VIM Book

내용

[1 Basic Editing 2](#_Toc178180315)

[시작하기 전에 2](#_Toc178180316)

[Running Vim for the First Time 2](#_Toc178180317)

[Modes 2](#_Toc178180318)

[Editing for the First Time 3](#_Toc178180319)

[Moving Around 3](#_Toc178180320)

[Deleting Characters 3](#_Toc178180321)

[Undo and Redo 3](#_Toc178180322)

[Getting Out 3](#_Toc178180323)

[Discarding Changes 3](#_Toc178180324)

[Other Editing Commands 3](#_Toc178180325)

[Inserting Characters at the End of a Line 3](#_Toc178180326)

[Deleting a Line 3](#_Toc178180327)

[Opening Up New Lines 3](#_Toc178180328)

[Help 4](#_Toc178180329)

[Other Ways to Get Help 6](#_Toc178180330)

[Using a Count to Edit Faster 6](#_Toc178180331)

[The Vim Tutorial 6](#_Toc178180332)

[2 Editing a Little Faster 6](#_Toc178180333)

[A. Installing Vim 7](#_Toc178180334)

[UNIX 7](#_Toc178180335)

# 1 Basic Editing

이 장에서는 편집을 시작하는 데 필요한 10가지 기본 Vim 명령을 알려줍니다. 이 장에서는 다음을 배웁니다.

* 네 가지 기본 이동 명령
* 텍스트를 삽입하고 삭제하는 방법
* 도움말을 얻는 방법(매우 중요)
* 편집기 종료

## 시작하기 전에

Vim을 설치하지 않은 경우 부록 A, "Vim 설치"를 읽고 편집기를 설치해야 합니다.

UNIX에서 실행 중이라면 다음 명령을 실행하세요.

$ touch ~/.vimrc  
~/.vimrc를 생성하면 Vim 모드에서 사용하고 싶다고 Vim에 알립니다. 이 파일이 없으면 Vim은 Vi 호환 모드에서 실행되고 많은 고급 Vim 기능에 액세스할 수 없게 됩니다. 그러나 언제든지 다음 명령을 사용하여 Vim 내에서 고급 기능을 활성화할 수 있습니다. :set nocompatible<Enter>.

## Running Vim for the First Time

Vim을 시작하려면 다음 명령을 입력하세요.

$ gvim file.txt

### Modes

Vim 편집기는 모달 편집기입니다. 즉, 편집기는 모드에 따라 다르게 동작합니다. 화면 하단에 파일 이름이 표시되거나 비어 있으면 일반 모드입니다. 삽입 모드에 있는 경우 표시기에 --INSERT--가 표시되고, 시각적 모드에 있는 경우 표시기에 --VISUAL--이 표시됩니다.

## Editing for the First Time

### Moving Around

h, j, k, l

### Deleting Characters

문자를 삭제하려면 커서를 해당 문자 위로 옮기고 x를 입력합니다. (이것은 옛날 타자기 시절로 돌아가서, xxxx를 입력해서 삭제하던 방식입니다.)

### Undo and Redo

너무 많은 내용을 삭제했다고 가정해 보겠습니다. 다시 입력할 수도 있지만 더 쉬운 방법이 있습니다. **u** 명령은 마지막 편집 내용을 취소합니다.

너무 많이 실행 취소하면 **CTRL-R**(다시 실행)을 눌러 이전 명령을 되돌릴 수 있습니다. 즉, 실행 취소를 취소합니다.

### Getting Out

종료하려면 **ZZ** 명령을 사용합니다. 이 명령은 파일을 쓰고 종료합니다.

### Discarding Changes

**:q!**

자세한 내용에 관심이 있는 분들을 위해, 이 명령의 세 부분은 콜론(:)으로, 명령 모드로 들어갑니다. q 명령은 편집기를 종료하라고 알려줍니다. override 명령 수정자(!)입니다.

## Other Editing Commands

### Inserting Characters at the End of a Line

**i** 명령은 커서 아래의 문자 앞에 문자를 삽입합니다.

커서 뒤에 텍스트를 삽입하는 작업은 **a** (append) 명령으로 수행합니다.

### Deleting a Line

줄을 삭제하려면 **dd** 명령을 사용합니다.

### Opening Up New Lines

새로운 줄을 추가하려면 **o** 명령을 사용하여 커서 아래에 새 줄을 엽니다.그러면 편집기가 삽입 모드로 전환됩니다.

## Help

마지막으로, 또 하나의 중요한 명령이 있습니다. 바로 help 명령입니다. 도움말을 얻으려면 다음을 입력하세요.

