Aaron Krauss – Outline
Estimated Speaking Time: 40 min
What to Know as You're Becoming a Developer

I. **About Me**
   a. Name, age, basic info, etc.
   b. My path to programming
   c. Where I work now, what we do, and my job as a developer

II. **Importance of Learning about Programming on Your Own**
   a. 'You can be the best programmer in the world, if you take the initiative'
   b. Intro into working in the programming world
      i. Pros: You can find a job literally anywhere, options for working remotely, pay is good, incredibly cool coworkers, always an opportunity to learn
      ii. Cons: occasional pressure to keep up with other developers
   c. List the languages I know, and how/why I learned them. Explain that the programming world is massive with how much you can learn.
   d. It's not about how much you know, it's about showing that you want to learn more. That's what people want to see.
   e. Very common interview question: 'What projects do you do on the side?'
   f. Big point: Find something that you like, and take the time to learn more about it on your own time. That's the key to being the best developer.

III. **Software Dev. vs. Web Dev. vs. Mobile Dev.**
   a. List different types of jobs/careers for each, and which are most common
   b. Discuss languages used for each

IV. **Microsoft vs. Unix Development Environment**
   a. Explain languages used for each type of development
   b. List types of companies that typically use each language, and the environment you'd probably experience
   c. Explain why I personally like open-source technology:
      i. It's completely free, massive community, you have the ability to modify code and even change the original code, typically in a Unix/Linux environment, you use the Terminal very heavily, the GitHub community … but mostly, it's completely free and you can use so many tools/languages for whatever purpose you want.

V. **Coding in a Team**
   a. Version Control:
      i. Every coding team uses version control – it's not programming, but it's a skill you'll absolutely need to be familiar with.
      ii. Explain the problem of managing code when you don't have versioning
      iii. Briefly explain what versioning is and how it solves that problem. Mention GitHub.
   b. Write Readable Code:
      i. You won't ever be the only one to touch your code. Make sure other people can read it easily.
      ii. That means follow standards, style guides, and comment code when needed.

VI. **Conclusion**
   a. It doesn't matter how much you know right now. The most important thing for you to do if you want to be a programmer is start learning something programming related on your own time. It can be anything, but that's what's going to open the most doors in your career.

VII. **Contact & Getting Involved**
   a. Twitter & Email
   b. Introduce local developer groups and meet-up dates/times