Breakdown

breakdown into the smallest blocks, then use tags/labels to naviagte them

we could also collect together things like typeclass what why pages as a list of links

each note should have one purpose, and say what it contains and its goal/aim e.g. this note tells you how the list data type is constructed in haskell. BY the end you shold understand the data definition of lists in haskel and why it is like that

Things to sort: types - what types - why (motivation) haskell types - basic (PoDs) haskell types - custom types haskell types - common structures product types sum types polymorphism / generalisation of types type classes - what type classes - why type classes - examples / common ones type classes - making one pattern matching if statemenents in haskell case statements in haskell haskell - guards haskell where clauses haskell let clauses how i program in haskell haskeel - function syntax - named and unnamed point free style lazy eval eager evel equational readoning in haskell structural induction proofs haskell lists haskell list comprehensions recurion recursive functions recursive data types map function #common functions #list function folds foldr #common functions #list function monoids monoid type class show type class

^ instead go through each set of notes in turn, creating a small module with aim, just like the quidditch drills, then slowly add to it from different sources

NON LINEAR factlets

we should have a glorrasry, and always link technical words like nlab

ways of navigating: beaten path - a suggested nav forging own way - link jumping, going from something u know then following suggested links teleport - search when u know what your looking for