**:help**

주제를 제공하지 않으면 :help는 일반 도움말 창을 표시합니다. Vim의 제작자는 도움말 시스템에 매우 영리한(또는 매우 게으른) 일을 했습니다. 그들은 help 창을 일반 편집 창으로 만들었습니다. 모든 일반 Vim 명령을 사용하여 도움말 정보를 이동할 수 있습니다. 따라서 h, k, j, l은 왼쪽, 위, 아래, 오른쪽 등으로 이동합니다.

도움말 시스템을 종료하려면 편집기를 종료하는 데 사용하는 것과 동일한 명령인 ZZ를 사용하세요.

도움말 텍스트를 읽으면 세로 막대로 둘러싸인 텍스트가 보일 것입니다(예: |:help|). 이는 하이퍼링크를 나타냅니다. 커서를 막대 사이 아무 데나 놓고 CTRL+](태그로 이동)를 누르면 도움말 시스템이 지정된 주제로 이동합니다. (여기서는 설명하지 않은 이유로, 하이퍼링크에 대한 Vim 용어는 태그입니다. 따라서 CTRL+]는 커서 아래에 있는 단어가 지정한 태그 위치로 이동합니다.)

몇 번 점프한 후에는 돌아가고 싶을 수도 있습니다. CTRL+T(태그 팝)는 이전 화면으로 돌아갑니다. 또는 Vim 용어로 "태그 스택에서 태그를 팝합니다."

이 화면의 맨 위에는 \*help.txt\*라는 표기가 있습니다. 이것은 도움말 시스템에서 태그(하이퍼링크 대상)를 정의하는 데 사용됩니다. 7장, “프로그래머를 위한 명령”에서 태그에 대해 자세히 설명합니다.

주어진 주제에 대한 도움말을 얻으려면 다음 명령을 사용하세요.

**:help subject**

예를 들어, x 명령에 대한 도움말을 보려면 다음을 입력하세요.

:help x

텍스트를 삭제하는 방법을 알아보려면 다음 명령을 사용하세요.

:help deleting

사용 가능한 전체 인덱스를 보려면 다음 명령을 사용하세요.

:help index

제어 문자 명령(예: CTRL-A)에 대한 도움말이 필요할 때는 접두사 CTRL-를 붙여 입력해야 합니다.

:help CTRL-A

Vim 편집기에는 다양한 모드가 있습니다. 기본적으로 도움말 시스템은 일반 모드 명령을 표시합니다. 예를 들어, 다음 명령은 일반 모드 CTRL-H 명령에 대한 도움말을 표시합니다.

:help CTRL-H

다른 모드를 식별하려면 모드 접두사를 사용합니다.

이 명령의 삽입 모드 버전에 대한 도움말이 필요하면 키 앞에 i\_를 붙입니다. 그러면 다음 명령이 생성됩니다.

:help i\_CTRL-H

표 1.1에는 여러 가지 다른 모드 접두사가 나열되어 있습니다.

Vim 편집기를 시작하면 여러 명령줄 옵션을 사용할 수 있습니다. 이러한 옵션은 모두 대시(-)로 시작합니다. 예를 들어 -t 명령줄 옵션이 무엇을 하는지 알아보려면 명령을 사용합니다.

:help -t

Vim 편집기에는 편집기를 구성하고 사용자 정의할 수 있는 여러 옵션이 있습니다. 옵션에 대한 도움말이 필요한 경우 작은 따옴표로 묶어야 합니다. 예를 들어 숫자 옵션이 무엇을 하는지 알아보려면 다음 명령을 사용합니다.

:help ‘number’

다음 표는 특수 접두사를 요약한 것입니다.

텍스트, 폰트, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

특수 키는 꺾쇠 괄호로 묶습니다. 예를 들어 위쪽 화살표 키에 대한 도움말을 찾으려면 다음 명령을 사용합니다.

:help <Up>

부록 B, “<> 키 이름”에서는 키 이름의 전체 목록을 제공합니다.

### Other Ways to Get Help

<F1> 키를 눌러 도움말 화면으로 이동할 수 있습니다. 그러면 일반 도움말 화면이 표시되고, 거기에서 탐색할 수 있습니다. 키보드에 <Help> 키가 있으면 사용할 수도 있습니다.

## Using a Count to Edit Faster

예를 들어, 9줄 위로 이동하고 싶다면 kkkkkkkkk를 입력하거나 **9k** 명령을 입력하면 됩니다.

