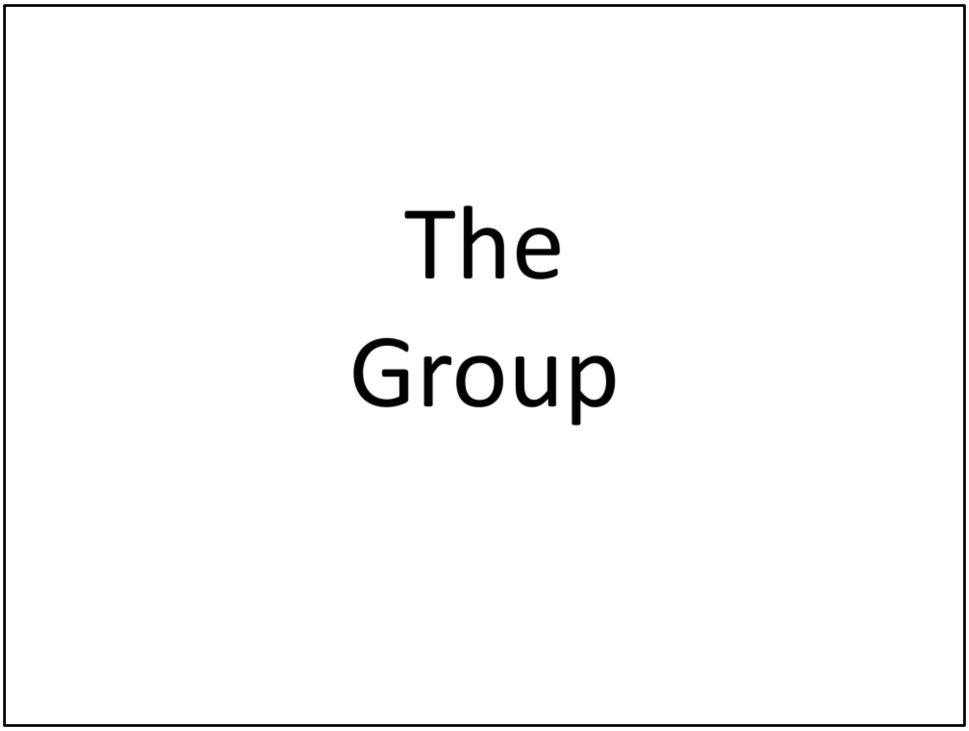
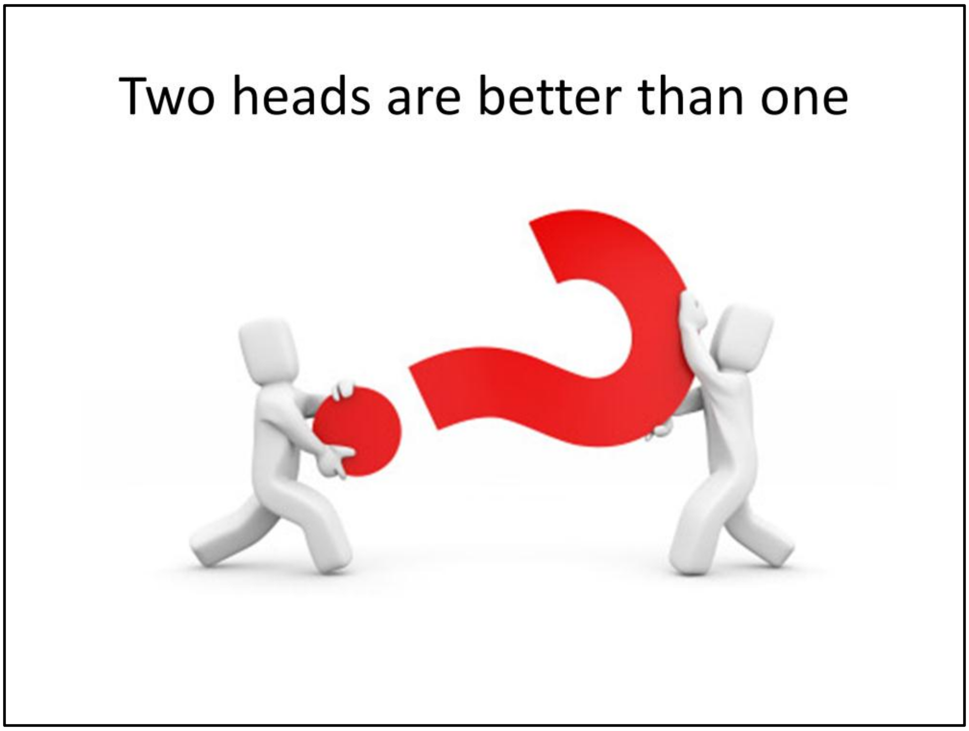


The Group

I’m going to talk a little about the group and how we started it, some of the choices we made up front.



Two heads are better than one

Sanj and I sent many emails back and forth bouncing ideas/thoughts/plans/initial timing before we asked anyone to join.

Planning for uniquely creative events is always easier with someone to validate or invalidate your thoughts.

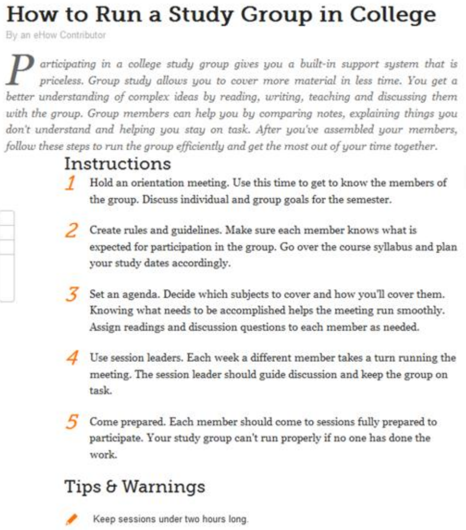
Commitment

We had a gentlemen’s agreement that even if no one else showed up we’ll still keep the group going until the end – our goal was to get through the book and motivate each other.

And we’d always be on hand to help out any members of the group that got stuck.

http://upload.wikimedia.org/wikipedia/commons/thumb/d/de/John_Hill_and_John_Linthicum_shaking_hands.jpg/886px-John_Hill_and_John_Linthicum_shaking_hands.jpg

Do some research



How to Run a Study Group in College

By an eHow Contributor

Participating in a college study group gives you a built-in support system that is priceless. Group study allows you to cover more material in less time. You get a better understanding of complex ideas by reading, writing, teaching and discussing them with the group. Group members can help you by comparing notes, explaining things you don't understand and helping you stay on task. After you've assembled your members, follow these steps to run the group efficiently and get the most out of your time together.

Instructions

1. Hold an orientation meeting. Use this time to get to know the members of the group. Discuss individual and group goals for the semester.
2. Create rules and guidelines. Make sure each member knows what is expected for participation in the group. Go over the course syllabus and plan your study dates accordingly.
3. Set an agenda. Decide which subjects to cover and how you'll cover them. Knowing what needs to be accomplished helps the meeting run smoothly. Assign readings and discussion questions to each member as needed.
4. Use session leaders. Each week a different member takes a turn running the meeting. The session leader should guide discussion and keep the group on task.
5. Come prepared. Each member should come to sessions fully prepared to participate. Your study group can't run properly if no one has done the work.

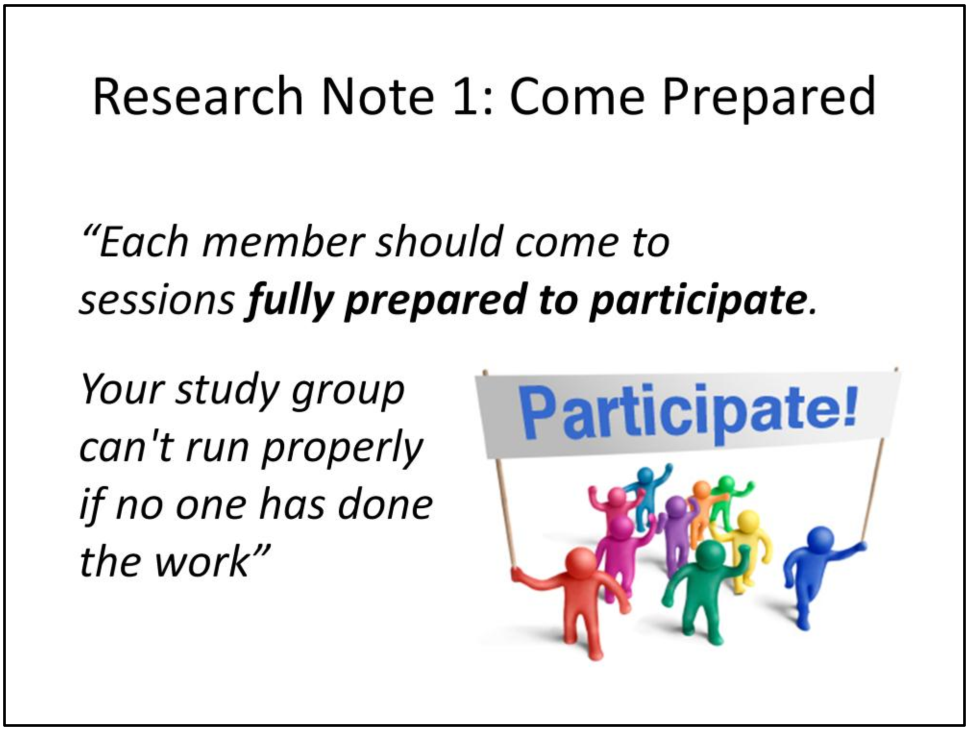
Tips & Warnings

- Keep sessions under two hours long.

http://www.ehow.com/how_2050686_run-study-group-college.html

How to run a study group

We searched on how to start a study group and found some of the following tips…



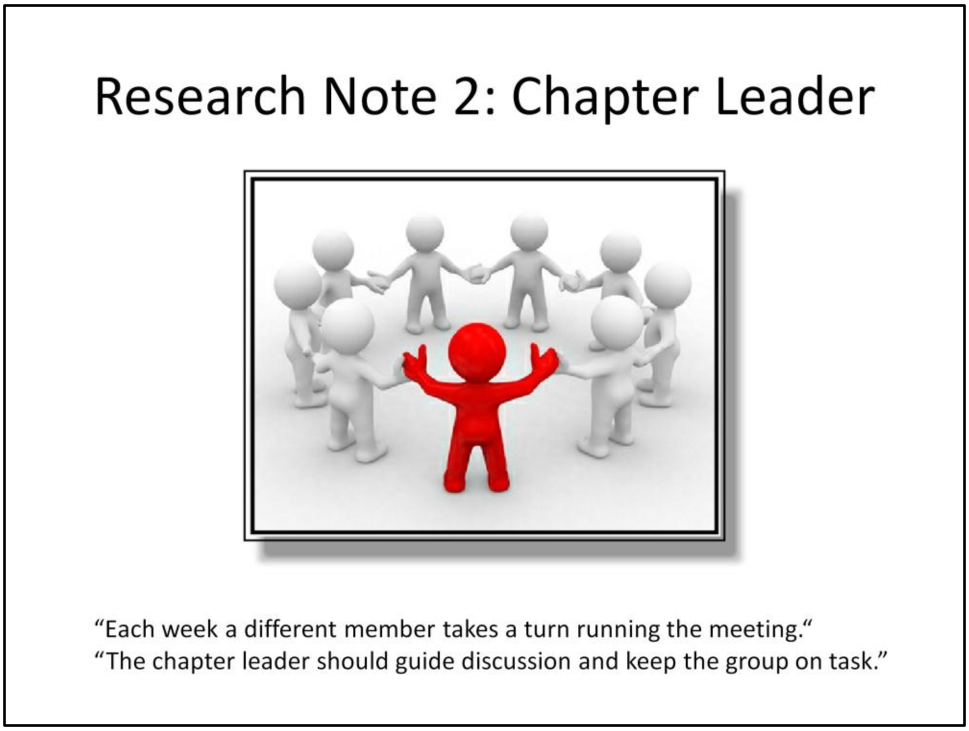
Research Note 1: Come Prepared

"Each member should come to sessions ***fully prepared to participate****.*

Your study group can't run properly if no one has done the work"

Participate!

