1. Loyalty towards project
   1. Architecture
      1. You know why the number of files is n. (where n is number of files)
   2. Uncertainty…
      1. Lower the cost of failure
      2. Be willing to a do a poc
2. Intent .. if we don’t want to get kidnapped by terminogies of company
   1. Generalize your learning
      1. You will less things to remember
3. Microscope .. programmer
   1. 5/5 variables
      1. Data type
      2. scope
   2. 5/5 in functions
      1. Calling a function
         1. Open documentation
         2. How do you know function failed or function suceeded
      2. Defining a function
         1. Think who is going to call it
         2. You should know how to tell the caller failed or succeeded
      3. Assigning a function
         1. Lamda
4. Datastructures
   1. What operations you plan to perform
   2. Pick one from the library
   3. Be conscious about what is data type of content in the datastructure
   4. Other person design
      1. \_\_equals \_\_\_
      2. \_\_str\_\_
5. Class
   1. Treat it like a project specific data type or architecture specific data tye
   2. Emp, Product -- Project specific data type
   3. Logic, Database – architecture related data type
6. Database
   1. You should databasenmae, table name, table structure
   2. Insert – fails only if constraint is violated..
   3. Update –fail constraint violation and where condition failure..
   4. Select – you have move the contents to the object
   5. Prefer sqlalchemy when compared to raw database access by using sql …
   6. Tool usage
7. Web services
   1. Chocolate and chocolate cover
   2. If you have a doubt on the chocolate focus on the chocolate ( Logic)
   3. If you have doubt on the chocolate cover ie http.. focus on that only.
   4. Once you are comfortable with 7.b and 7.c then join them.
8. Requests library
   1. Library that will make http requests and get the response..
   2. This is useful to other people api.
9. Logging, unit testing, security, tool usage
   1. Your project team members should help you better than going to google.
   2. Google, youtube, enjoy overload of information which is not relevant to you in the current project.
   3. Environment
      1. Virtual environment
10. Any programmer
    1. Problem solving..
       1. Rather than using memory.. consider using a checklist…
          1. Checklist = repeatability
          2. Memory ==noise…
    2. Should be good at variables/functions
    3. Doubts will keep coming on syntax, concept… tolerating uncertainty is important.