article 클래스를 파헤쳐보자

2008 년 공주대 워크숍 강좌팀

2008 년 5 월 24 일

요 약

TEX(텍)은 탁월한 조판 프로그래밍 언어입니다. 한글 텍 사용자 그룹은 텍과 그 파생 매크로 집합인 LATEX, ConTeXt, AMSLATEX등을 사용하여 논문을 작성하거나 프로그래밍하거나 책을 만들거나 하는 일을 하는 사용자들의 모임입니다. 텍에 대하여 궁금하시면 TEX 페이지를, LATEX에 대해서는 LATEX 페이지를 참조하십시오. 텍을 처음 사용하신다면 처음 시작하기를 읽어보시기 바랍니다. 이곳에서는 질답 게시판을 통하여 의견을 나눌 수 있고 위키나 기고 게시판을 통하여 유용한 지식을 다른 사람과 공유할 수 있습니다. Happy TEX'ing.

차 례

1 국어 수업 **3**

1.1 도널드 크누스 **3**

1.1.1 이름 3

1.1.2 생애와 업적 3

1.1.3 수상 경력 6

2 영어 수업 **8**

2.1 Donald Ervin Knuth **8**

2.2 Education and academic work **8**

2.2.1 수상 경력 10

3 수학 수업 **13**

3.1 간단한 수식의 표현 **13**

3.2 복잡한 수식의 표현 **15**

찾아보기 **18**

제 1 장

국어 수업

1.1 도널드 크누스

도널드 어빈 크누스(Donald Ervin Knuth, 1938년 1월 10일 위스콘신 주 밀워키 출생)는 저명한 컴퓨터 과학자이며, 현재 스탠퍼드 대학교의 명예교수이다.

1.1.1 이름

본인의 홈페이지 FAQ에 의하면 그의 성을 커누스(KaNOOTH)라고 읽는다. 한자로 된 이름 高德纳도 있다.

1.1.2 생애와 업적

크누스는 컴퓨터 과학 분야에서 가장 권위있는 책인 *The Art of Computer Programming*의 저자로 가장 널리 알려져 있다. 이 책의 1권은 크누스가 28세일 때 지은 것이다. 알고리즘 분석 분야를 실질적으로 창조했으며, 이론 컴퓨터 과학의 여러 분야에서 기초적인 중요한 공헌을 했다. 조판 시스템과 메타폰트 폰트 디자인 시스템을 만들었으며 문학적 프로그래밍(literate programming)의 개념을 처음으로 주장했다.

그는 수준 높은 프로그래머였으며, geek 유머로 유명하다. 예를 들면 그가 쓴 책에서 오타나 실수를 찾아내면 상금으로 2.56 달러를 주는데, 왜냐하면 "256 센트가 16 진수로 1 달러이기 때문"이다. (예외도 있는데 *3:16 Bible Texts Illuminated* 라는 책에서 잘못된 부분을 찾는 사람에게 걸린 현상금은 3.16 달러이다). 소프트웨어의 버전번호는 3, 3.1, 3.14 와 같은 방식로 π 로 수렴한다. METAFONT 의 버전 번호는 같은 방식으로 e 에 수렴한다.

학부는 Case Institute of Technology(현재 Case Western Reserve University)에서 수학을 전공했고, 1963 년에 캘리포니아 공대에서 수학 박사학위를 받았다. 1968 년에 스탠퍼드 대학교의 교수가 되었다. 튜링상, 미국 국가 과학 훈장등 여러 상을 수상했다.

취미 생활로는 음악을 즐기며 특히 오르간 연주를 즐긴다고 한다. (실제로 크누스의 집에 파이프 오르간이 있다고 한다.) 다만 연주실력이 썩 훌륭하지는 않다고 한다. 크누스는 또한 이메일을 사용하지 않는 것으로 유명하다. 약 1975 년부터 1990 년 1 월 1 일까지만 이메일을 사용했다고 하는데 그 이후로는 그 때 사용해본 것만으로도 충분하다면서 평생 동안 이메일을 사용하지 않고 일반 우편만 사용한다고 한다. 크누스는 우편물을 "배치 모드(batch mode)"로 처리하는 것이 더 효율적이라는 것을 깨달았다고 한다. (예를 들어 3 개월동안 받은 우편물을 하루 만에 처리하는 방식을 말한다.)

