

# Lesson plan: “Reproducible reporting of computational analyses”

Stephan Koenig

## Topic: Literate programming

Literate programming mixes natural language with code.

### Learning outcomes

- Describe why combining natural language and code enables reproducible analysis and research.
- Discover how symbols and text convey formatting and the structure of a document.

### Engage (5 minutes)

- Make connection to previous module on diversity metrics. Give a simple code example that is also part of the worksheet to increase familiarity and reduce distraction for later.
- Think-pair-share: Increase awareness that analyses and their reporting are often disconnected and might get out of sync.

### Explore (10 minutes)

- Worksheet activity encourages students to explore the rules (syntax) of a sample literate programming document (Quarto format) in real life: reviewing someone else’s work. Exercises guide students to identify simple formatting markup (Markdown) and overall document structure (use of YAML, natural language and code) as part of Pandoc (enables the creation of many outputs).

### Explain (5 minutes)

- Explanation focuses on the most straightforward syntax, Markdown, because of the time limitation. This is the first time Markdown and Quarto are explicitly named to avoid the early use of jargon in the class.
- Discuss the flexibility of Quarto as a tool to generate different outputs, including the course website, providing real-world relevance.
- Preview of lab (see below).

### Elaborate (in lab)

- Students will create their own website/blog to act as an ePortfolio.

### Evaluate

- Student will complete an reflective exercise for each module and post on their website (participation).

- Syntax for Quarto will be tested on an online assessment platform with formative assessments (practice exercises) and one summative assessment (quiz). In addition, students will be asked to submit assignments as Quarto documents.