# Abstraction

- Everything is about abstraction.
- Design, software, products, physics, reversing myopia, startups, business, users, strategy, countries, programming languages, variables, and concepts are all about abstraction.
- Abstraction is a way of meta-thinking.
- By abstracting things at a high level, you summarize them and can handle them in a batch. It is like a function or a method. It is like summarization in writing. It is like a table of contents. It is like a company structure.
- You can abstract things in different aspects.
- The standards for abstraction are what we need to think about. It is like how we extract or organize the code.
- We want to organize things by level and level. Each level should only interact with adjacent levels.
- If there is a problem or a bug, you can find the problems by starting from a high level. You should trace down to the root cause. It is like the corruptions of a country. They usually happen in some specific groups or people.
- The downside of abstraction is that it involves some high-level concepts. Individuals have more freedom in this world. The rules for a high level may not be suitable for all objects included in those levels.
- Things change. We may need to refactor code to have better abstraction. We may reorganize the company structure to better reflect the business and serve the customers.
- Think about the standards between every level. Do they have some patterns? If they do, it can be written as lambda code to construct high-level order functions.
- People wearing eyeglasses with full prescription to watch a blackboard (5m away) and a mobile phone is a poor abstraction. It does not better reflect the interests of the eyeball.
- The functions, methods, and procedures to apply that abstraction should be thought through carefully. Do we need to do the same task for the whole batch of objects, or should we treat them separately?