Functional Programming

Concept overview

Functional programming is a paradigm of programming–in other words a specific genre of computational thinking–in which problems are represented as functions, or transformations of inputs into desired outputs. Within functional programming, breaking complex problems down into simpler problems can be accomplished by composing functions, or making new functions out of simpler functions.

One characteristic aspect of functional programming is that functions are often passed as arguments to other functions–either to produce new functions, or to transform data. This perspective is important in data science: when working with pandas DataFrames, you apply functions to entire rows, columns, or tables of data. There are a number of interdisciplinary contexts in which a functional approach might be familiar, such as parsing writing with grammar rules, resolving mathematical expressions using order of operations, or planning complex logistical tasks in terms of subtasks.

How OOP could fit into my teaching context

As a sixth- and seventh-grade CS teacher with relative freedom to choose my curriculum, I could definitely imagine functional programming being part of my teaching. The main challenge is that I teach mostly using Scratch, and Scratch seems to commit pretty solidly to an object-oriented approach, not a functional approach. Functions are represented as blocks, and you can compose blocks (defining a new block by stacking existing blocks), but you cannot pass a block as an argument to another block. Furthermore, the main action in Scratch is oriented toward having sprites interact on a stage–very OOP.

I do some unplugged activites with my students too, and this could be a context for learning to model a problem in functional terms. We sometimes have conversations in class about what it means to grow up in the computer age; I wonder whether a functional perspecive might resonate with my students' experiences of moving through systems as if they were data moving through a functional pipeline.

How I might teach OOP

I really liked how the Pipes lab introduced functional programming, and I could see using this as a standalone lab with my students. It doesn't really involve much programming, I'm already teaching my students how to interact with the Terminal (they seem to quite enjoy it!), and I think they would enjoy both the domain results (word trivia) and the sense of power that one might feel interacting with such a scale of data. You can write a program which almost instantly searches, filters, and sorts all the words in the English language!