를 만든다. `\setlength`는 길이 변수에 값을 할당한다. 할당값은 `5mm plus 1mm minus .5mm`와 같이 유동적인 값으로 지정할 수도 있다. 길이 변수의 현재 값에 지정된 값을 더한다. 지정된 값이 음수면 기존 길이 변수에서 지정된 만큼 값을 뺀다. 길이 변수에 임의의 텍스트의 폭만큼의 값을 지정하려면 `\settowidth`를 사용한다.

```
\newdimen\변수이름
\ifdim\변수이름1<비교연산자\변수이름2
```

`\newdimen`은 산술 연산이 가능한 길이 변수를 만든다. `\ifdim`은 두 변수를 비교하는데 이용된다. 비교 연산자로는 `>`, `<`, 혹은 `=`를 쓸 수 있다.

### 5 카운터 조작

```
\newcount\카운터이름
\newcounter{카운터이름}[부모카운터이름]
\setcounter{카운터이름}{정수}
\stepcounter{카운터이름}{정수}
\addtocounter{카운터}{정수}
```

`\newcount`는 새로운 카운터 변수를 정의한다. `\newcounter`도 마찬가지로의 동작을 한다. 이 때 새로 만드는 카운터가 종속되는 부모 카운터의 이름을 지정할 수 있다. `\setcounter`는 카운터의 값을 주어진 정수로 지정한다. `\stepcounter`는 카운터의 값을 1 증가시킨다. 이 때 이에 종속된 카운터들을 값이 모두 0이 된다. `\addcounter`는 카운터에 주어진 정수만큼 더한다.

## 6 박스 조작

```
\newsavebox{\박스변수이름}
\savebox{\박스변수이름}[폭][위치]{박스내용}
\sbox{\박스변수이름}{박스내용}
\usebox{\박스변수이름}
\parbox[위치][높이][내부위치]{폭}{텍스트}
```

`\newsavebox`는 새로운 박스 변수를 정의한다. `\savebox`는 주어진 박스 변수에 박스로 만들 내용을 할당한다. 이 때 폭과 박스 내용의 수평 위치를 선택적으로 지정할 수 있다. 위치는 l(왼쪽 정렬), r(오른쪽 정렬), c(가운데 정렬), s(배분 정렬) 가운데에서 지정할 수 있다. `\sbox`는 `\savebox`의 줄임말로 선택 사항은 지정할 수 없다. `\usebox`는 `\savebox`에 의해 지정된 박스를 표시하는데 사용한다. `\parbox`는 하나의 문단 전체를 박스로 만든다. 위치는 베이스라인(baseline)을 기준으로 한 박스의 수직 위치, 내부 위치는 박스 내의 텍스트의 수직 위치를 지정한다. 위치 지시는 t(박스의 첫 줄이 베이스라인에 정렬), b(박스의 마지막 줄이 베이스라인에 정렬), c(베이스라인을 기준으로 세로로 가운데에 정렬), s(c와 비슷하게 세로로 배분 정렬) 가운데 하나로 할 수 있다. 내부 위치의 경우에는 베이스라인이 박스의 밑변이다.

```
\begin{minipage}[위치][높이][내부위치]{폭}
  텍스트
\end{minipage}
```

minipage 환경은 `\parbox`와 비슷하게 동작하여 하나의 작은 페이지처럼 다룰 수 있는 박스를 만든다. 따라서 minipage에는 각주를 달 수 있다.

```
\hbox{박스내용}
\vbox{박스내용}
\vtop{박스내용}
```

`\hbox`는 박스 내용의 폭만큼 수평 방향으로 박스를 만든다. 높이는 고정된다. `\vbox`는 가로는 고정된 상태에서 세로로 위쪽으로 길어지는 박스를 만든다. 박스 내용 텍스트는 길어지면 줄바꿈이 일어난다. 그러나 페이지 나눔을 일어나지 않는다. `\vtop`은 `\vbox`와 비슷하지만 베이스라인을 기준으로 아래쪽으로 박스가 길어진다.

## 7 행 / 페이지 나눔 조작

```
\linebreak[숫자]
\nolinebreak[숫자]
\pagebreak[숫자]
\nopagebreak[숫자]
```

`\linebreak`는 명령이 주어진 위치에서 행 나눔을 한다. `\nolinebreak`는 명령의 주어진 위치에서의 행 나눔을 억제한다. 선택적으로 주어지는 숫자는 0에서 4사이의 숫자로 동작의 강도를 지정한다. 4가 가장 강도가 크다. 이 값이 주어지지 않을 때의 기본값은 4이다. `\pagebreak`와 `\nopagebreak`는 페이지 나눔을 조작하며 사용법은 행 나눔 명령과 같다.

## 8 기타

```
\def\명령어이름{명령어정의}
\def\명령어이름#인자개수{명령어정의}
```

`\newcommand`와 비슷하게 동작하여 새 명령어를 정의한다.

```
\csname단어들\endcsname
```

단어들을 문자로만 이루어진 단어가 될 때까지 계속 확장한 다음, 그 문자 단어(토큰)로 제어 문자열(control sequence)을 만든다. 즉, 명령어 이름을 만든다.

```
\makeatletter
\makeatother
```

문자 는 `.tex` 파일 내에서는 명령어의 일부로 쓰일 수 없고, 스타일 파일(`.sty`)과 클래스 파일(`.cls`)에서만 명령어의 일부로 쓰일 수 있다. `.tex` 파일에서 이 문자를 명령어의 일부로 사용하려면 `\makeatletter`를 이용해야 한다. 이 명령의 효과를 마감하는 명령어는 `\makeatother`이다.

```
\usepackage [패키지선택사항]{패키지이름}
```

사용할 스타일 (패키지)을 지정하는데 이용한다.