질 크누스 여사와 결혼했으며 슬하에 두 자녀가 있다.

그의 업적에 대한 내용은 《컴퓨터를 만든 15 인의 과학자Out of their minds》(데이스 샤사 지음. 세종연구원 출판. 1998 년) 에서 한 장(章, chapter) 에 걸쳐서 설명하고 있다. 책에서 설명하고 있는 크누스에 대한 유명한 일화를 소개한다.

다음 문장은 quotation 환경으로 조판된 것이다.

> 내(앨런 케이)가 스탠퍼드에서 AI 프로젝트를 맡고 있던 시절 (1960 년대 후반) 매년 추수감사절 때마다 가졌던 행사 가운데 하나는 그 샌프란시스코 만안 지방(Bay area) 에서 연구 프로젝트에 종사하고 있는 사람들과 프로그래밍 경연대회를 여는 것이었습니다. 상품은 칠면조였던 것으로 기억됩니다.
>
> 문제는 주로 매커시가 출제하였지요. 한 해는 크누스가 그 대회

> 에 참가해서 가장 빠른 시간에 프로그래밍을 실행시킨 사람에게 주는 상과 알고리즘을 가장 빨리 실행하는 사람에게 주는 상을 모두 휩쓸었습니다. 그는 윌버(Wilbur) 시스템이라는 원격 일괄 시스템을 이용해 최악의 시스템에서 그것을 해 내었지요. 그는 간단하게 모든 참가자들을 바보로 만들어 버렸습니다.
>
> 사람들이 그에게 이렇게 물었지요. "어떻게 이런 일이 가능하죠?" 그의 대답은 이런 것이었습니다. "내가 프로그래밍을 배울 때는 하루에 기계와 5분을 보낼 수 있으면 꽤 운이 좋은 경우였죠. 프로그램이 작동하도록 만들고 싶으면 프로그램을 정말로 제대로 작성해야 했습니다. 그래서 사람들은 돌에 글을 새기듯이 프로그래밍을 배웠습니다. 그냥 그것을 향해 조금씩 다가가기만 하면 됩니다. 난 그렇게 프로그래밍을 익혔습니다."

다음 문장은 quote 환경으로 조판된 것이다.

> 내(앨런 케이)가 스탠퍼드에서 AI 프로젝트를 맡고 있던 시절 (1960년대 후반) 매년 추수감사절 때마다 가졌던 행사 가운데 하나는 그 샌프란시스코 만안 지방(Bay area)에서 연구 프로젝트에 종사하고 있는 사람들과 프로그래밍 경연대회를 여는 것이었습니다. 상품은 칠면조였던 것으로 기억됩니다.
>
> 문제는 주로 매커시가 출제하였지요. 한 해는 크누스가 그 대회에 참가해서 가장 빠른 시간에 프로그래밍을 실행시킨 사람에게 주는 상과 알고리즘을 가장 빨리 실행하는 사람에게 주는 상을 모두 휩쓸었습니다. 그는 윌버(Wilbur) 시스템이라는 원격 일괄 시스템을 이용해 최악의 시스템에서 그것을 해 내었지요. 그는 간단하게 모든 참가자들을 바보로 만들어 버렸습니다.
>
> 사람들이 그에게 이렇게 물었지요. "어떻게 이런 일이 가능하죠?" 그의 대답은 이런 것이었습니다. "내가 프로그래밍을 배울 때는 하루에 기계와 5분을 보낼 수 있으면 꽤 운이 좋은 경우였죠. 프로그램이 작동하도록 만들고 싶으면 프로그램을 정말로 제대로 작성해야 했습니다. 그래서 사람들은 돌에 글을 새기듯이 프로그래밍을 배웠습니다. 그냥 그것을 향해 조금씩 다가가기만 하면 됩니다. 난 그렇게 프로그래밍을 익혔습니다."

