From ed to vi and Vim

A concise historical reference with key milestones and design influences

Line editors before visual editing

Early interactive editing was shaped by teletypes and printing terminals, where repeatedly displaying a whole screen was slow or impossible. ed, written by Ken Thompson for Unix around 1969–1971, used concise commands to address lines, search, substitute and write files. Its command language reflected a world where every character printed had a cost.

The QED family of editors influenced ed. Many ed conventions survive in sed, regular expressions and vi's colon-command heritage. Exact release dates vary among histories because development and distribution often happened at different times.

```
ed file.txt
1,$p
%s/old/new/g
w
q
```

em, en, ex and vi

George Coulouris developed em (Editor for Mortals) in the mid-1970s as an early screen-oriented editing experiment. Bill Joy's en and ex work at Berkeley in the later 1970s developed the line editor into a more capable environment. vi emerged as the visual interface to ex, taking advantage of addressable video terminals.

vi is generally dated to 1976–1978, with Bill Joy as its principal original author. Its modal design allowed short editing commands and composable motions without requiring a dedicated modifier key for every action. The underlying ex command language remained accessible with a colon.

```
:set number
:%s/old/new/g
:wq
```

Standardisation and descendants

vi became a widely expected Unix editor, eventually specified by POSIX. Implementations varied across proprietary Unix systems, and the term vi could mean subtly different feature sets. Elvis, nvi and other reimplementations appeared as Unix-like systems spread to new platforms.

Bram Moolenaar released Vim (Vi IMproved) in 1991. Vim extended vi with multiple undo levels, syntax highlighting, scripting, splits, plugins and many other features while preserving a familiar command grammar. Neovim began in 2014 as a Vim-derived project focused on extensibility, asynchronous integration and modern interfaces.

```
vi --version
vim --version
nvim --version
```

Why the command grammar persists

A key idea is composability: operators such as d, c and y act on motions or text objects. The same movement knowledge applies to deleting, changing and yanking. Counts extend operations: 3dw removes three words, c$ changes to the end of a line, and d} deletes through a paragraph motion.

The historical interface may look cryptic, but the grammar reduces the number of independent commands a user must memorise. Vim and Neovim add capabilities without requiring abandonment of this model.

```
cw
3dw
c$
d}
ci"
```

Chronology at a glance

Approximate milestones: QED lineage, 1960s; ed, around 1969–1971, Ken Thompson; em, mid-1970s, George Coulouris; en/ex, mid-to-late 1970s, Bill Joy and Berkeley collaborators; vi, around 1976–1978, Bill Joy; Vim, 1991, Bram Moolenaar; Neovim, 2014, community project. Dates are deliberately approximate where projects evolved over several releases.

Useful primary references include the Unix Programmer's Manual, Berkeley Unix history, POSIX vi and ex specifications, and the Vim help documentation.

```
man ed
man vi
vim -c help
```

Reference scope: commands and defaults vary by distribution and software version. Consult the locally installed manual pages for exact syntax and behaviour.