I am not an IT specialist, but I am good at "Tech Stuff." Want to know my secret?

A few years ago, I discovered the biggest secret of "Tech-ie" people. By "Tech-ie," I mean the folks who use words like defrag and reboot. People who talk about IP addresses and go on rants about Big Data. You might be thinking. I'm not a "Tech-ie" person, so what does that have to do with me? Well, to you, I'll say, this secret is more extensive than you think. It's so vast that it covers "Tech-ie" and non "Tech-ie" people alike. By non "Tech-ie," I mean those who are unaware that a USB c port is very different from a USB port and those who yank out their flash drive without ejecting them first. This secret is for all of us.



The secret is Google.

You can use Google, Firefox, Safari, or Bing. Even Duckduckgo will do. The key is using a web browser to reach out and sort through the plethora of resources available online. Top professionals and non-specialists alike tend to Google things when they are unsure how to do something or cannot remember every step of a process. The main difference between Pros and Novice searches, in my opinion, is the specificity of the question asked. For example, if a novice and a pro wanted to know something about coding, a beginner might type into the Google search bar something like, “how do I code”.

On the other hand, a pro would type something like “how to use C# to call an action in Unity for 2d gaming” into the Google search bar. It is like the difference between Googling “how to learn a foreign language” vs. “What are some French words related to baking”. There is a lot of information out there, so it pays to be specific. An excellent way to get more specific is to create a plan. Take, for example, this coding-specific remixed communication plan.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Goals: | What are your coding goals? Why | How can you break your coding goal down into smaller parts? | What does the completion of your coding goals look like? | Using the smallest part of your coding goal what will be your first project? |
| Projects: | What does the completion of your first project look like? | How can you break your first project down into smaller parts? | What do you need that you do not have to complete your first project? How can you attain those/that thing? | What message are you trying to communicate with this first project? |
| Feed-Back: | Who do you most want to see your finished first project? Why? | What reaction do you want people to have when they experience your completed first project? | How can you attempt to ensure that people have the reaction you planned to your first project? | How will you be able to tell how people are reacting to your first project? |
| Medium: | How will your first project look? How is it displayed or how do people interact with it? | How will you create aspects of your projects look? Digital manipulation, physical creation, something else? | Will your project consist of multiple forms of interaction/display? For example, will you have a blog and a newsletter and a website or just one or the other? | Could you change the way people engage with your project? What difference would that make? |
| Milestones: | What tasks do you need to complete? Are some more important than others? Will that effect your workflow? | What accomplishments would you consider most important? How will you factor those into your project timeline? | Do certain aspects of your first project have a greater priority than others? How will you factor that into your project timeline? | How will you maintain your motivation for reaching your milestones in a timely manner? What would a reward system look like for you? |
| Engagement: | How do people interact with your first project? | Does participation with your project rely on some sort of action on your part? For example, do you need to continually create new content to keep people engaged or is your project a one off? | What encourages people to interact with your project? | Do you have a call to action for people interacting with your project? Is one necessary, why or why not? |
| Evaluation: | What does a complete and successful first project look like to you? Could it look any other way? Why or why not? | Can you express the success of your first project with measurements? | Once this project is complete how will it tie into your overall coding goals? How will apply what was learned to your next project and/or the reassessment of your coding goals? | What does the failure of your first project look like? What happens if you do not get the results that you want? |

The best way to learn about coding is to code. From my experience, learning how to code can be frustrating. For example, say you are coding, and after creating fifty lines of code you cannot get your code to work correctly and you don’t know why. To find a solution you spend hours reviewing your code and googling potential solutions. After all the reviewing and researching you realize that your code does not work because on the third line of your fifty lines of code you typed out a colon instead of a semicolon. After a few of those frustrating situations you may start to feel that coding is too much for you and want to quit. It may even make you begin to question why you wanted to learn how to code in the first place. Something I think can help when dealing with that level of frustration is the hopeful delayed gratification of producing something you think is fantastic—a fantastic something you are super excited about and invested in creating.