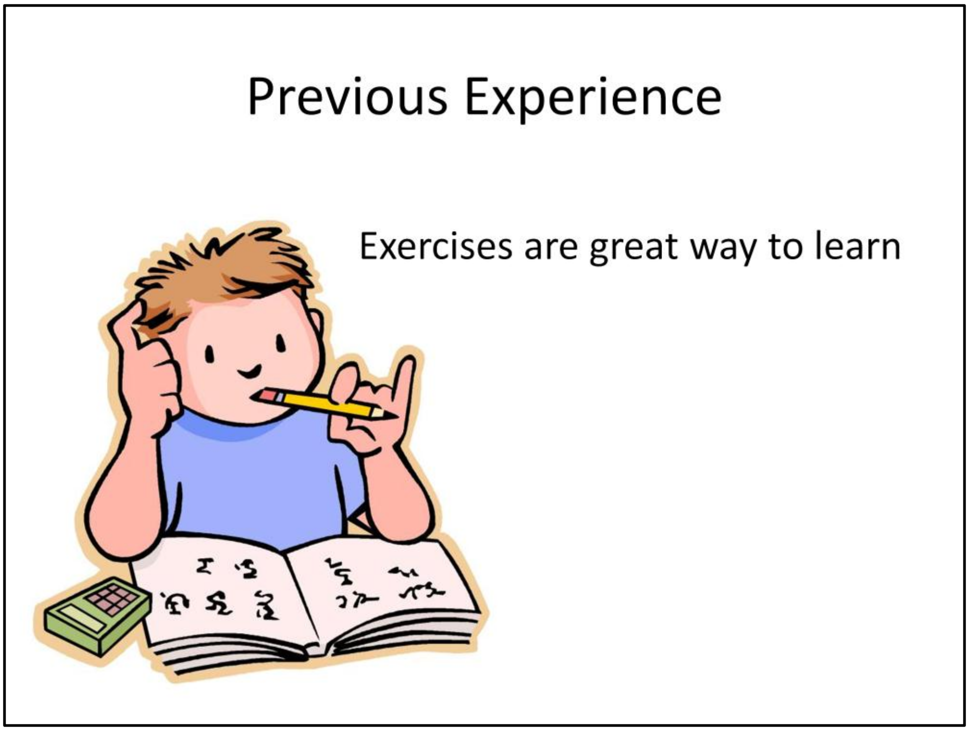
http://www.greenyourope.net/wp-content/uploads/participate.jpg



Research Note 2: Chapter Leader

“Each week a different member takes a turn running the meeting.“
“The chapter leader should guide discussion and keep the group on task.”

At first, people were a little apprehensive about taking on this role as none of us knew Haskell, let alone felt qualified to “lead” others. But everyone embraced the role. Sharing it around meant no one was burdened too much.



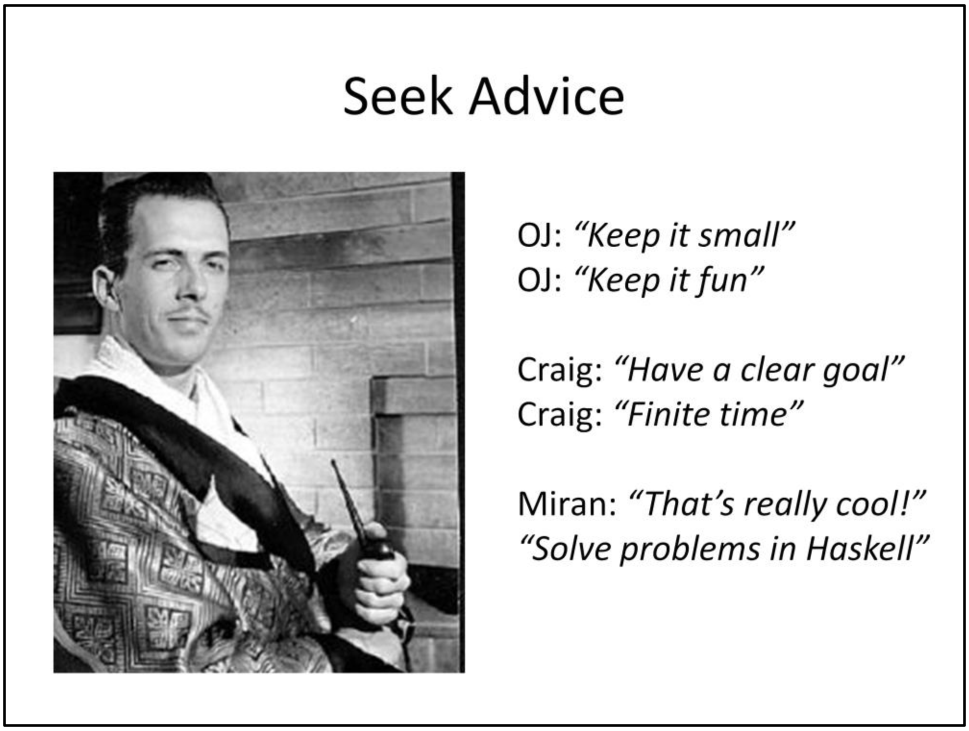
Previous Experience

Exercises are great way to learn

From previous groups we'd been part of we knew that exercises where a great way to apply learning.

We decided we'd set exercises in advance rather than do them live in the meeting – more on that later.

http://media.photobucket.com/image/do%20homework/teachertho/E6U5/3D223590509744AD804996300C0498A3.jpg



Seek Advice

OJ: *“Keep it small”*
OJ: *“Keep it fun”*

Craig: *“Have a clear goal”*
Craig: *“Finite time”*

Miran: *“That’s really cool!”*
“Solve problems in Haskell”

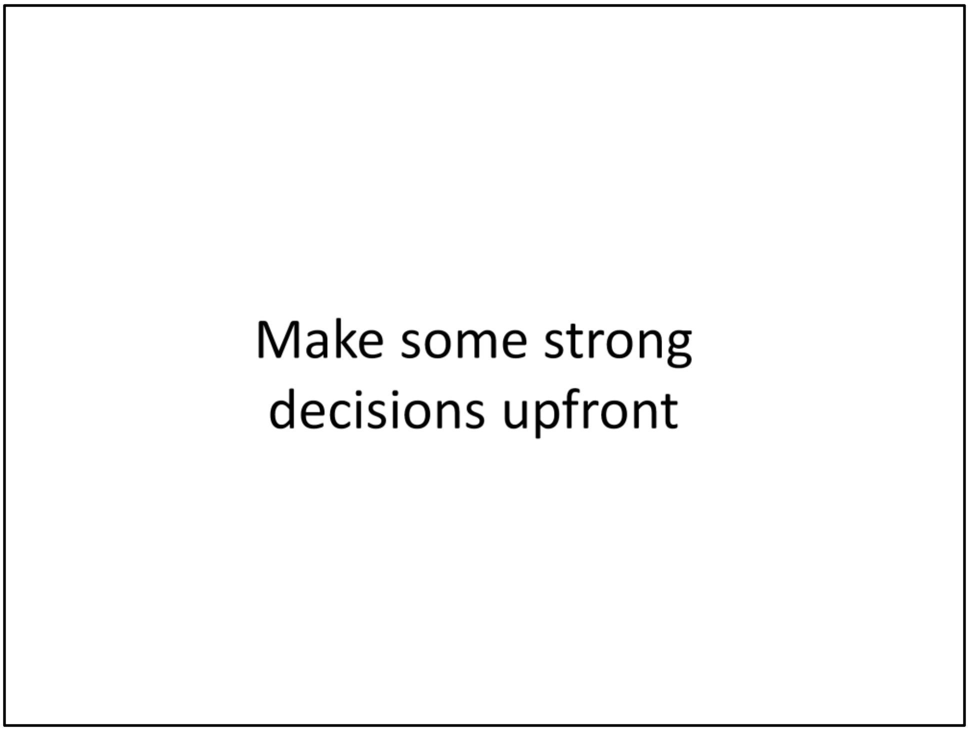
We sought the advice of a few people we knew had experience running FP groups.

OJ’s advice:
- Small groups of people are a much better option as motivation is way easier to keep up. I think it's a good idea and a great way to get across some of the key concepts without the "intimidating" presence of other people en-masse.
- Keep it to a group of 6 to 10 and no more.
- Get people interested and excited.
- Work through problems together.
- Keep it open and light-hearted.
- No experts <review>
- Avoid the purists view at all costs as it's off-putting to many.
- When you hit a wall, call for help.

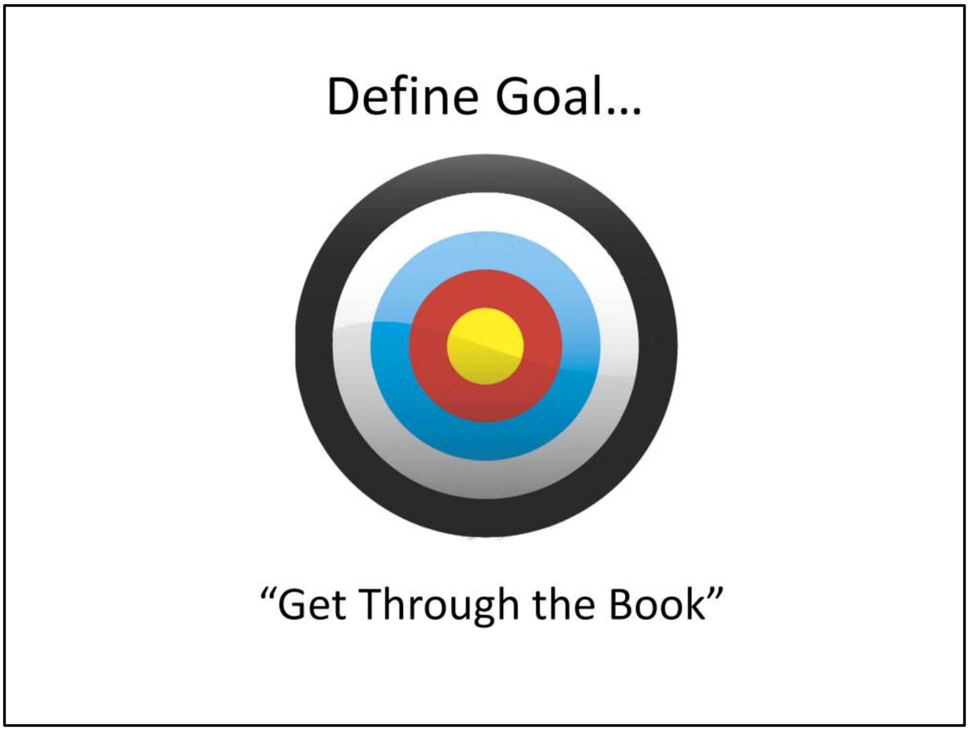
Got some ideas from Craig Aspinall:
- I guess the best advice I can give you is to have the purpose and scope of the group well defined from the outset, which I think you're doing anyway.
- Make it clear from the outset that you will be following the book and when you've finished the book, the group ends.
- That doesn't mean you won't start something else, but you're not putting yourself under pressure to!

