Smol Launch

```
 ____                     _     _                                 _
/ ___|  _ __ ___    ___  | |   | |     __ _  _   _  _ __    ___ | |__
\___ \ | '_ ` _ \  / _ \ | |   | |    / _` || | | || '_ \  / __|| '_ \
 ___) || | | | | || (_) || |   | |___| (_| || |_| || | | || (__ | | | |
|____/ |_| |_| |_| \___/ |_|   |_____|\__,_| \__,_||_| |_| \___||_| |_|
```

Brutalist News

Insideparadeplatz

My News Links

Quorn mercury

CNN rt

CSM rt

NPR rt

Project

Project

Soil-less Microgreens

The Rise, Fall, and Return of Text-Based Computing

The Rise, Fall, and Return of Text-Based Computing

The early days of personal computing were centered around text-based interfaces like the command line terminal. Operating systems like UNIX provided users with powerful but cryptic commands for manipulating files and running programs. Communication was conducted on text-only platforms like Usenet or early email protocols like UUCP. The world wide web had not yet been invented.

This text-focused era reflected both the technical limitations and pioneering spirit of early hackers and computer enthusiasts. Graphics capabilities were limited, but the flexibility of the command line interface gave users unfettered access to the operating system. A single line of code could automate complex file management tasks. Text communication protocols like UUCP, Telnet, and FTP allowed the first online communities to spring up, connecting people across phone lines and primitive network connections.

The introduction of the graphical user interface (GUI) by Apple in the 1980s signaled a radical shift toward visual computing and mouse-driven interfaces. Microsoft Windows soon popularized the model further. Graphics, images, and multimedia content began to dominate software and the emerging world wide web. The average user was empowered by the intuitive simplicity of the graphical interface.

However, the transition to visual computing has also imposed limitations on how users can access and control their devices. GUIs hide the complexities of the system underneath, restricting access to only what the interface permits. The rise of mobile computing and locked-down hardware like smartphones and tablets has further restricted user control in favor of foolproof ease of use.

But text interfaces still have unique advantages that make them compelling, even necessary, for certain use cases. For expert and power users, the flexibility and automation capabilities of the command line remain unparalleled. Interface conventions like piping allow commands to be easily chained together to perform advanced operations not supported by GUI software. Scripting gives users fine-grained control to customize and automate repetitive tasks.

For communication and networking, open text-based protocols provide alternatives to closed corporate ecosystems like social media networks and centralized HTTP services. While less flashy and immediate than visual interfaces, protocols like email, IRC, Usenet, and FTP allow users to own and control their own content and connections. Protocols like Gemini and Gopher leverage the simplicity and universality of plain text to create open and decentralized alternatives to the web.

Importantly, text-based online communication protocols offer more censorship resistance than centralized and graphical services. Many regimes block access to social networks and other sites to censor dissent. But legacy text protocols continue to function as access to core internet infrastructure remains open. For example, Usenet remains popular for political discussion because the governments finds the decentralized text network impossible to fully block. Email and IRC channels similarly persist as channels for free speech.

Text-based tools also provide more freedom, privacy, and anonymity compared to audio/visual computing. Communicating and networking without photos, videos, or audio recordings allows users to share ideas and build connections without emphasis on identity and self-presentation. It returns the focus to quality of ideas over flashy presentation. Anonymity enables free speech without fear of repercussions.

Of course, visual interfaces also have benefits—they can be more intuitive, immediate, and expressive. But computing doesn't have to be an all-or-nothing choice between visual or text. The two can coexist, each used for its strengths. Graphical interfaces help bring computing to the mainstream, while text interfaces give experts and tinkerers access underneath to customize their experience. Visual social networks provide connections to the world, while text-based tools let users control their digital identities.

Computing has always evolved via cycles of openness and control. Mainframes gave way to personal computers. The openness of the early internet made way for consolidation under tech giants. But open protocols and grassroots communities persist, providing an important counterbalance. By continuing to use and develop text-based tools alongside modern graphical interfaces, we can keep computing open, empowering, and accessible to all. The text interface is dead, long live the text interface.