# What we talk about when we talk about monads

Tomas Petricek[a]

a The Alan Turing Institute, London, UK

**Abstract** Computer science provides an in-depth understanding of technical aspects of programming concepts, but if we want to understand how programming concepts evolve, how programmers think and talk about them and how they are used in practice, we need to consider a broader perspective that includes historical, philosophical and cognitive aspects. In this paper, we develop such broader understanding of monads, a programming concept that has an infamous formal definition, syntactic support in several programming languages and a reputation for being elegant and powerful, but also intimidating and difficult to grasp.

This paper is not a monad tutorial. It will not tell you what a monad is. Instead, it helps you understand how computer scientists and programmers talk about monads and why they do so. To answer these questions, we review the history of monads in the context of programming and study the development through the perspectives of philosophy of science, philosophy of mathematics and cognitive sciences.

More generally, we present a framework for understanding programming concepts that considers them at three levels: formal, metaphorical and implementation. We base such observations on established results about the scientific method and mathematical entities – cognitive sciences suggest that the metaphors used when thinking about monads are more important than widely accepted, while philosophy of science explains how the research paradigm from which monads originate influences and restricts their use.

Finally, we provide evidence for why a broader philosophical, sociological look at programming concepts should be of interest for programmers. It lets us understand programming concepts better and, fundamentally, choose more appropriate abstractions as illustrated in a number of case studies that conclude the paper.

**ACM CCS 2012**

- *General and reference → Surveys and overviews;*
- *Social and professional topics → History of computing;*

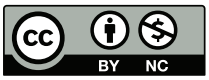
**Keywords** monad, programming concept, abstraction, philosophy of science, metaphor

The Art, Science, and Engineering of Programming

**Submitted** December 1, 2017

**Published** March 29, 2018

**DOI** 10.22152/programming-journal.org/2018/2/12




In *The Art, Science, and Engineering of Programming*, vol. 2, no. 3, 2018, article 12; 27 pages.