Miran: We emailed the author of the book as we'd noticed the book and the online version were slightly different and to ask for any advice.
- That's really cool! It's always nice to see people organizing to learn and do neat stuff like this. Kudos.
- As for tips, hmm, I've never really done something like that, but your ideas seem like what I'd do.
- I'd organize a project that people build together and collaborate on.
- I'd also encourage people to bring problems or ideas to the group and then discuss how they can use Haskell to solve them.
- Nice! I checked out your wiki, looks real good man! Hope everything is going well!

http://3.bp.blogspot.com/-nx6D-f5mhSQ/T1h89j32B6I/AAAAAAAAfxw/Q9zrVqTS6bo/s400/Deploy+smoking+jackets.jpeg



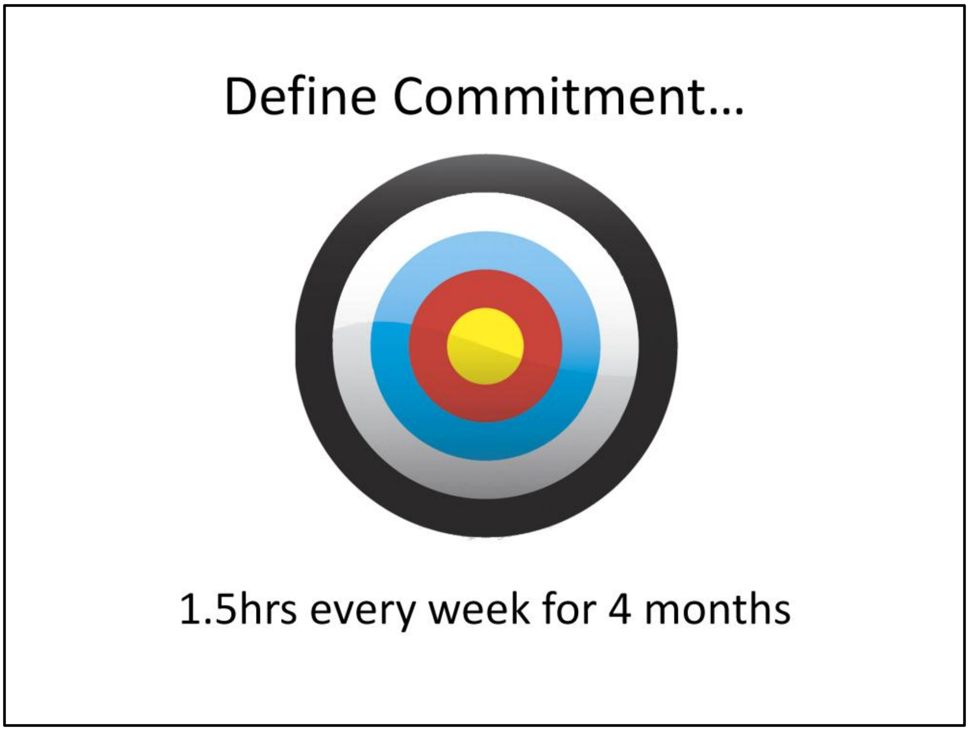
Make some strong decisions upfront



Define Goal…

“Get Through the Book”

http://www.communitycircuits.com/wp-content/uploads/2011/12/target1.png



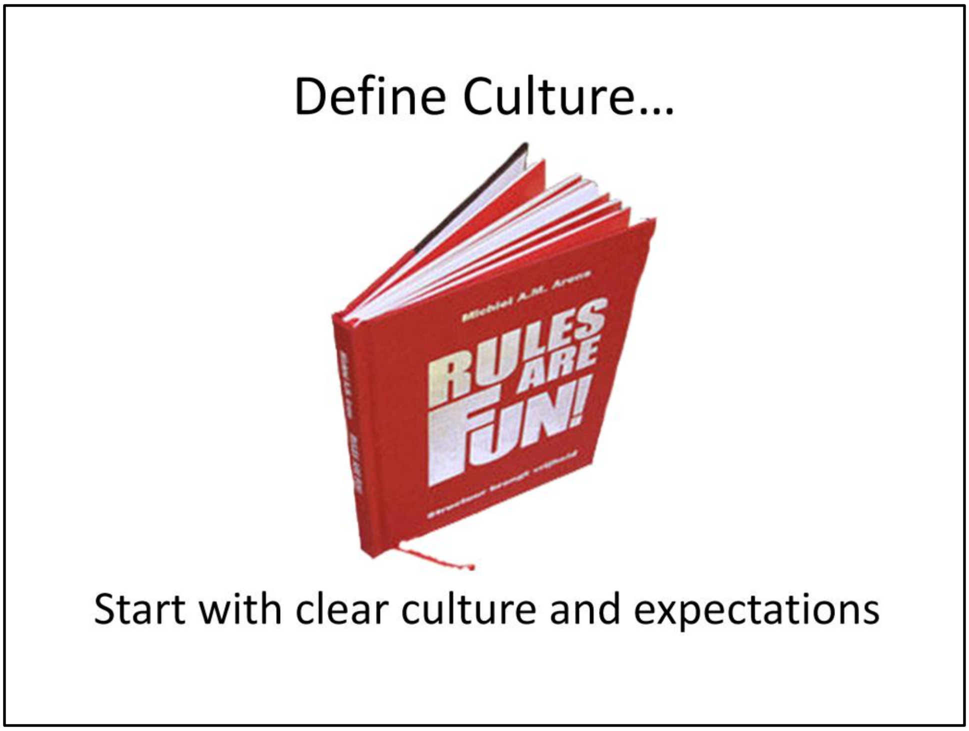
Define Commitment…

1.5hrs every week for 4 months

The goals and commitments created a natural selection criteria for those who want to join the group.

We recognised that FP is hard to learn – you will need to be committed to be a part of the group.

We decided to run the group at lunch time as it’s often hard to get a leave-pass for an evening session each week.

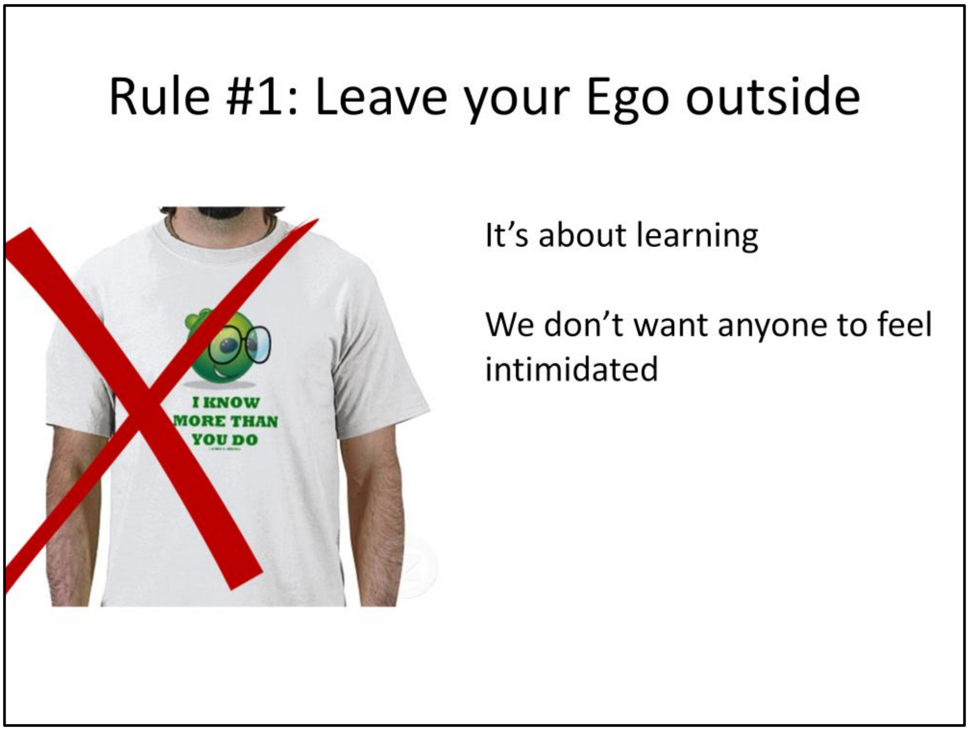


Define Culture…

Start with clear culture and expectations

Respect people’s time by making it clear what’s expected of everyone from the outset – important to get right from the outset before group culture sets in.

http://www.mavim.co.nz/assets/images/news/book-rules-are-fun.gif



Rule #1: Leave your Ego outside

I KNOW
MORE THAN
YOU DO

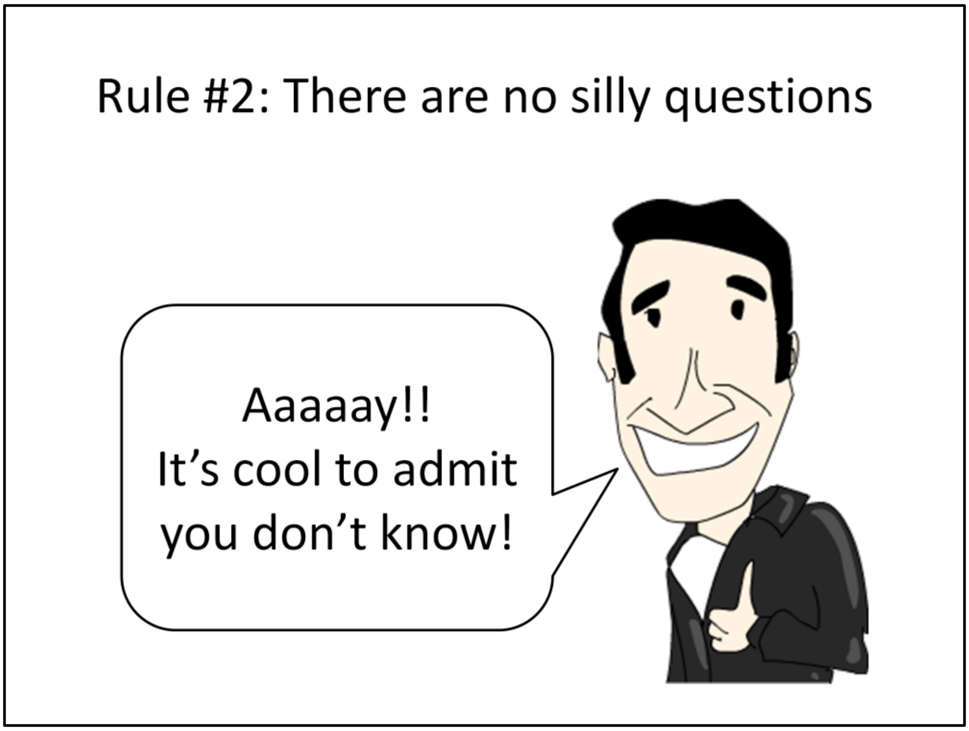
It’s about learning

We don’t want anyone to feel intimidated

We wanted to create an environment that was safe for learning.

