**Motivation and Process**

* Do take a moment to outline your motivation to learn this stuff. Here are a few example motivations. Pardon the exaggerated tone. Some of them are empowering and some are burdensome. Take a moment to clarify your motivations and ensure that they are aligned with your career goals.
  + I want to get started with system design literacy - this is my field and I want to dig into some of these concepts so I have a better toolkit. I am looking forward to it!
  + I want to see how some big systems are designed so I can do more fun work! So I can keep up with design discussions better and so I can start contributing to system design.
  + I love decoding what's happening under the hood so I think this would be a great way for me to ease into the guts of the systems that I use and hear about!
  + I want to learn the basics, for interviews. And then, I would love to dig deeper into whatever sparks further curiosity.
  + It is crucial to know this, I should know this. If I don’t know this, things can go wrong for me. I will be left behind in my professional journey.
  + It's embarrassing not knowing some of these things and makes me feel less than. So, I must rush to learn all this.
  + I get funny looks from other engineers when I don’t know things and it feels awful so I want to not feel that way. Therefore, ready or not, I am deploying/ordering myself to conquer this battle with system design.
* Consider a loose structuring of this learning.
  + Pick a starter curriculum (fundamentals from HiredInTech, maybe?).
  + Now plan to graze and re-graze through this curriculum. Give it time to trickle in.
  + Round One: Perhaps read through all lessons first, with a soft focus, with no pressure to retain anything.
  + Round two: re-read each lesson/concept again. This time, pause after each page/section and try and summarize what you read. Classify what you read as “concept” or “design example/story”. So, basic scalability terminology will fall into the concept category and a section on designing youtube will be a story. You can read that story, it's ok if you do not understand things completely. If you let go of the expectation of complete in-depth understanding, then you can be comfortable with what you did understand and you can clearly express what you did not understand about the other parts. Perhaps things will get clearer in the future rounds.
  + Round three: You get the idea.
* Most of these resources are from interview preparation websites and programs. Don’t let that bother you. These are pretty decent compilations of concepts.

System Design Resources (Disclaimer: these are compiled from various sources, I have not consumed this information myself. Request: If you especially like or dislike anything, please let me know, I will update this for others!)

* Fundamentals (Pick one, perhaps?)
  + Scalability Fundamentals <https://www.hiredintech.com/classrooms/system-design/lesson/60> Some fundamental scalable design concepts and a link to the Harvard CS75 lecture.
  + <https://www.educative.io/courses/grokking-the-system-design-interview> (This includes stories and examples as well)
  + Basic terminology: <https://www.interviewbit.com/tutorial/basic-terminologies/> (This might be paid as well but I was able to click around and read a few things)
  + Chapters 3,4,9, 14 and 25 of the interview camp course cover system design curriculum (this is a paid course): <https://interviewcamp.io/courses/101687/lectures/16778545>
* Stories:
  + Start out with examples and stories of some interesting problems solved by some of the more successful products in the market now.
  + The [educative](https://www.educative.io/courses/grokking-the-system-design-interview).io course has examples of design stories.
  + Free: <https://www.hiredintech.com/classrooms/system-design/lesson/61>
  + YouTube design : <https://youtu.be/w5WVu624fY8>
    - Or this: <http://highscalability.com/youtube-architecture>
  + Facebook design: <https://youtu.be/UH7wkvcf0ys>
  + Dropbox design: <https://youtu.be/PE4gwstWhmc>
* Designing Data Intensive Applications:
  + For looking up principles and their application towards designing scalable systems: <https://dataintensive.net/> The biggest takeaway from the book will be to understand the fundamental principles of designing large scale systems, their tradeoffs and make informed decisions which result in building predictable systems at scale.
* IO and Filesystems**:** Try installing a FUSE system and debugging it. This will give immense confidence, just understanding this simple file system in userspace will prepare the mind to understand all the IO patterns and use cases in various types of applications all around us

[ML System Design guide](http://patrickhalina.com/posts/ml-systems-design-interview-guide/): This guide by Patrick Halina might be interesting for those dabbling more on the ML side of things.