1.1.3 수상 경력

표 1: itemize 환경

- 1971 년 제 1 회 ACM 그레이스 머레이 호퍼 상
- 1974 년 튜링 상
- 1979 년 National Medal of Science
- 1995 년 존 폰 노이만 메달
- 1995 년 이스라엘의 기술연구소 Technion 으로부터 하비 상
- 1996 년 교토 상

표 2: tabbing 환경

1971 년 제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년 튜링 상
1979 년 National Medal of Science
1995 년 존 폰 노이만 메달
1995 년 이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년 교토 상

표 3: tabular* 환경

1971 년	제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년	튜링 상
1979 년	National Medal of Science
1995 년	존 폰 노이만 메달
1995 년	이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년	교토 상

표 4: tabularx 환경

1971 년	제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년	튜링 상
1979 년	National Medal of Science
1995 년	존 폰 노이만 메달
1995 년	이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년	교토 상

표 5: tabulary 환경

1971 년	제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년	튜링 상
1979 년	National Medal of Science
1995 년	존 폰 노이만 메달
1995 년	이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년	교토 상

제 2 장

영어 수업

2.1 Donald Ervin Knuth

Donald Ervin Knuth (pronounced [kəˈnuːθ])[1] (b. 10 January 1938) is a renowned computer scientist and Professor Emeritus of the *Art of Computer Programming* at Stanford University.

Author of the seminal multi-volume work *The Art of Computer Programming* ("TAOCP"), Knuth has been called the "father" of the analysis of algorithms, contributing to the development of, and systematizing formal mathematical techniques for, the rigorous analysis of the computational complexity of algorithms, and in the process popularizing asymptotic notation.

In addition to fundamental contributions in several branches of theoretical computer science, Knuth is also the creator of the TEX computer typesetting system, the related METAFONT font definition language and rendering system, and the Computer Modern family of typefaces.

A prolific writer and scholar, Knuth is also creator of the WEB/CWEB computer programming systems designed to encourage and facilitate literate programming, as well as designer of the MMIX instruction set architecture.

2.2 Education and academic work

Knuth was born in Milwaukee, Wisconsin, where his father owned a small printing business and taught bookkeeping at Milwaukee Lutheran High School,

[1] http://www-cs-faculty.stanford.edu/~knuth/faq.html

which he attended. He was an excellent student, earning achievement awards. He applied his intelligence in unconventional ways, winning a contest when he was in eighth grade by finding over 4,500 words that could be formed from the letters in "Ziegler's Giant Bar." This won him a television set for his school and a candy bar for everyone in his class.

Knuth had a difficult time choosing physics over music as his major at Case Institute of Technology (now part of Case Western Reserve University). He then switched from physics to mathematics, and in 1960 he received his bachelor of science degree, simultaneously receiving his master of science degree by a special award of the faculty who considered his work outstanding. At Case, he managed the basketball team and applied his talents by constructing a formula for the value of each player. This novel approach was covered by Newsweek and by Walter Cronkite on the CBS television network.

While doing graduate studies, Knuth worked as a consultant, writing compilers for different computers. In 1963, he earned a Ph.D. in mathematics (advisor: Marshall Hall) from the California Institute of Technology, where he became a professor and began work on *The Art of Computer Programming*, originally planned to be a single book, and then planned as a six, and then seven-volume series. In 1968, he published the first volume. That same year, he joined the faculty of Stanford University.

In 1971, Knuth was the recipient of the first ACM Grace Murray Hopper Award. He has received various other awards including the Turing Award, the National Medal of Science, the John von Neumann Medal and the Kyoto Prize. After producing the third volume of his series in 1976, he expressed such frustration with the nascent state of the then newly developed electronic publishing tools (esp. those which provided input to phototypesetters) that he took time out to work on typesetting and created the TEX and METAFONT tools.

In recognition of Knuth's contributions to the field of computer science, in 1990 he was awarded the singular academic title of Professor of *The Art of Computer Programming*, which has since been revised to Professor Emeritus of *The Art of Computer Programming*.

In 1992 he became an associate of the French Academy of Sciences. Also that year, he retired from regular research and teaching at Stanford University in order to finish The Art of Computer Programming. In 2003 he was elected as a foreign member of the Royal Society. As of 2004, the first three volumes of his series have been re-issued, and Knuth is currently working on volume four, excerpts of which are released periodically on his website. Meanwhile, Knuth gives informal lectures a few times a year at Stanford University, which he calls Computer Musings. He is also a visiting professor at the Oxford University

Computing Laboratory in the United Kingdom.