So we made it clear that an “I know more about Haskell than you” attitude was counter-productive.

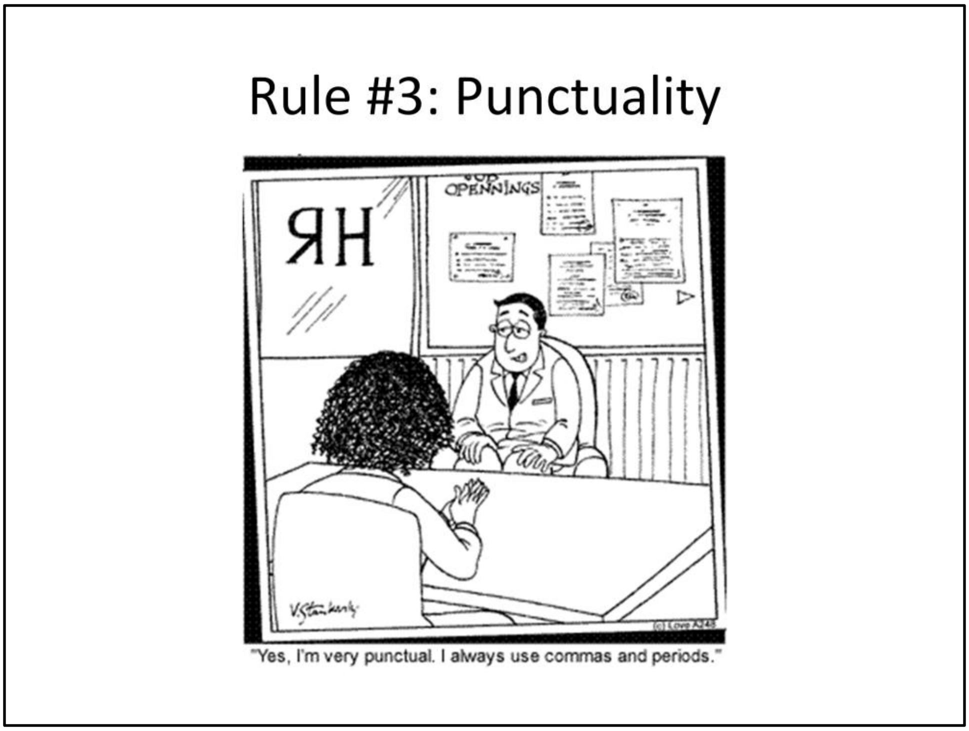
http://rlv.zcache.com/i_know_more_than_you_do_green_alien_expression_tshirt-p235265419555661170bahkm_400.jpg



Rule #2: There are no silly questions

Aaaaay!!
It’s cool to admit
you don’t know!

We worked hard to create an environment where people felt comfortable not understanding everything they read.



Rule #3: Punctuality

"Yes, I'm very punctual. I always use commas and periods."

Rewarding the late and punishing the tardy is a trap! You end up starting later and later to accommodate everyone, wasting the time of those who came on time.

We started every session at 12:00pm or at least by 12:05pm. And surprisingly every single session went for exactly 1.5hrs even when we thought we didn't have much to talk about.

http://xlab.berkeley.edu/images-content/punctual-participation.gif

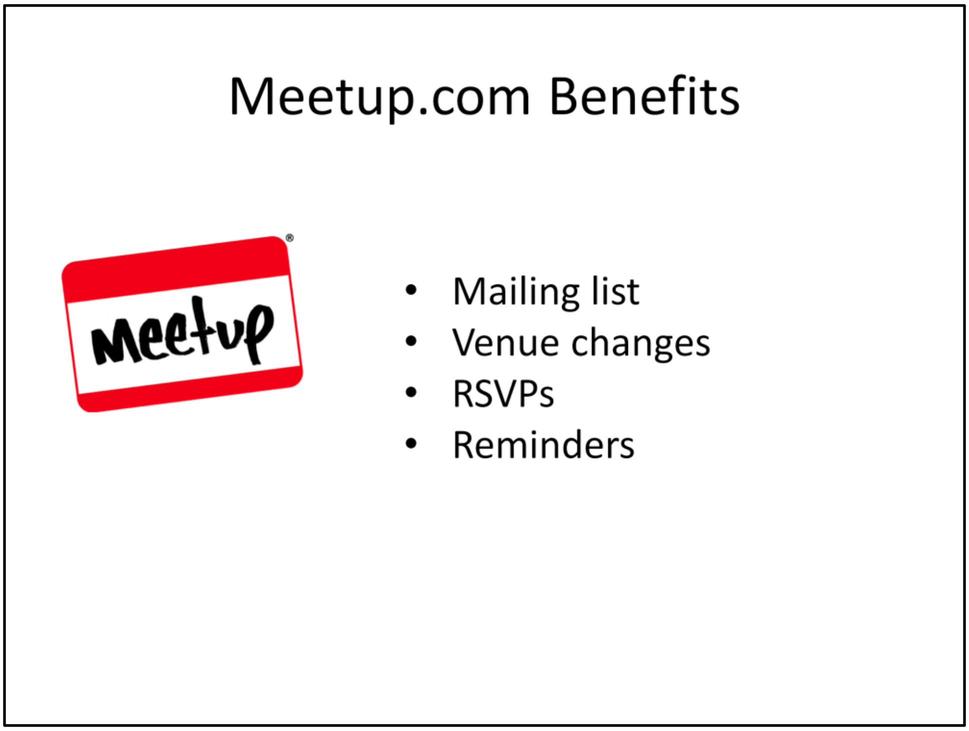
Rule #4: Have Fun!

Keep it lighthearted.

We even all bought matching Lambda t-shirts – just for fun.

Meetup.com

http://images.fastcompany.com/upload/610-meetup2.jpg



Meetup.com Benefits

- Mailing list
- Venue changes
- RSVPs
- Reminders

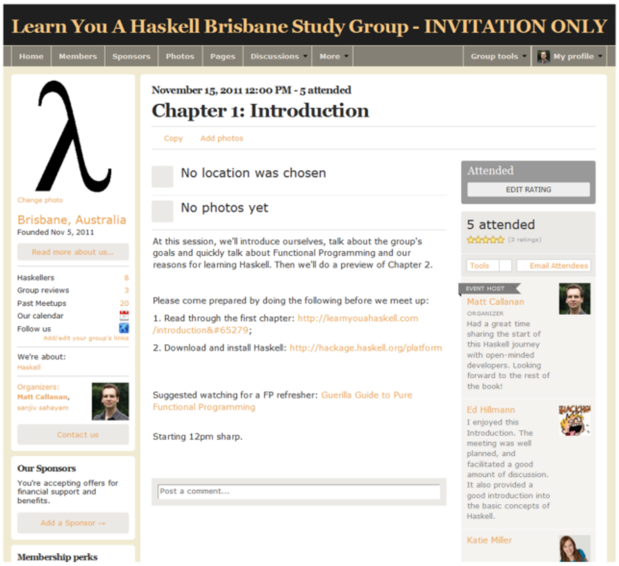
Benefits
- Email list – don't have to adminstrate everyone's email addresses.
- Venues – keeps track and informs people of changes
- RSVPs – collects RSVPs for you
- Reminders – reminds people of the sessions so you don't have to

We made our Meetup group closed as we wanted to keep the numbers small. So it was a private limited-size group that just happened to use the benefits of Meetup to get started and organise things.

- Initial settings:
  Either: Set it to "Anyone can try to join, but they must be approved by a group Organizer "
  Or: Leave it open and have a first in best dressed until you meet your quota
- Later: Set it to "No one, the group is closed to all new members"

$72 6mths

Meetup.com



Learn You A Haskell Brisbane Study Group - INVITATION ONLY

November 15, 2011 12:00 PM - 5 attended

Chapter 1: Introduction

No location was chosen

No photos yet

At this session, we'll introduce ourselves, talk about the group's goals and quickly talk about Functional Programming and our reasons for learning Haskell. Then we'll do a preview of Chapter 2.

Please come prepared by doing the following before we meet up:

1. Read through the first chapter: http://learnyouahaskell.com/introduction&#65279;

2. Download and install Haskell: http://hackage.haskell.org/platform

Suggested watching for a FP refresher: Guerilla Guide to Pure Functional Programming

Starting 12pm sharp.

Brisbane, Australia
Founded Nov 5, 2011

Haskellers	8
Group reviews	3
Past Meetups	20
Our calendar	
Follow us	

We're about:
Haskell

Organizers:
Matt Callanan, sanjiv sahayam

Our Sponsors

You're accepting offers for financial support and benefits.

Membership perks

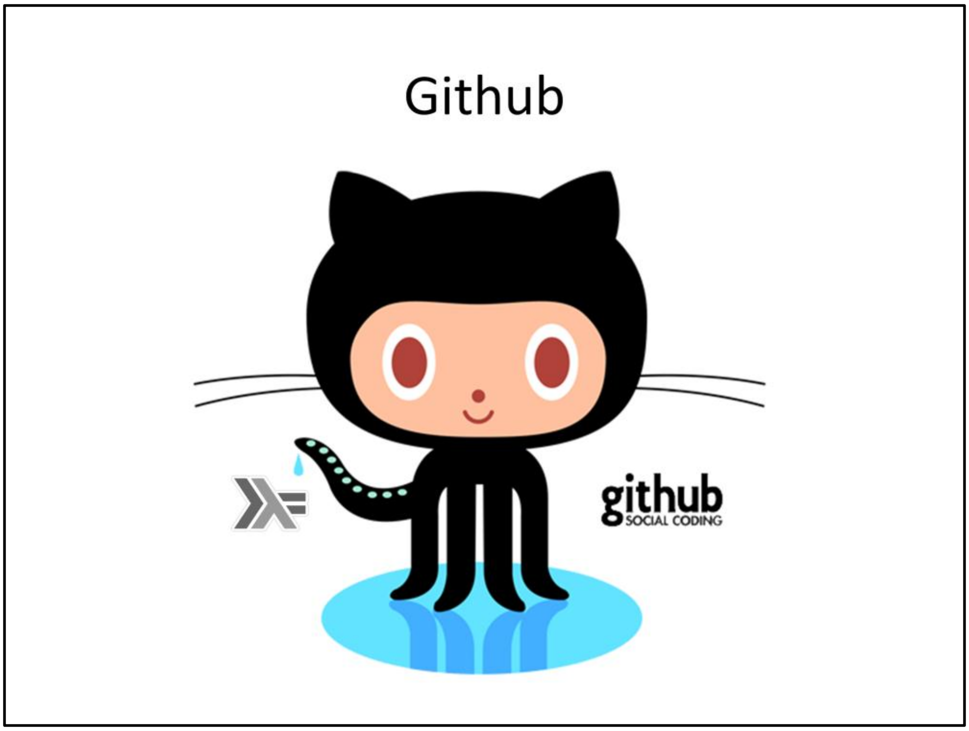
Attended

5 attended
★★★★★ (3 ratings)

Matt Callanan
ORGANIZER
Had a great time sharing the start of this Haskell journey with open-minded developers. Looking forward to the rest of the book!