To that end, instead of just finding random coding activities online, figure out what you want to do with code. Or think about things you have interacted with that are made up of code that you like, like a mobile game app or an online chat room. What made you like it? What made you hate it? How could you refurbish that concept for your own personal uses? Use that insight as a jumping-off point to create a custom coding adventure. At this point, instead of just learning code for code's sake, you are learning code relevant to you and your coding goals. So, when things get super frustrating, you will be less tempted to throw your computer out the window.

 Now, how does one even begin to create their personalized custom coding adventure, you ask. Let's run with this example to see how to make a great plan.



For this example, we will say that you go on a hike through your neighborhood woodland area for an hour every morning. On your hikes, you take your journal and make notes about your experience. You write about what the weather was like, what exciting things you saw, how you felt about the hike etc. You don't like to reread through all the journal entries you have collected, but you would love to make something that you can enter some of that data into that will make it easier to see trends over time. You come to an Infinity Pi meeting at the Northeast Branch of the library one day, and when asked what kind of project could you see yourself coding, that data collection thing comes to mind.

With the goal: see if you can create this half-imagined data thing with code. We now need to break that down into smaller, more manageable pieces. Let's start with defining what we want to create because Googling how to make a “half-imagined data thing” is not very specific. Instead, it would be better to type “what is a data set” into Google. Using the facts that you know you want to deal with data that you have collected, and you know that there is a certain set of aspects from that data that you want to track. Data sets are an excellent place to start.

A brief reading of the beginning of the wiki page for data sets indicates a database. So now we know we will be creating a database. That leads to another specific Google search for what code is used to create databases. Looking at the search results page, you will see the word SQL in many of the titles and the mini descriptions without even clicking on any links. Now we know that our custom coding adventure will consist of SQL, even though at this point, we have no clue what SQL is.

 Next, type “what is SQL” or “how is SQL used to create databases” into your Google search bar. You could even search for “what coding languages are used along with SQL”. Googling any of those things is a giant and more specific step away from just Googling “how to code”. It will also make it easier to create a plan for your journey and keep your enthusiasm high.

When activating your specific Googling skills, remember to read everything you can find. Read articles, wiki pages, and even comments in forums. Speaking of forums, there are many that have users who are professionals actively working in the field, as well as other very knowledgeable people who often link resources in their answers. When you come across language or words you don't understand, don't get frustrated. Just look at it as a new caveat to your custom coding adventure. The more you learn, the easier it will be to develop your game plan for how you want to break down your overall goal into smaller bite-size pieces.

The process of creating those bite-sized pieces is creating your own coding-specific remixed communication plan. In your custom plan, be sure to make goals that are SMART.



S for Specific! Instead of saying I want to learn to code, you say I want to use MySQL because it's free and open-sourced to create a database to store my hiking observations data. I also want to learn PHP, HTML/CSS, and JavaScript to help build my database and an excellent web page to display my data.

M for Measurable! That means you want a way that, if it's not quantifiable, at least suggests an indicator of progress. For example, you can tally the number of hours you spend researching online for your project research. Then, you can use that number as a measure of how much you have learned when you talk to your friends about your project.

A for Attainable! That means making your project's scope something you can do right now vs. something you could probably do one day. For example, if you are not a runner, it would be possible for you to run a marathon one day, but not today. The attainable version of that is acknowledging that you want to run a marathon one day, and with that in mind making it a goal to start walking for 15 minutes daily. After enough of those 15-minute walks, you will have a new level of attainability, for example, walking for 10 minutes and running for 5 minutes. That's how you can stack attainable actions until that less attainable action becomes achievable.

R for Relevant! Aka, the whole reason you are learning MySQL, PHP, HTML/CSS, and JavaScript is that you want to, not because someone said it was a good idea. You are working on a specific project because it relates to particular things you enjoy.

T for Time-Based! For example, maybe you decided you will spend 1 hour a week researching and learning code. That means that if you keep it up in six months, you will have logged 24 hours of code/research. In your opinion, anyone who has logged 24 hours of code research should be able to create something out of code. Now you have the time-based goal that in 6 months, you should have something built. It may not be the completion of your goal project, but it would be something, and something is better than nothing.

Now that you know the secret get out there and start coding. If you want some company check out the Infinity Pi Coding Clubs Website, https://rcplne.github.io/. Infinity Pi consists of people of all ages who gather at the Northeast Branch to code together.

Happy coding!