In addition to his writings on computer science, Knuth, a devout Lutheran, is also the author of *3:16 Bible Texts Illuminated* (1991), ISBN 0-89579-252-4, in which he attempts to examine the Bible by a process of stratified sampling, namely an analysis of chapter 3, verse 16 of each book. Each verse is accompanied by a rendering in calligraphic art, contributed by a group of calligraphers under the leadership of Hermann Zapf.

He is also the author of *Surreal Numbers* (1974) ISBN 0-201-03812-9, a mathematical novelette on John Conway's set theory construction of an alternate system of numbers. Instead of simply explaining the subject, the book seeks to show the development of the mathematics. Knuth wanted the book to prepare students for doing original, creative research.

On January 1, 1990, Knuth announced to his colleagues that he would no longer have an email address, so that he may concentrate on his work. Knuth is a fan of Wikipedia, but he's a bit leery of the concept, saying that he would not want to have to remain forever on guard after making technically complex contributions, lest his comments be badly reedited.

In 2006, Knuth was diagnosed with prostate cancer. He underwent surgery in December that year and started "a little bit of radiation therapy as a precaution but the prognosis looks pretty good," as he reported in his video autobiography.

2.2.1 수상 경력

표 6: itemize 환경

- 1971 년 제 1 회 ACM 그레이스 머레이 호퍼 상
- 1974 년 튜링 상
- 1979 년 National Medal of Science
- 1995 년 존 폰 노이만 메달
- 1995 년 이스라엘의 기술연구소 Technion 으로부터 하비 상
- 1996 년 교토 상

표 7: tabbing 환경

1971 년 제 1 회 ACM 그레이스 머레이 호퍼 상

1974 년 튜링 상
1979 년 National Medal of Science
1995 년 존 폰 노이만 메달
1995 년 이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년 교토 상

표 8: tabular* 환경

1971 년	제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년	튜링 상
1979 년	National Medal of Science
1995 년	존 폰 노이만 메달
1995 년	이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년	교토 상

표 9: tabularx 환경

1971 년	제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년	튜링 상
1979 년	National Medal of Science
1995 년	존 폰 노이만 메달
1995 년	이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년	교토 상

표 10: tabulary 환경

1971 년	제 1 회 ACM 그레이스 머레이 호퍼 상
1974 년	튜링 상
1979 년	National Medal of Science
1995 년	존 폰 노이만 메달
1995 년	이스라엘의 기술연구소 Technion 으로부터 하비 상
1996 년	교토 상

제 3 장

수학 수업

한국과학기술원산업수학연구실 · 고기형 지음 (1995,1998), 《한글 LaTeX의 모든 것—한글과 TeX》(청문각, 제 1 판)의 각 장별 chapterprecis들과 연습문제를 모은 것이다. 이 책은 현재 절판되어 구할 수 없게 되어 있지만 장별 개요와 연습문제만으로도 책의 내용을 짐작할 수 있을 것이다. 이 문서는 Hangul-ucs와 memoir/memhangul-ucs로 작성하는 것으로 원래의 책이 만들어졌던 hLaTeXp와는 전혀 다른 환경을 쓰고 있지만 내용은 거의 동일하게 보이도록 하였으며 수식은 $\mathcal{AMS}$-LaTeX 형식으로 구현하였다. 이 문서는 김강수와 조인성이 함께 작성하였다.

3.1 간단한 수식의 표현

TeX은 원래부터 수식을 잘 처리하도록 만들어졌다. 수식 처리는 TeX의 가장 큰 장점의 하나로 다른 프로그램들에 비해 월등한 능력을 발휘한다. 이 장에서는 간단한 수식을 표현하는 예들을 보이며 배열, 여러 줄에 걸친 수식 등 좀더 복잡한 수식은 3.2 절에서 다시 자세히 다룬다.

