Lab 1: Introduction to Racket

In this lab, you'll get introduced to a new programming languages used in this course: Racket. Your TAs are prepared to help you get your software setup on your own computer, and will also be able to answer questions about the programming languages as you get started. However, one of the big goals of this course is to get you comfortable learning new programming languages independently, and so we'll also be encouraging you to look up documentation on your own for things like syntax and built-in functions.

Task 1: Software installation

If you're using your own computer, your first step should be installing the relevant software on your computer. Please follow the instructions in the software guide.

Note: You can skip this if you're working on a departmental machine, but we find most students want to use their own computer at some point during the term.

Task 2: Some interactive guides

In the software guide, we've provided two recommended interactive readings to orient yourself with the basics of Racket (and, later, Haskell). Please go through the Racket reading now. Note that for maximum learning, it's not good enough to just read the webpages; actually follow along by *typing any provided code on your computer*. Even though the contents of these references may seem pretty basic right now, you want to ensure you're extremely comfortable with the new languages before moving onto more complex material.

Task 3: Writing some basic functions

After you've gotten the hang of both Racket and Haskell, spend the rest of your lab time working on Exercise 1. For this first week only, your TAs are prepared to help you with this exercise, although we certainly don't expect you to complete it in the remaining lab time! (Your TAs will introduce new course material in future labs, but we're starting slow for the first week.)