

# How To L<sup>A</sup>T<sub>E</sub>X

## 1강 - L<sup>A</sup>T<sub>E</sub>X 기초

---

201811206 황인택

March 28, 2023

- $\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 이 무엇인지 안다.
- 텍으로 과제를 만들어서 제출할 수 있다.

한 강의당 1시간-1시간 30분 정도의 분량으로 진행될 예정이다. 각 강의의 주제는 다음과 같다.

**1강** L<sup>A</sup>T<sub>E</sub>X 기초

**2강** 수식, 그림, 그리고 표

**3강** L<sup>A</sup>T<sub>E</sub>X 프로그래밍

# $\text{T}_{\text{E}}\text{X}$ 이 뭔가요?

$\text{T}_{\text{E}}\text{X}$ ,  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , pdfTeX... 어려운 말들이 너무 많다!

**$\text{T}_{\text{E}}\text{X}$**  Donald Knuth가 1978년에 만든 조판 시스템(언어 + 컴파일러).

**$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$**  Leslie Lamport가 1984년에 만든 조판 시스템.  $\text{T}_{\text{E}}\text{X}$  언어로 작성되어 있으며,  
 $\text{T}_{\text{E}}\text{X}$ 과도 당연히 호환된다.

**pdfTeX, XeTeX, LuaTeX**  $\text{T}_{\text{E}}\text{X}$  언어의 구현체들.

# T<sub>E</sub>X을 왜 쓰나요??

T<sub>E</sub>X의 장점...

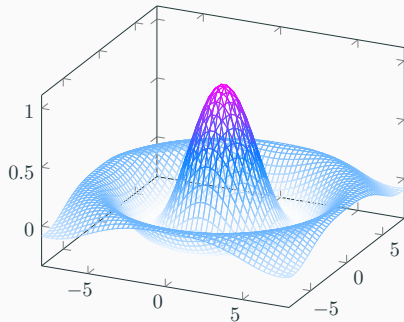
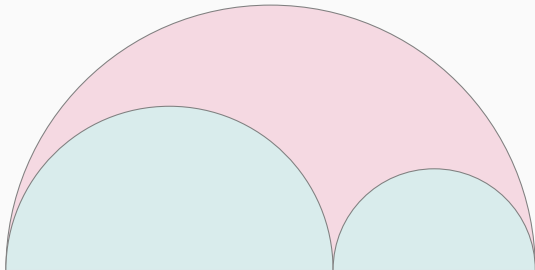
- 수식이 포함된 문서를 작성하는 데 최고다.
- 문서의 관리가 편하다.
- 내용과 스타일을 분리할 수 있다.
- 간지난다.

그럼에도 불구하고, T<sub>E</sub>X의 단점...

- 배우기 어렵다.

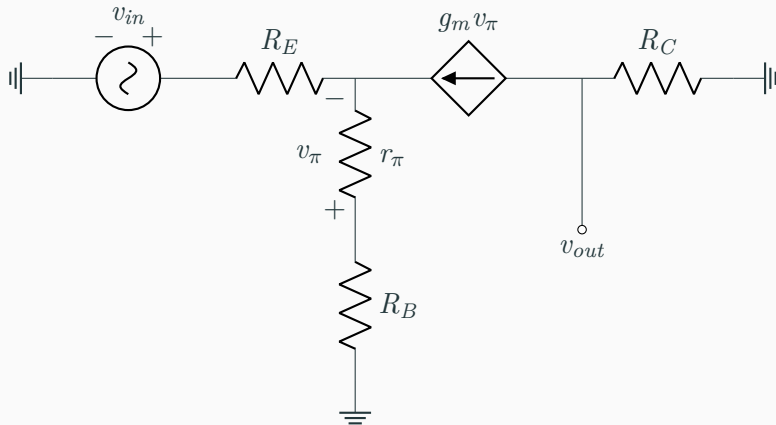
# $\text{T}_\text{E}\text{X}$ 으로 무엇을 할 수 있나요?

TikZ, pgfplots를 이용한 그림과 플로팅...



## TEX으로 무엇을 할 수 있나요?

다이어그램...



# T<sub>E</sub>X으로 무엇을 할 수 있나요?

Syntax Highlighting...

```
package main
```

```
import "fmt"
```

```
func main() {  
    fmt.Println("Hello, World!")  
}
```



따라서 한 번 쳐보자.

```
\documentclass{oblivoir}
```

```
\usepackage{kotex}
```

```
\title{My First Document}
```

```
\author{[당신의 이름]}
```

```
\begin{document}
```

```
  \maketitle
```

```
  Hello, World! 안녕, 세계!  $a^2 + b^2 = c^2$ 
```

```
   $E = mc^2$ 
```

```
\end{document}
```

**\begin {document}** 이전까지의 부분은 **preamble**이라고 부른다. preamble에는 보통 다음의 명령어들이 온다.

- **\documentclass**: 이 문서의 클래스를 결정한다. 클래스에 따라서 문서의 모양과 형식이 달라진다. 유명한 클래스로는 article, book, beamer, 그리고 앞으로 사용할 oblivoir 등이 있다.
- **\usepackage**: 사용할 패키지들을 임포트한다. 텍에서의 패키지는 다른 언어의 패키지들과 유사하다.
- 기타 매크로 및 설정 (자세한 것은 3강에서 다룬다.)

L<sup>A</sup>T<sub>E</sub>X에서는 이어지는 화이트스페이스를 전부 하나의 빈 칸으로 대체한다.

