Helpers Module

Help

Big problems have to be broken into small pieces before they can be solved. The recipes we have learned so far already do this to some extent. Data design is broken into parts, so is function design. And world design breaks the problem into several phases, ending with multiple functions to design.

In this module, we learn several new rules for breaking design problems into pieces, learning several rules for breaking a function down into multiple functions.

The good news is that while the problems get larger, the work to do at any moment in time does not get harder. Most of the work is HtDD and HtDF work you already know how to do. That's the contribution of the design method, to reduce ever more complex problems to smaller pieces you know how to solve.

The material in this module should take **approximately 5-7 hours** of dedicated time to complete, including working along with the lecture videos, doing the practice problems, doing the homework problems and the short quiz.

Learning Goals

- Be able to design functions that use helper functions for each of the following reasons:
  - at references to other non-primitive data definitions (this will be in the template)
  - to form a function composition
  - to handle a knowledge domain shift
  - to operate on arbitrary sized data

Lecture Videos, Notes and Starter Files

Topic	Length (mm:ss)	Starter File	Downloads
Helpers - Introduction Big design problems need to be broken into smaller pieces in order to be tractable. In this week we learn several new rules for breaking down function and data designs. We also see more examples of the form of information flows through to the data and the functions that operate on that data.	1:00	*none*	⬇ ☰ ☰
Helpers - Function Composition A function should be split into a function composition when it performs two or more distinct and complete operations on the consumed data.	13:45	arrange-images-starter.rkt	☰ ⬇ 🗎 ☰ ☰
Helpers - Laying Out a List of Images			☰

This function design requires no new techniques. It needs to be done as part of the larger problem, and it serves as practice and review.	4:05	arrange-images-v2.rkt	⬇ 📄 ≡ ☰
Helpers - Operating on a List When an expression must operate on a list -- and go arbitrarily far into that list -- then it must call a helper function to do that.	10:55	arrange-images-v3.rkt	≡ ⬇ 📄 ≡ ☰
Helpers - Domain Knowledge Shift When the body of a function must shift to a new knowledge domain it should call a helper function to do the work in the new domain.	14:48	arrange-images-v4.rkt	≡ ⬇ 📄 ≡ ☰
Helpers - Wrap Up The wish list process organizes the process of working through a design process, consisting of potentially many helper functions. The moment at which the last helper is finished, and "all tests pass" is very satisfying.	7:19	arrange-images-v5.rkt	≡ ⬇ 📄 ≡ ☰

Lecture Problems

Module Kind #	Assignment	Duration	Difficulty	Code Files	Requires Lecture
Helpers L1	Creating a program to sort a list of images and lay them out next to each other, as demonstrated in lecture, with all intermediate solutions.	60 min.	■	arrange-images-starter.rkt arrange-images-v1.rkt arrange-images-v2.rkt arrange-images-v3.rkt arrange-images-v4.rkt arrange-images-v5.rkt arrange-images-	Helpers - all

				v6.rkt	

Practice Problems

Module Kind #	Assignment	Duration	Difficulty	Code Files	Requires Lecture

Homework Problems

Module Kind #	Assignment	Duration	Difficulty	Code Files	Requires Lecture
Helpers H1	Design a world program to make it rain where you want it to. Uses all helper function rules to date. Total of 9 functions in our solution, so this problem takes some time. Do as much of it as you have time for.	120 min.	■	making-rain-filtered-starter.rkt	Helpers - all

Module Quiz

Be sure to complete the homework problems before you do the module quiz. The quiz itself can be found on the All Quizzes page.

Tips for Success

The programs get a bit longer again in this module but most of the work is with existing recipe skills. To avoid feeling overwhelmed it's really important to keep track of what it is you need to do NOW. Don't think about the whole problem at once - think about the current recipe step now and do the rest later.

To help with this you may find it helpful to keep a piece of paper showing where you are in the process. This would be similar to the diagram developed in the first several videos of the module.

Created Thu 29 Aug 2013 8:01 PM EDT (UTC -0400)

Last Modified Sun 29 Sep 2013 11:19 PM EDT (UTC -0400)