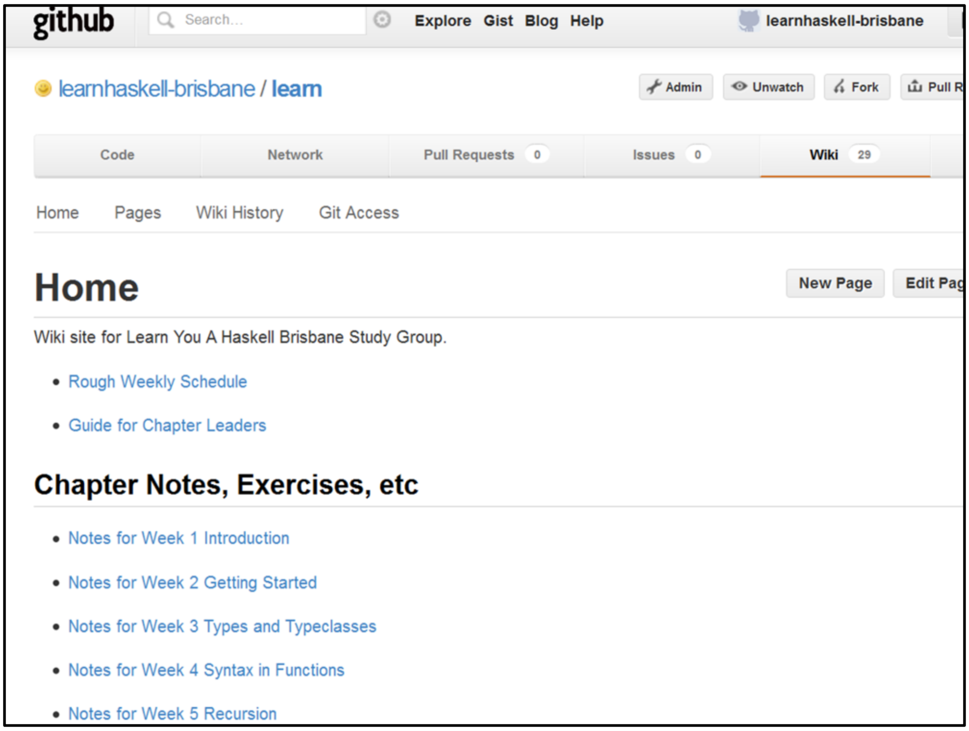
Ed Hillmann
I enjoyed this Introduction. The meeting was well planned, and facilitated a good amount of discussion. It also provided a good introduction into the basic concepts of Haskell.

Katie Miller



Github

Code collaboration
Wiki
Weekly Schedule
Assignments
Each chapter
- Links
- Notes
- Exercises
- Observations
- Questions
- Side notes



Home

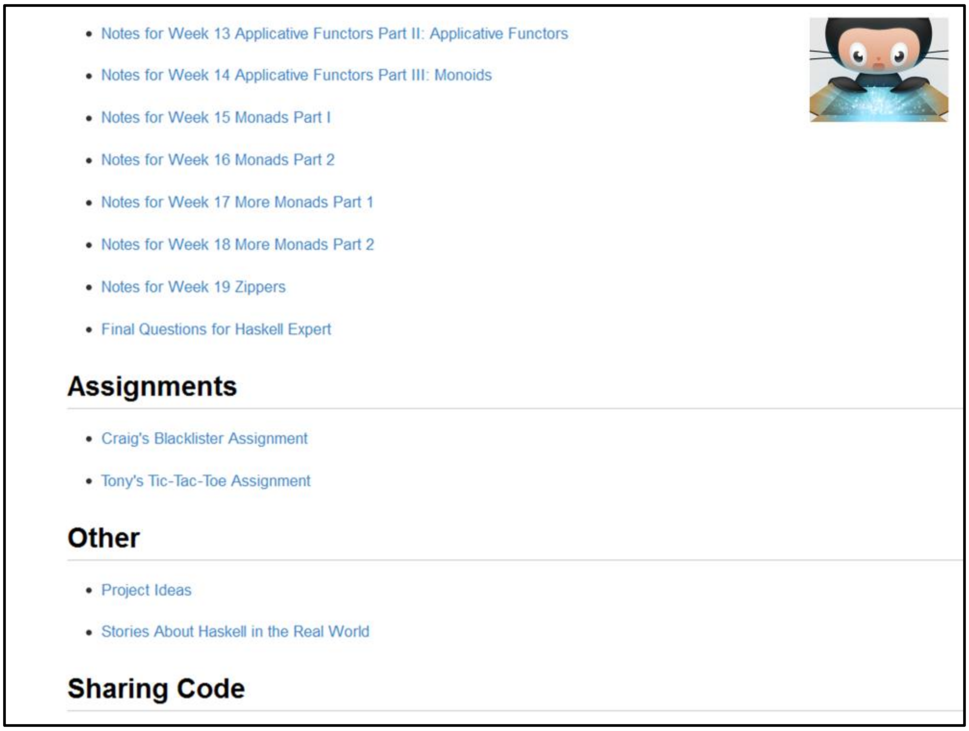
Wiki site for Learn You A Haskell Brisbane Study Group.

- Rough Weekly Schedule
- Guide for Chapter Leaders

Chapter Notes, Exercises, etc

- Notes for Week 1 Introduction
- Notes for Week 2 Getting Started
- Notes for Week 3 Types and Typeclasses
- Notes for Week 4 Syntax in Functions
- Notes for Week 5 Recursion

We used GitHub wiki



- Notes for Week 13 Applicative Functors Part II: Applicative Functors
- Notes for Week 14 Applicative Functors Part III: Monoids
- Notes for Week 15 Monads Part I
- Notes for Week 16 Monads Part 2
- Notes for Week 17 More Monads Part 1
- Notes for Week 18 More Monads Part 2
- Notes for Week 19 Zippers
- Final Questions for Haskell Expert

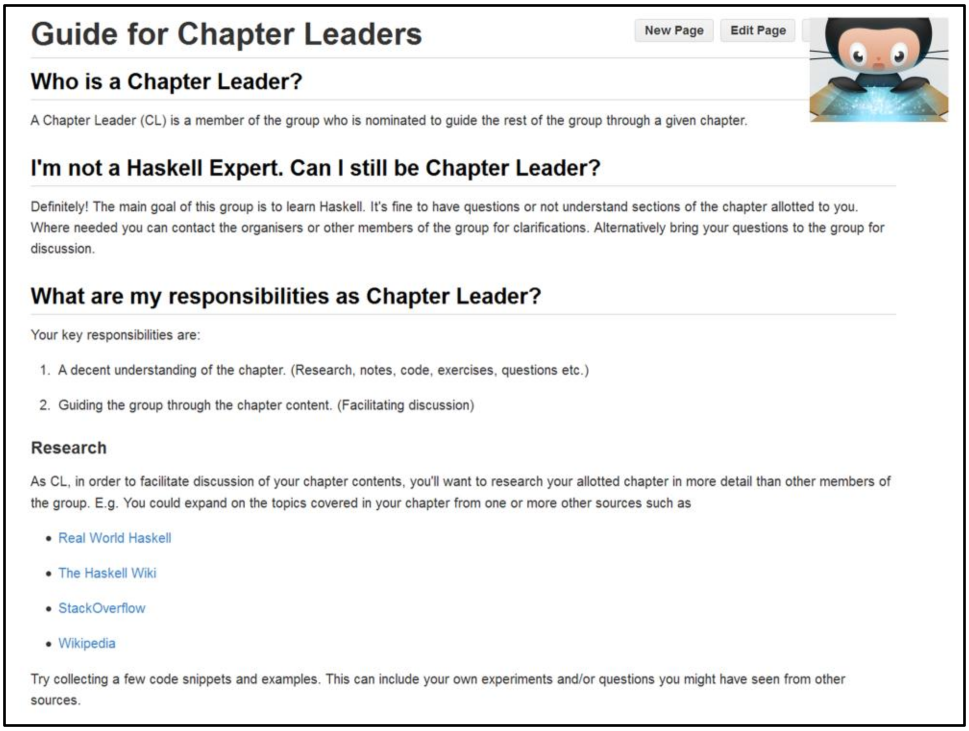
Assignments

- Craig's Blacklister Assignment
- Tony's Tic-Tac-Toe Assignment

Other

- Project Ideas
- Stories About Haskell in the Real World

Sharing Code



Guide for Chapter Leaders

Who is a Chapter Leader?

A Chapter Leader (CL) is a member of the group who is nominated to guide the rest of the group through a given chapter.

I'm not a Haskell Expert. Can I still be Chapter Leader?

Definitely! The main goal of this group is to learn Haskell. It's fine to have questions or not understand sections of the chapter allotted to you. Where needed you can contact the organisers or other members of the group for clarifications. Alternatively bring your questions to the group for discussion.

What are my responsibilities as Chapter Leader?

Your key responsibilities are:

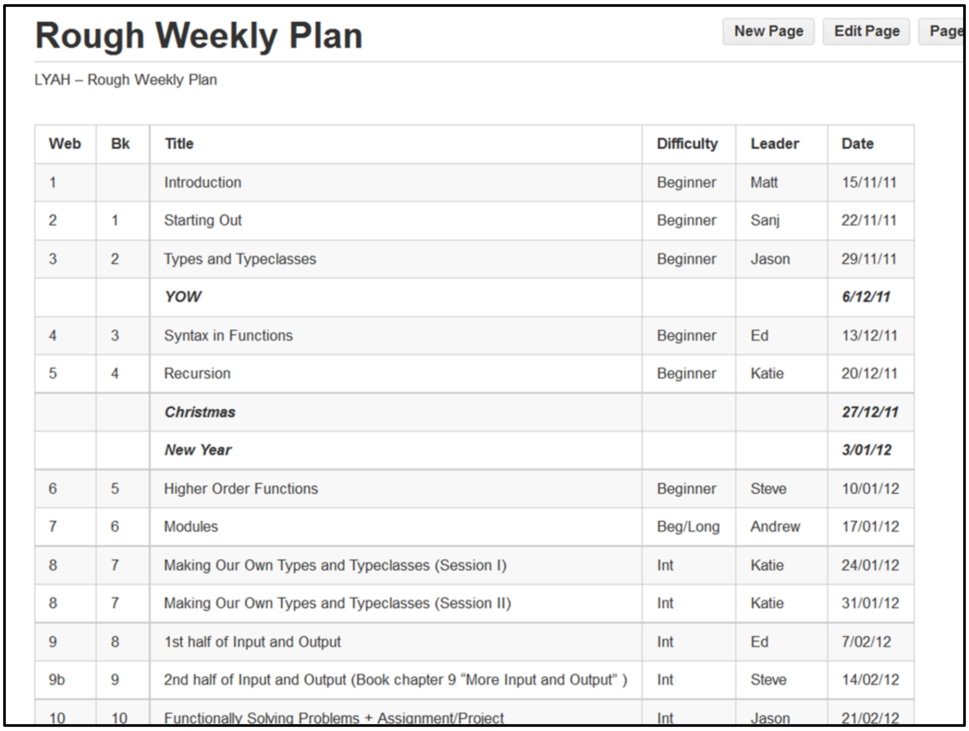
1. A decent understanding of the chapter. (Research, notes, code, exercises, questions etc.)
2. Guiding the group through the chapter content. (Facilitating discussion)

