Literate programming is a programming paradigm introduced by Donald Knuth in which a computer program is given an explanation of its logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which compilable source code can be generated. The approach is used in scientific computing and in data science routinely for reproducible research and open access purposes. Literate programming tools are used by millions of programmers today.

The literate programming paradigm, as conceived by Knuth, represents a move away from writing computer programs in the manner and order imposed by the computer, and instead enables programmers to develop programs in the order demanded by the logic and flow of their thoughts. Literate programs are written as an uninterrupted exposition of logic in an ordinary human language, much like the text of an essay, in which macros are included to hide abstractions and traditional source code.

Literate programming tools are used to obtain two representations from a literate source file: one suitable for further compilation or execution by a computer, the "tangled" code, and another for viewing as formatted documentation, which is said to be "woven" from the literate source. While the first generation of literate programming tools were computer language-specific, the later ones are language-agnostic and exist above the programming languages.

This author provides practical advice about how to do transparent and reproducible data analysis and writing. We note that doing research in this way today will not only improve the cumulation of knowledge within a discipline, but it will also improve the life of the researcher tomorrow. We organize the argument around a series of homilies that lead to concrete actions.

(1) Data analysis is computer programming.
(2) No data analyst is an island for long.
(3) The territory of data analysis requires maps.
(4) Version control prevents clobbering, reconciles history, and helps organize work.
(5) Testing minimizes error.
(6) Work can be reproducible.
(7) Research ought to be credible communication.

following practices and recommended by Fredrickson, Testa, and Weidmann and Healy among others working on these topics allows our computerized analyses of our collections of stuff to be credible. If someone then quibbles with our analyses, our future self can shoot them the archive required to reproduce our work. We can say, “Here is everything we need to reproduce the work.” To be extra helpful we can add “Read the README file for further instructions.” And then we can get on with enjoying your life full of friends, children, laundry, news, or even journal reviews.