문단을 나눌 때는 반드시 한 줄을 비워놓는다. 화이트스페이스와 마찬가지로, 여러 줄을 비워놓아도 한 줄을 비운 것으로 인식한다.

```
So, _ _ so _ you _ think  
you _ can _ _ _ tell
```

```
Heaven _ _ from _ hell?
```

```
Blue _ skies _ from _ _ pain?
```

```
So, so you think you can tell  
Heaven from hell?  
Blue skies from pain?
```

다음의 문자들은 특수한 용도가 있으므로 일반적으로 입력할 수 없다.

# \$ % ^ & \_ { } ~ \

입력하고 싶다면, 앞에 \를 붙이면 된다. (\#, \\$, ...)

\는 \textbackslash로 입력한다.

따옴표는 왼쪽 따옴표와 오른쪽 따옴표가 구분된다. 왼쪽은 ` , 오른쪽은 ' 이다. 비교해보자.

'Hello, World!'	'Hello, World!'
`Hello, World!'	‘Hello, World!’
``Hello, World!''	“Hello, World!”

문서 구조는 크게 다음처럼 나눌 수 있다.

- part
- chapter
- section
- subsection
- paragraph
- subparagraph

보통 chapter, section, paragraph 정도만 본다.

# L<sup>A</sup>T<sub>E</sub>X의 명령어

L<sup>A</sup>T<sub>E</sub>X은 프로그래밍 언어이므로, 명령어의 개념이 있다. L<sup>A</sup>T<sub>E</sub>X의 명령어는 command와 environment로 분류할 수 있다.

command는 다음과 같은 형식을 가진다.

**\command**[option1, option2, ...]{arg1}{arg2}...

argument가 한 글자일 경우, 대괄호를 생략할 수 있다. 이 경우 토큰을 하나만 먹는다.

<b>\sqrt</b>	{2}	$\sqrt{2}$
<hr/>		
<b>\sqrt</b>	2	$\sqrt{2}$
<hr/>		
<b>\sqrt</b>	23	$\sqrt{23}$

environment는 다음과 같은 형식을 가진다.

```
\begin{environment}
```

...

```
\end{environment}
```

environment는 **\begin**과 **\end**로 둘러싸인 구간에서 작동한다.



다음을 쳐 보자.

```
\begin{center}  
  \textbf{Centered Text}:  $\sqrt{2}$ ,  $\sqrt[3]{2}$   
\end{center}
```

다음과 같이 텍스트에 효과를 넣을 수 있다.

<code>\textbf</code>	<code>{text}</code>	<b>text</b>
----------------------	---------------------	-------------

---

<code>\textit</code>	<code>{text}</code>	<i>text</i>
----------------------	---------------------	-------------

---

<code>\textsl</code>	<code>{text}</code>	<i>text</i>
----------------------	---------------------	-------------

---

<code>\texttt</code>	<code>{text}</code>	text
----------------------	---------------------	------

다음과 같이 텍스트의 크기를 바꿀 수 있다.

<code>\tiny</code>	text	text
<code>\small</code>	text	text
<code>\normalsize</code>	text	text
<code>\Large</code>	text	text
<code>\Huge</code>	text	text

이 효과는 문서 전체에 적용된다. 이것이 싫다면 다시 `\normalsize` 명령어를 사용하거나, 대괄호로 묶거나, 아니면 environment 형태로 사용하자.

정렬을 변경하려면, `flushleft`, `flushright`, `center` 환경을 사용한다.

```
\begin{flushleft}
```

```
left aligned
```

```
\end{flushleft}
```

```
\begin{center}
```

```
center aligned
```

```
\end{center}
```

```
\begin{flushright}
```

```
right aligned
```

```
\end{flushright}
```

left aligned

center aligned

right aligned

글을 나열하는 방법으로는 `itemize`, `enumerate`, `description` 환경이 있다.

```
\begin{itemize}
```

```
  \item a
```

```
  \item b
```

```
\end{itemize}
```

- a

- b

```
\begin{enumerate}
```

```
  \item a
```

```
  \item b
```

```
\end{enumerate}
```

1. a

2. b

```
\begin{description}
```

```
  \item[n] a
```

```
  \item[k1] a
```

```
  \item[k2] b
```

```
\end{description}
```

k1 a

k2 b

## 폰트 바꾸기

폰트를 바꾸려면 fontspec 패키지를 로드한다. 이 패키지는 XeTeX과 LuaTeX에서만 사용할 수 있다.

```
\usepackage{fontspec}  
\setmainfont{serif font}  
\setsansfont{sans-serif font}  
\setmonofont{mono font}
```

oblivoir 클래스 혹은 koTeX을 사용 중이라면, 다음 명령어를 이용해 한글 폰트를 바꿀 수 있다.

```
\setmainhangulfont{serif font}  
\setsanshangulfont{sans-serif font}  
\setmonohangulfont{mono font}
```

oblivoir 클래스의 경우, fapapersize 패키지를 이용한다.

```
\usepackage{fapapersize}
```

```
\usefapapersize{*,*,30mm,30mm,30mm,30mm}
```

기타 클래스의 경우, geometry 패키지를 이용한다.

백문이 불여일견, 직접 문서를 작성해보자!



$\text{T}_{\text{E}}\text{X}$ 의 세계는 방대하기 때문에, 모든 것을 다 외운다는 생각은 버려야 한다.

아래의 자료들과 구글을 활용해서, 그때그때 문제를 해결하면서 작성하도록 하자.

- lshort-ko - 역사와 전통의  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  입문서이다.
- 권현우 교수님의 강의 - 유튜브에서 무료로 볼 수 있다.

Thank you for listening!