Research

As CL, in order to facilitate discussion of your chapter contents, you'll want to research your allotted chapter in more detail than other members of the group. E.g. You could expand on the topics covered in your chapter from one or more other sources such as

- Real World Haskell
- The Haskell Wiki
- StackOverflow
- Wikipedia

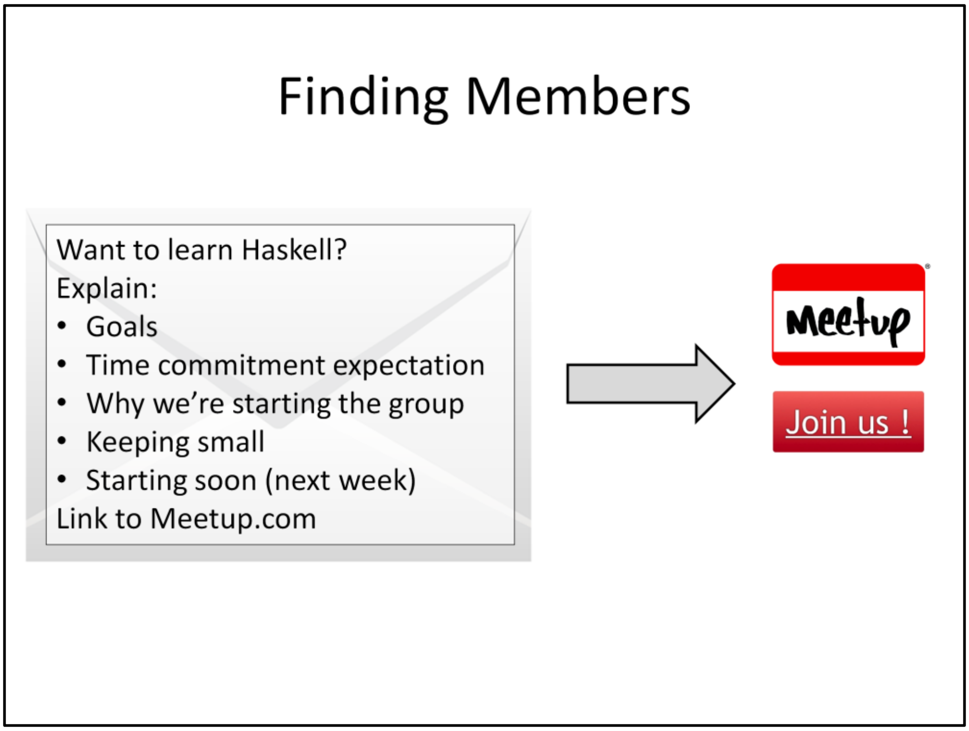
Try collecting a few code snippets and examples. This can include your own experiments and/or questions you might have seen from other sources.



Rough Weekly Plan

LYAH – Rough Weekly Plan

Web	Bk	Title	Difficulty	Leader	Date
1		Introduction	Beginner	Matt	15/11/11
2	1	Starting Out	Beginner	Sanj	22/11/11
3	2	Types and Typeclasses	Beginner	Jason	29/11/11
		YOW			***6/12/11***
4	3	Syntax in Functions	Beginner	Ed	13/12/11
5	4	Recursion	Beginner	Katie	20/12/11
		Christmas			***27/12/11***
		New Year			***3/01/12***
6	5	Higher Order Functions	Beginner	Steve	10/01/12
7	6	Modules	Beg/Long	Andrew	17/01/12
8	7	Making Our Own Types and Typeclasses (Session I)	Int	Katie	24/01/12
8	7	Making Our Own Types and Typeclasses (Session II)	Int	Katie	31/01/12
9	8	1st half of Input and Output	Int	Ed	7/02/12
9b	9	2nd half of Input and Output (Book chapter 9 "More Input and Output")	Int	Steve	14/02/12
10	10	Functionally Solving Problems + Assignment/Project	Int	Jason	21/02/12



Finding Members

Want to learn Haskell?
Explain:

- Goals
- Time commitment expectation
- Why we’re starting the group
- Keeping small
- Starting soon (next week)

Link to Meetup.com

Meetup
Join us !

Hey,

Thought you might be interested in a Haskell study group we're starting up. It's for people who want to learn Haskell from scratch, know some of it and want to know more, or who have forgotten what they've learnt and are interested in learning it again with the current resurgent interest in Functional Programming. Haskell's a great language to teach yourself FP principles and gives you skills you can apply to many other languages.

We'll be going through "Learn You A Haskell For Great Good” one chapter per week - starting from Chapter 1. It will take approx 1.5hrs of your lunchtime as well as personal time to dedicate to reading the book/website during the week.

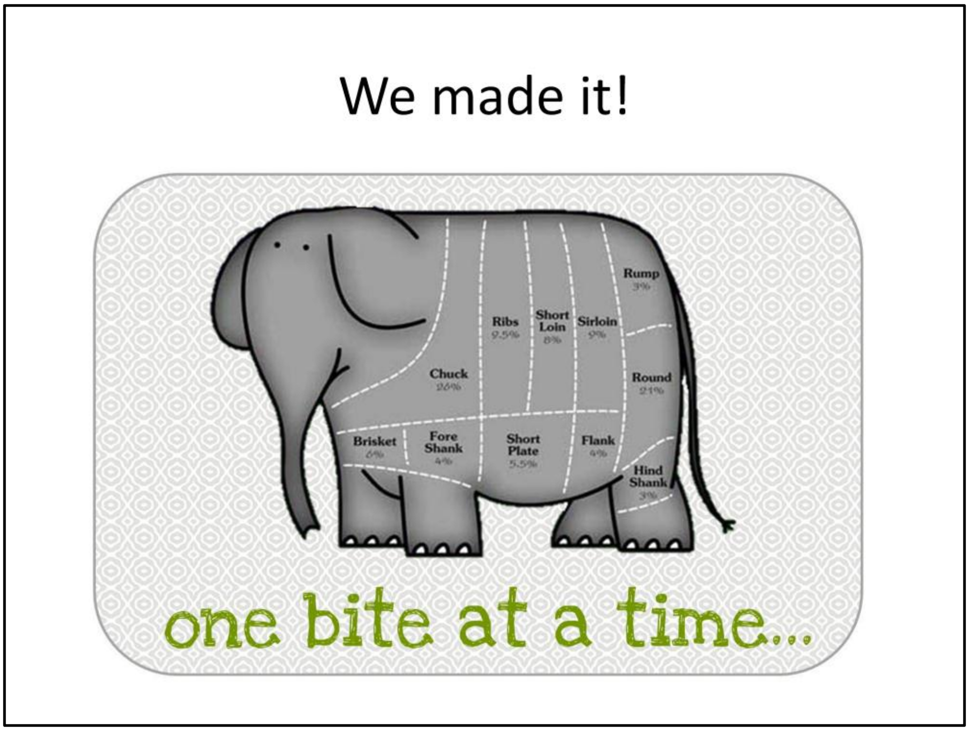
We're starting the group because some of us started the book at some point but didn't get all the way through it. The idea behind the group is that we help motivate each other with the goal of actually getting through the whole book (our plan is to finish it within 4 months - breaking for two weeks over Christmas/NewYear). Along the way we'll learn more about FP, discuss real-world applications and hopefully collaborate on some mini Haskell projects.

We're starting really soon (Tuesday week - 15th Nov) at lunchtimes in the CBD (location TBA). All you need to do before that first meetup is read the Introduction chapter and install Haskell as per the Introduction.

We're trying to keep the numbers small to maximise learning productivity - it's by invitation only - so let me know if you have any questions or sign up straight away at the Meetup page we're using to organise meetings:
http://www.meetup.com/LearnHaskell-Brisbane/

The Experience

Our experience:
- Running the group
- The Book itself



We made it!

I think we were all surprised we actually made it through the book… it's not often you get to read a book cover to cover.

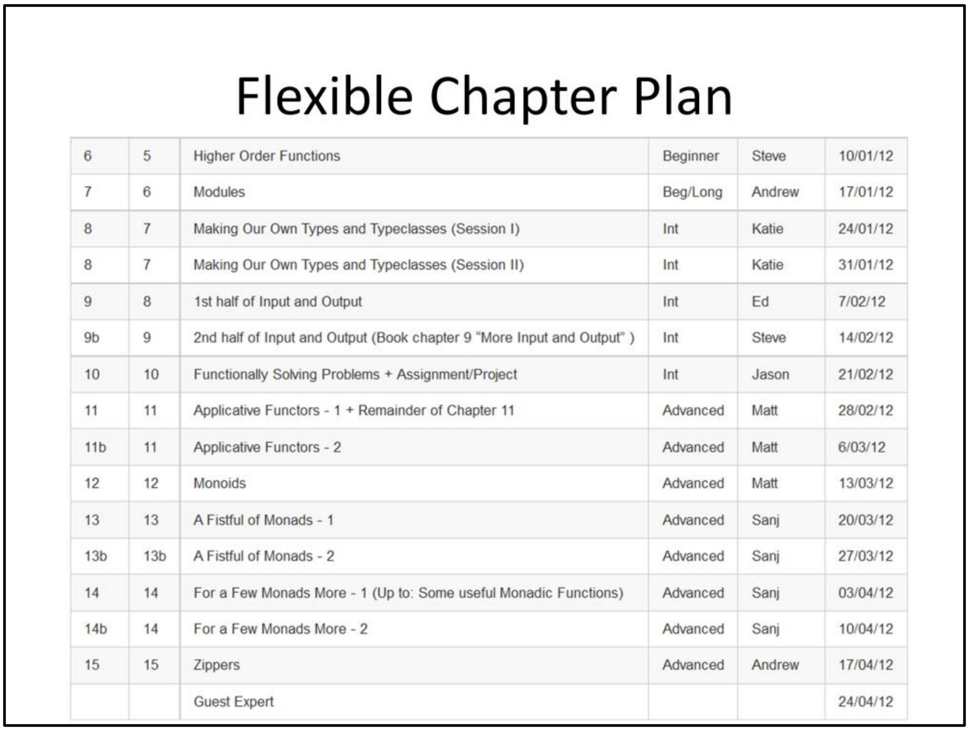
One of the biggest advantages of the group was a consistent, rhythmic commitment. Knowing that you'd be meeting next week to discuss the chapter and exercises drove you to find the time to do it (even if it was the morning of the meetup!).

There was a fair bit of effort involved in running the group too – we bounced a lot of ideas around.

But spread across 6 months and looking back it doesn't seem that burdensome – we're not saying "I'm glad that's over because it consumed my life" – we're saying "what do we do next?"