사실, 모든 이동 명령 앞에 숫자를 붙일 수 있습니다. 예를 들어, 이 장의 앞부분에서 a!!!<Esc>를 입력하여 줄 끝에 느낌표 세 개를 추가했습니다. 이를 수행하는 또 다른 방법은 **3a!<Esc>** 명령을 사용하는 것입니다. 3의 카운트는 a 명령에 뒤에 오는 것을 (!) 세 번 삽입하라고 말합니다.

마찬가지로, 세 개의 문자를 삭제하려면 **3x** 명령을 사용합니다.

## The Vim Tutorial

Vim 편집기의 UNIX 버전에는 대화형 튜토리얼이 제공됩니다. 1과에서는 이 장에서 설명하는 많은 명령을 다룹니다.

UNIX에서 튜토리얼을 호출하려면 다음 명령을 사용합니다.

**$ vimtutor**

# 2 Editing a Little Faster

## Word Movement

이동부터 시작해 보겠습니다. 커서를 한 단어 앞으로 이동하려면 **w** 명령을 사용합니다. **b** 명령은 한 단어 뒤로 이동합니다. 대부분의 Vim 명령과 마찬가지로 숫자 접두사를 사용하여 여러 단어를 지나 이동할 수 있습니다. 예를 들어, **4b**는 네 단어 뒤로 이동합니다.

## Moving to the Start or End of a Line

**$** 명령은 커서를 줄의 끝으로 옮깁니다. 사실, 여러 키가 "줄 끝" 명령에 매핑됩니다. 이러한 키에 대한 Vim 이름은 $, **<End>**, <kEnd>입니다. (<kEnd> 키는 Vim에서 키패드 End 키에 대한 이름입니다.)

$ 명령은 숫자 인수도 받습니다. 존재하는 경우 편집기가 다음 줄의 끝으로 이동합니다. 예를 들어, **1$**는 첫 번째 줄(현재 줄)의 끝으로 이동하고, **2$**는 다음 줄의 끝으로 이동합니다.

**^** 명령은 줄의 첫 번째 공백이 아닌 문자로 이동합니다. **<Home>** 또는 <Home> 키는 그림 2.3에서 볼 수 있듯이 줄의 첫 번째 문자로 이동합니다. (**0** [영] 명령도 동일한 작업을 수행합니다.)

## Searching Along a Single Line

이동은 가장 일반적인 편집 활동입니다. 가장 유용한 이동 명령 중 하나는 단일 문자 검색 명령입니다. 명령 **fx(전방 검색)**는 줄에서 단일 문자 x를 검색합니다.  
예를 들어, 다음 줄의 시작 부분에 있다고 가정해 보겠습니다.  
To err is human. To really foul up you need a computer.  
human의 h로 이동하고 싶다고 가정해 보겠습니다. **fh** 명령을 실행하면 커서가 h 위에 위치합니다.  
To err is human. To really foul up you need a computer.  
**5f<Space>**: 명령을 사용하여 5개 단어를 앞으로 스페이스할 수 있습니다.  
To err is human. To really foul up you need a computer.

**F** 명령은 왼쪽으로 검색합니다. 그림 2.4는 f 및 F 명령의 효과를 보여줍니다.

**tx(search ‘til)** 명령은 fx 명령과 비슷하게 작동하지만 지정된 문자 바로 앞에서 한 문자만 멈춥니다. 이 명령의 이전 버전은 **Tx**입니다.

때로는 검색을 시작하다가 잘못된 명령을 입력한 것을 깨닫는 경우가 있습니다. 예를 들어, 뒤로 검색하려면 f를 입력하지만 실제로는 F를 의미했다는 것을 깨닫습니다. 검색을 중단하려면 검색 키로 **<Esc>**를 누릅니다. 따라서 f<Esc>는 뒤로 검색을 중단한 것입니다. (참고: <Esc>는 검색뿐만 아니라 대부분의 작업을 취소합니다.)

## Moving to a Specific Line

# A. Installing Vim

www.vim.org 웹사이트에서 VIM을 다운로드할 수 있습니다. 이 사이트에는 Vim 소스는 물론 다양한 시스템에 맞춰 미리 컴파일된 바이너리도 들어 있습니다.

## UNIX

www.vim.org에서 다양한 UNIX 시스템에 대한 사전 컴파일된 바이너리를 얻을 수 있습니다. www.vim.org로 이동하여 "Vim 다운로드" 링크를 클릭한 다음 "바이너리 페이지" 링크를 따르세요.

$ sudo apt install vim-gtk3