[1] 다음 수식을 만들어보시오.

$$\sum_{n=1}^{\infty} \frac{b_n}{(z-z_0)^n}$$

[2] 다음 수식을 만들어보시오.

$$\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right) \geq \left(\sum_{i=1}^{n} x_i y_i\right)^2$$

[3] 다음 수식을 만들어보시오.

$$u(\rho e^{i\phi}) = \frac{(r^2-\rho^2)}{2\pi} \int_0^{2\pi} \frac{u(re^{i\theta})}{r^2 - 2r\rho\cos(\theta-\phi)+\rho^2}\, d\theta$$

[4] 다음 수식을 만들어보시오.

$$N(\alpha) = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_s^{\alpha_s}$$

[5] 다음 수식을 만들어보시오.

$$P(z) = \prod_{n=1}^{\infty} E_p\left(\frac{z}{a_n}\right)$$

[6] 다음 수식을 만들어보시오.

$$|E_T| \leq \frac{b-a}{12} h^2 \max|f''(x)| = \frac{1}{2}\left(\frac{1}{n}\right) \max\left|\frac{2}{x^3}\right|$$

[7] 다음 수식을 만들어보시오.

$$f(c) = \frac{1}{b-a}\int_a^b f(x)dx$$

[8] 다음 수식을 만들어보시오.

$$(x_1^2 + \cdots + x_n^2)(y_1^2 + \cdots + y_n^2) \geq (x_1y_1 + \cdots + x_ny_n)^2$$

[9] 다음 수식을 만들어보시오.

$$\left(\frac{a}{p}\right) \equiv a^{\frac{p-1}{2}} \pmod{p}$$

3.2 복잡한 수식의 표현

4장에서 간단한 수식을 다루는 것에 대해 알아보았다. 그러나, 실제로 논문을 쓰거나 책을 쓰는 경우 매우 복잡한 수식을 아름답게 표현해야 할 때가 있다. 이 장에서는 TEX의 가장 큰 장점인 수식을 다루는 것에 대해 자세히 알아본다.

[1] 다음 수식을 만들어 보시오.

$$\begin{aligned} c(n) &= \sum_{n\leq x} a_n - 2\int_1^{[x]} \log t\, dt \\ &= 2x\log x - 2[x]\log[x] - 2 + O(x) \\ &= O(x) \end{aligned}$$

[2] 다음 수식을 만들어 보시오.

$$x_1 \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix} + \cdots + x_n x_n \begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix} = 0$$

[3] 다음 수식을 만들어 보시오.

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{pmatrix}$$

[4] 다음 수식을 만들어 보시오.

$$A(BC) = \begin{pmatrix} 2 & 1 & 5 \\ 1 & 3 & 2 \end{pmatrix} \begin{pmatrix} -1 & 5 \\ -3 & -5 \\ 1 & 5 \end{pmatrix} = \begin{pmatrix} 0 & 30 \\ -8 & 0 \end{pmatrix}$$

[5] 다음 수식을 만들어 보시오.

$$\begin{array}{ccccccc} A^1 & = & b_{11}X^1 & + & \cdots & + & b_{n1}X^n \\ \vdots & & \vdots & & & & \vdots \\ A^n & = & b_{1n}X^1 & + & \cdots & + & b_{nn}X^n \end{array}$$

[6] 다음 수식을 만들어 보시오.

$$f(x) = \begin{cases} x^2 + 1, & x \leq 0 \\ x^4 - 3x^2 - 7x + 1, & 0 \geq x \geq 1 \\ \frac{2x}{x^2+1}, & x \geq 1 \end{cases}$$

[7] 다음 수식을 만들어 보시오.

$$\left(\begin{array}{c} \left[\begin{array}{cc} x & y \\ z & w \end{array} \right] \\ a \\ b \end{array} \right)$$

[8] |align*| 환경을 사용하여 다음 수식을 만들어 보시오.

$$\begin{aligned} a + b + c \\ b + c + d \\ c + d + e \\ a + 2b + 3c + 2d + e \end{aligned}$$

[9] |alignat*| 환경을 이용하여 다음 수식을 만들어 보시오.

$$\begin{aligned} F_i &= x_i f_i - g_i & G_j &= y_j g_j - f_i & H_k &= L_k + 1 \\ f_i &= F_{i-1} + G_{i-1} & g_j &= F_{j-1} - G_{j-1} & L_k &= \{H_{k-1}\}^2 \end{aligned}$$

찾아보기

책 이름
3:16 Bible Texts Illuminated, 3, 9
Art of Computer Programming, 8
Out of their minds, 4
Surreal Numbers, 9
The Art of Computer Programming, 3, 8, 9
컴퓨터를 만든 15 인의 과학자, 4
한글 LaTeX의 모든 것—한글과 TeX, 12