If you told us 6 months ago we'd be doing a presentation about how to learn Haskell we would have laughed at you.

http://www.ahappyplacecalledhome.com/storage/One%20bite%20at%20a%20time-001.jpg?__SQUARESPACE_CACHEVERSION=1334012020518



Flexible Chapter Plan

6	5	Higher Order Functions	Beginner	Steve	10/01/12
7	6	Modules	Beg/Long	Andrew	17/01/12
8	7	Making Our Own Types and Typeclasses (Session I)	Int	Katie	24/01/12
8	7	Making Our Own Types and Typeclasses (Session II)	Int	Katie	31/01/12
9	8	1st half of Input and Output	Int	Ed	7/02/12
9b	9	2nd half of Input and Output (Book chapter 9 "More Input and Output")	Int	Steve	14/02/12
10	10	Functionally Solving Problems + Assignment/Project	Int	Jason	21/02/12
11	11	Applicative Functors - 1 + Remainder of Chapter 11	Advanced	Matt	28/02/12
11b	11	Applicative Functors - 2	Advanced	Matt	6/03/12
12	12	Monoids	Advanced	Matt	13/03/12
13	13	A Fistful of Monads - 1	Advanced	Sanj	20/03/12
13b	13b	A Fistful of Monads - 2	Advanced	Sanj	27/03/12
14	14	For a Few Monads More - 1 (Up to: Some useful Monadic Functions)	Advanced	Sanj	03/04/12
14b	14	For a Few Monads More - 2	Advanced	Sanj	10/04/12
15	15	Zippers	Advanced	Andrew	17/04/12
		Guest Expert			24/04/12

Our initial naïve plan was to do one chapter per week – we did look a little ahead at the start and saw some big chapters and split them over two weeks.

Our initial rough weekly plan was changed a handful of times – usually to split up more chapters. I think we ended up with a good balance.

Split 5 out of 15 chapters across 2 weeks.
- Types & typeclasses
- IO
- Applicative Functors
- Monads 1
- Monads 2

22 sessions in total: 20 sessions for the book, one Haskell expert session, and a final lunch to wrap things up.

Group Motivation

- 6 months commitment is hard to sustain every week
- Life gets busy
- But commitment to group helps to prioritise

We all had periods over 6 months where we got really busy with life – sometimes it looked like some of us would drop out of the group.

Commitment to the group helped to prioritise and to regain focus

I think most of us wouldn’t have made it through the book if it weren’t for the group.

http://aerialamy.com/blog/wp-content/themes/simply-pink//2011/11/busy-calendar.jpg

Small Group

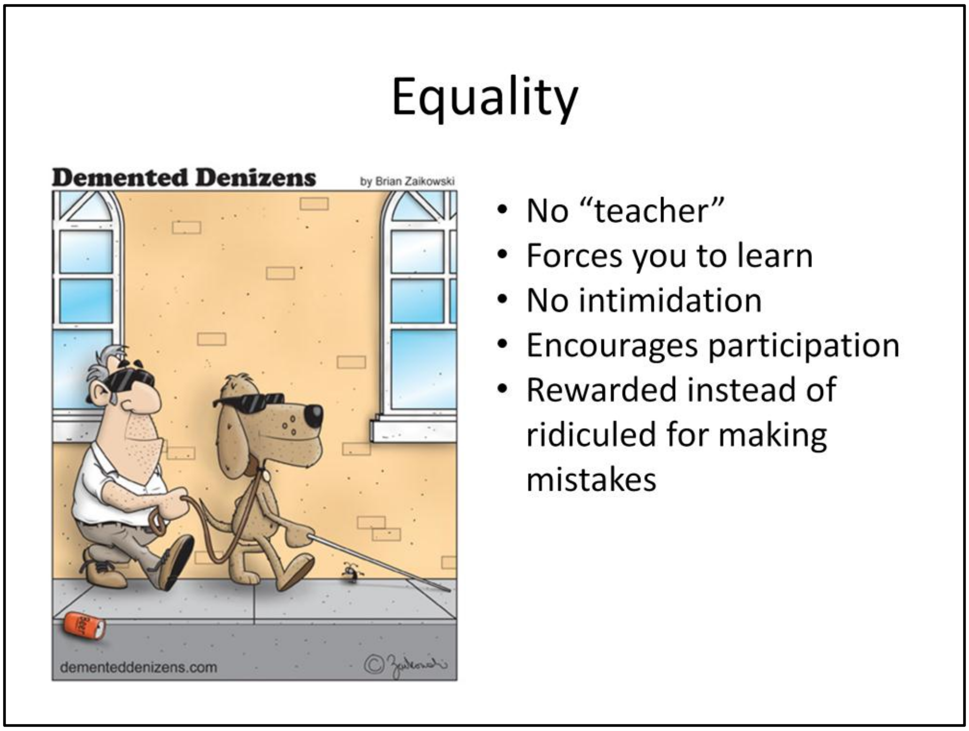
- Seven people
- Gave everyone a chance to contribute

Keeping the group size small was one of the biggest contributors to the success of the group.

Our group ended up being 7 people in total and it meant that everyone had a chance to contribute, to be Chapter Leader at least once.

Too much smaller and it might not have enough critical mass, too much bigger and it might start getting counter-productive or quieter people might not speak up.

http://clusters.wallonie.be/servlet/Repository/clustering-jpg.jpg?IDR=3762&IDQ=20&LANG=fr



Equality

- No "teacher"
- Forces you to learn
- No intimidation
- Encourages participation
- Rewarded instead of ridiculed for making mistakes

Not being guided by an expert was actually a good thing. This was one of OJ's suggestions.
- It forced us to connect the dots ourselves.
- Because we were all equals we knew we could all contribute rather than relying on expert to hand it down from on high to us.
- It also removed an element of intimidation and meant there was a freedom to make mistakes.

Provided a freedom to ask lots of questions you wouldn't normally ask at a group like BFPG because you don't want to waste anyone's time

Actually refreshing to admit you don't know or have all the answers… opens you and everyone else up for learning.

Making public mistakes in a supportive environment (failing early) means you learn and others learn – win-win for you and others and, in a way, everyone is rewarded for the mistakes you make.

http://www.dementeddenizens.com/images/blind_leading_blind.jpg.zip

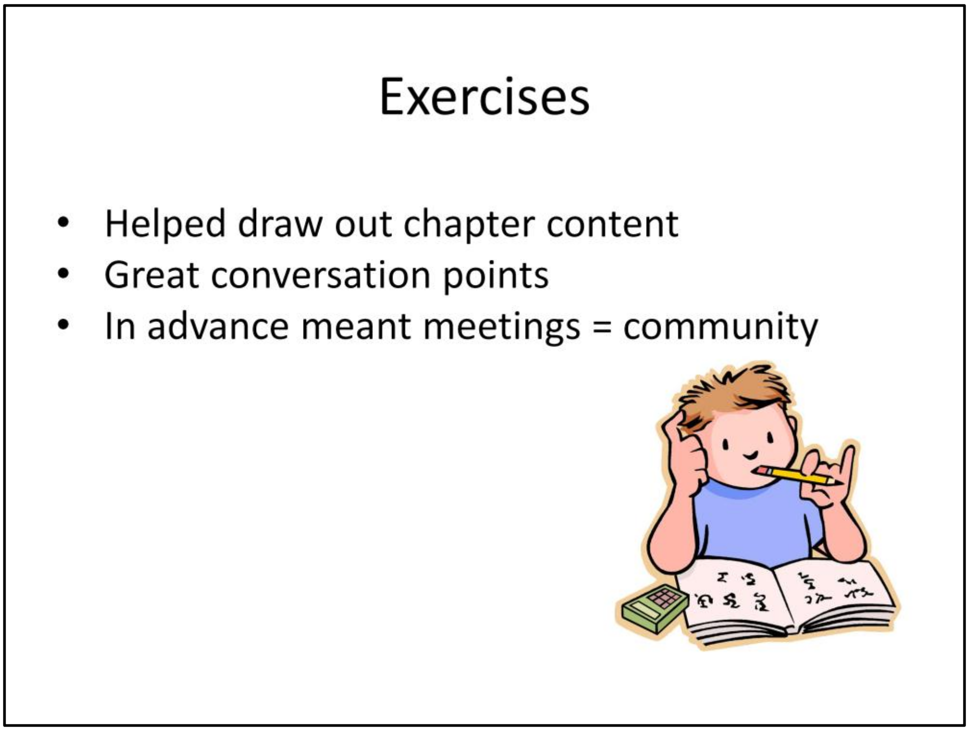
Outsource your learning

Everyone brings unique revelations to contribute

Everyone brings a different perspective to the session

Often find that even though you’ve read the whole chapter, someone will mention something that you didn’t notice.

It multiplies your own learning and means you learn more in-depth as a group than individuals.



Exercises

- Helped draw out chapter content
- Great conversation points
- In advance meant meetings = community

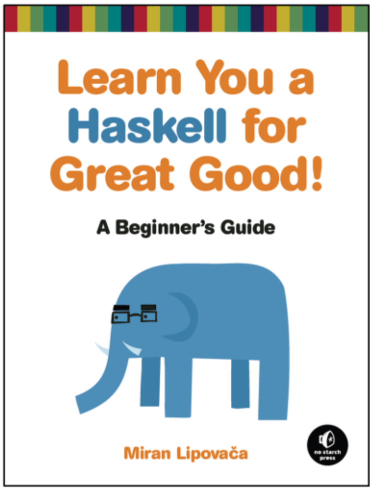
Setting exercises for the chapter ahead of the meeting really helped everyone to apply the content of the book.

Provided a lot of energy in the group sessions
- There was a buzz about the exercises before we'd even start the meeting
- Our sessions ended up being approx half book review, half talking through our answers to exercises with gotchas, lessons learned, different approaches.

Doing the exercises in advance meant that we made the most of our "together time" instead of being buried in our laptops.

http://media.photobucket.com/image/do%20homework/teachertho/E6U5/3D223590509744AD804996300C0498A3.jpg

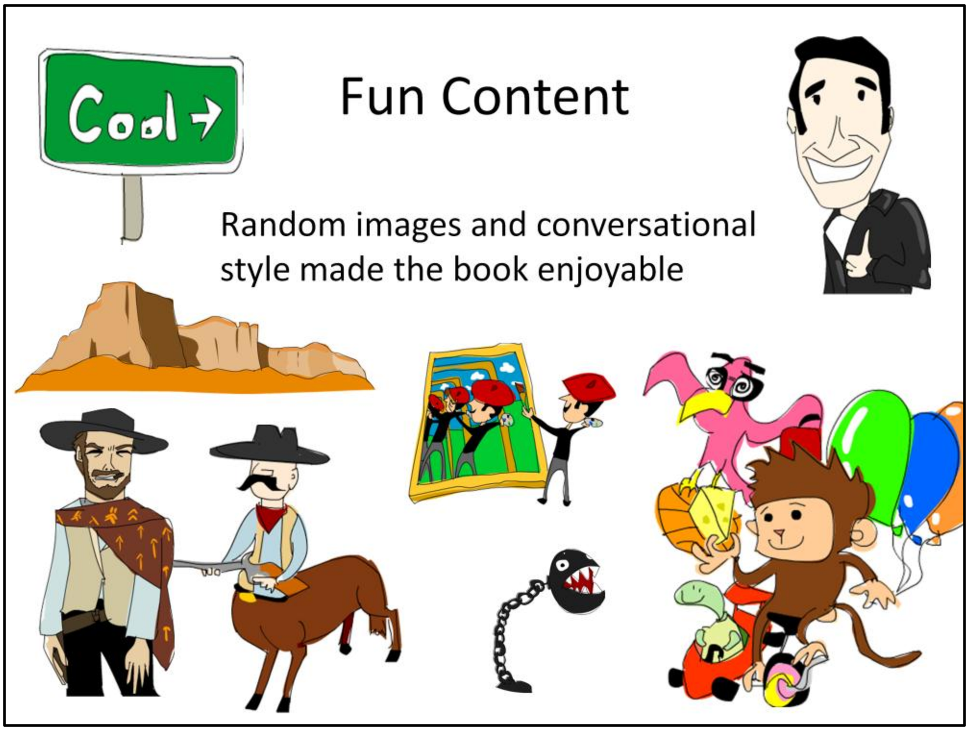
The Book



Learn You a Haskell for Great Good!

A Beginner's Guide

Miran Lipovača

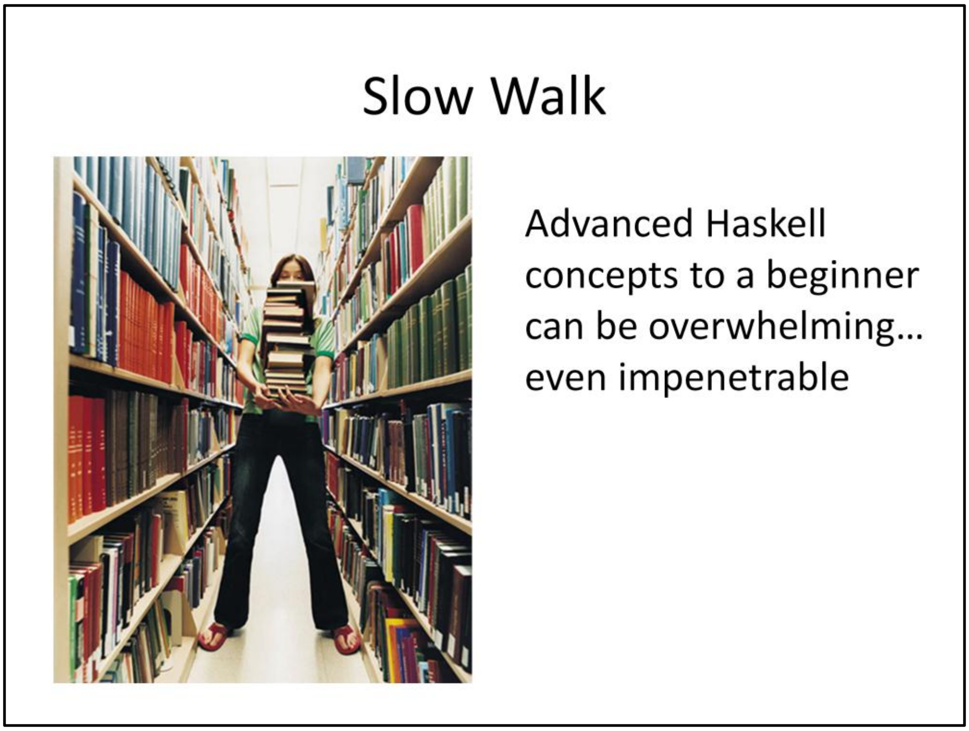


Fun Content

Random images and conversational style made the book enjoyable

Miran’s writing style added an element of fun to the group sessions.

The meaning of the images often provided conversation points in themselves.

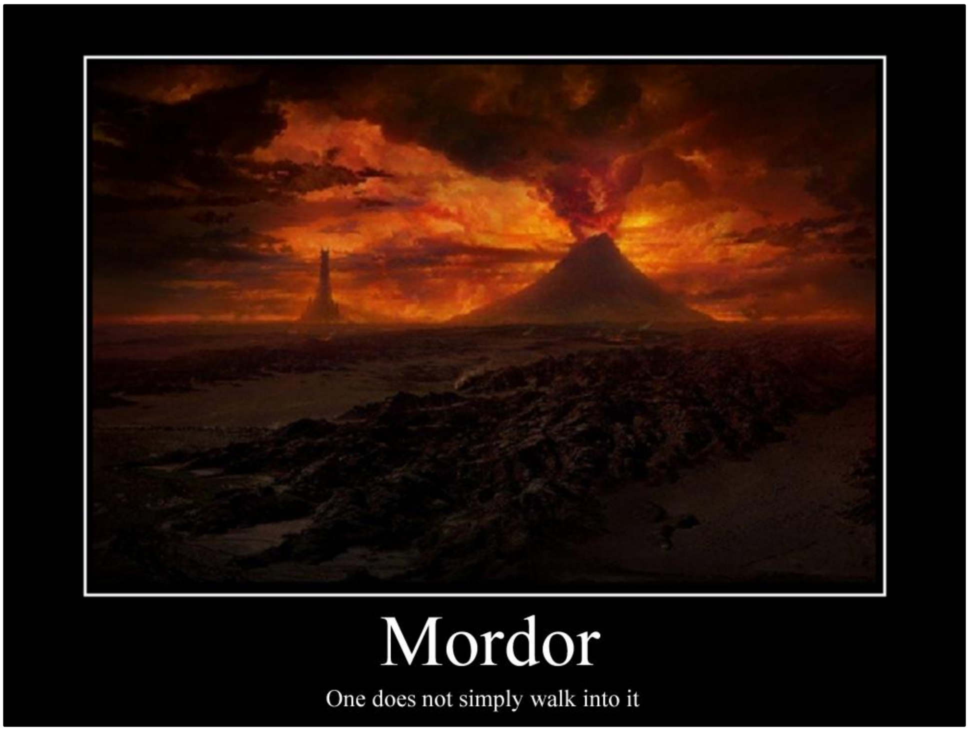


Slow Walk

Advanced Haskell concepts to a beginner can be overwhelming… even impenetrable

- More advanced Haskell concepts seem impenetrable to a beginner

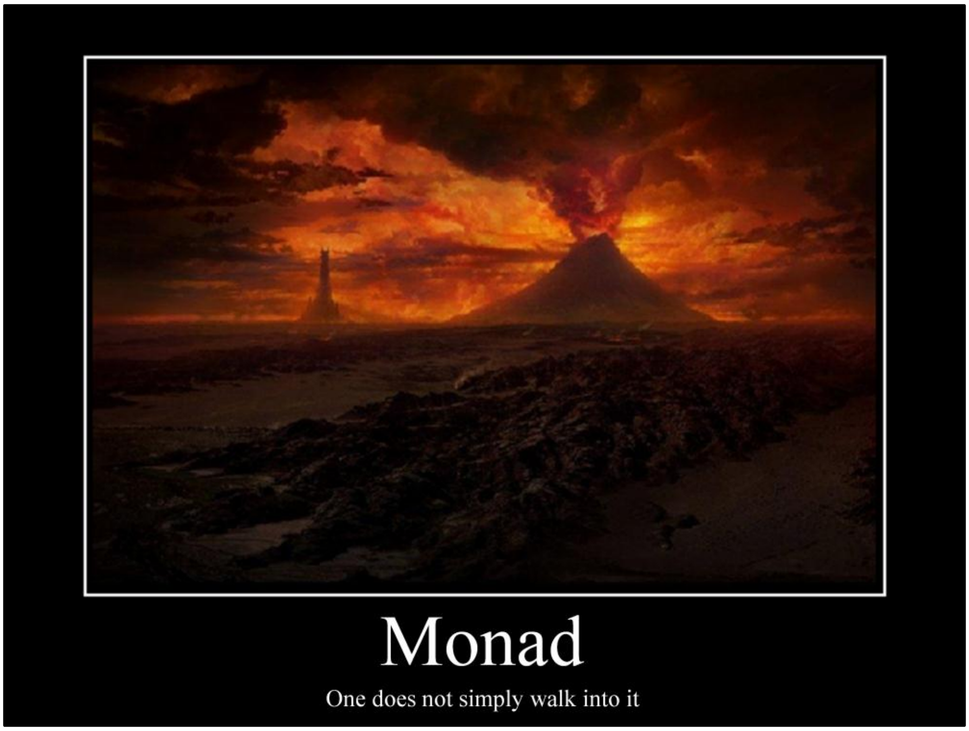
http://ifan.sce.pccu.edu.tw/inet/block/file/getimage.aspx?pname=19830401_phSu_20110914153555&lname=dv740057.jpg



This is the Google image result for “impenetrable”

The word Mordor reminded me of another scary word we find in Haskell…

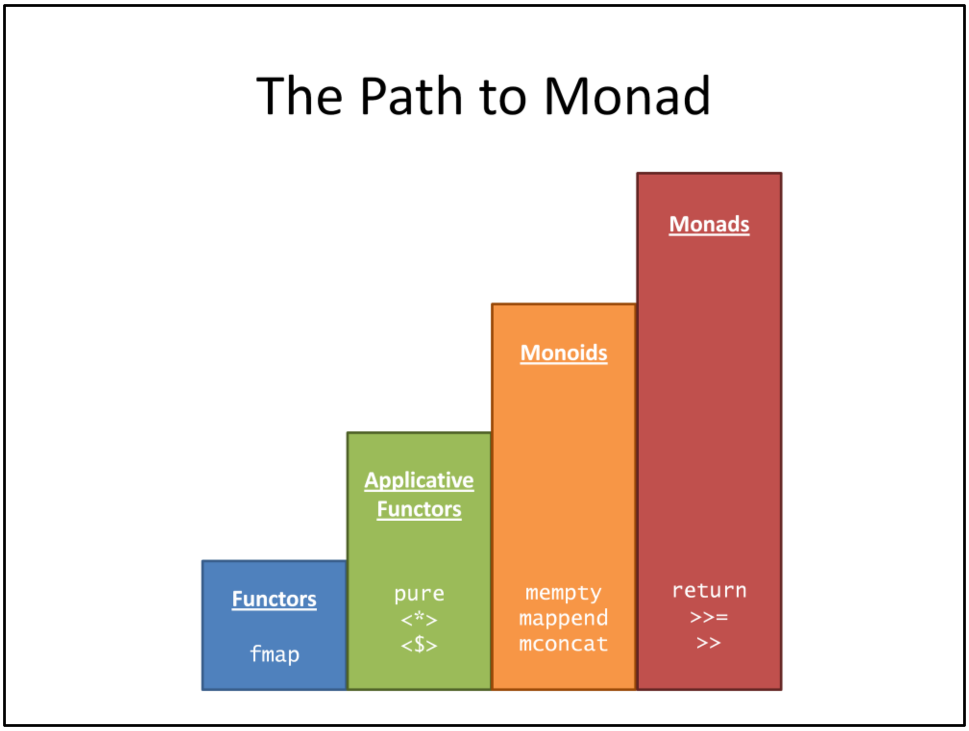
http://xspblog.files.wordpress.com/2008/06/mordor.png



One does not simply walk up to Monads.

It takes a bit of a warm up to get there.

And in fact the book provided us with a number of steps that made our climb to the peaks of Monad a little easier.



The Path to Monad

Functors

fmap

Applicative Functors

pure
<*>
<$>

Monoids

mempty
mappend
mconcat

Monads

return
>>=
>>

The path to Monad was a long one but made a lot more sense via this route.

A lot of other Haskell can be a bit intense when it comes to explaining concepts – you require a lot of extra knowledge to understand them.

The book provides a lot of code examples that are, on the whole, easy to understand… you can tell Miran has put in a lot of effort into understanding the concepts himself and he shares them in a non-intimidating way.