[00:00:01.22] - Speaker 1
Welcome back to lesson 3. In this lesson, we want to talk a little bit about AI and choosing AI and the processes for coding and processing and scripting with artificial intelligence. So artificial intelligence is AI, or AI as we call it. And you'll hear me use that, of course, already a lot through the— so far through this presentation set.

[00:00:32.22] - Speaker 1
AI is in the news all the time. It seems like every single time we turn around, we're hearing new things about AI. A lot of times AI has been, you know, deemed, put in a bad light, in a positive light. You know, it's taken jobs, then it's, you cause jobs, you know, cause job loss and layoffs because companies, tech companies say they're going to go through that. Other people, you know, talk about AI slop and how AI isn't as great as it should be.

[00:01:11.25] - Speaker 1
I'm not going to be here to really talk about the good and the bad of AI.

[00:01:18.14] - Speaker 1
I think that's something that you're going to make your decision about.

[00:01:23.12] - Speaker 1
If you want to use AI or not. Hopefully, if anything, you look at the— this opportunity gives you the ability to make a more informed judgment of how it might work for you.

[00:01:39.09] - Speaker 1
Um, so let's jump into this. So I want to show you the 4 major players that I've seen in the AI coding revolution, and that is, uh, first one, Claude Code. Claude Code is probably the biggest one that most of us hear about.

[00:01:57.22] - Speaker 1
It's done many— has been— is being utilized quite a bit. It's also got a lot of MCP integrations, which I'll talk about here in a little bit. And it can really build complex scripts. Cursor is another company that was recently bought by SpaceX, I believe. It is a, it is really a specific coding type of solution.

[00:02:32.17] - Speaker 1
It utilizes LLMs that are already pre-built from companies like Claude, but it really is focused on delivering a code-centric experience. And so it uses a, like, a VS Code, uh, experience. GitHub Copilot, it was the first out. Um, GitHub delivered one of the very first coding, uh, AI agents. It, uh, allows you to really kind of do some really good systematic, uh, coding, but some people have talked about it having some limitations, uh, when it comes to more complex process processes.

[00:03:19.04] - Speaker 1
And then Google has their Google Gemini, and that does have the ability to utilize the coding experience there. And it has a product called Antigravity, which kind of acts as an IDE native or a Visual Studio Code that allows you to write AI coding with it. So in all of these, I've used the Claude cursor in Google Gemini. I haven't really used GitHub Copilot, not really a big GitHub guy, uh, but I've utilized Claude cursor in Google Gemini. I actively use these 3 companies.

[00:04:03.29] - Speaker 1
I have subscriptions to all 3 of these, uh, services. So I'm actively using these 3 there. And you may say, Terry, 3? Well, you know, it, it's beneficial for me to be able to use Claude Code, Cursor, and Google Gemini. Google Gemini comes with my phone.

[00:04:26.08] - Speaker 1
Cursor I've been using for a while now, and Claude Code has just been an amazing kind of experience for a larger, more broad set because it doesn't focus just on coding. It has a larger context. Which I can hopefully touch on in a little bit.

[00:04:49.04] - Speaker 1
So, 4-step process for learning or for coding when you're using AI. So here are the steps of how I recommend you kind of take on this battle. So you're saying, Terry, what I want to do this, but I feel a little overwhelmed. I don't know what to do. And so these are the 4 steps I think are really good for you.

[00:05:13.00] - Speaker 1
First of all is to choose a programming language, whether that is like Kevin, you know, where he uses PHP, like me, which I use Python, JavaScript, you know, Go, Ruby. There are so many different other programming languages that you might choose. I choose Python because it's more of one of the most common languages. Out there. It's pretty simplistic.

[00:05:41.08] - Speaker 1
Uh, PHP is a good one as well because it allows you to run and create HTML scripting as well as just general scripts to run. So, you know, to me, those two are like the best. Um, and, and I can, I can code in PHP and I can code in Python. But, uh, what you want to do is look at what your skill set is. If you've used JavaScript or Python or PHP or whatever works, use it.

[00:06:10.25] - Speaker 1
Because I, the, the key is to feel comfortable with the coding because AI could write the code, but somebody needs to understand what it's doing. And you need to be able to read through it and scroll through each of those lines and go, okay, I understand what this does. I understand what this does. No, maybe this needs to be tweaked. Maybe I need to make this change.

[00:06:30.29] - Speaker 1
From a human perspective and I made a change in the code. So understanding the code and understanding what the programming was doing helps you ensure that you can do a quality check and evaluate and do an analysis on the code when AI spits it out. And then build a basic understanding. Read code flow, be able to edit or troubleshoot those things, things that you might find that are problematic and debug problems. And you can debug problems with your AI.

[00:07:05.17] - Speaker 1
There are many times where I'll be logging and I get an odd output and I'll put that into Claude or I'll put it into Cursor and say, your code, your script runs this. Where should I, what part of this should I start looking at if I'm just don't want to go through the 1,000 lines. You know, I want to say, hey, what we can look at. Then understanding what you can do within Blackboard using the REST APIs, because most of your scripting is going to be interacting with Blackboard's REST APIs. If you've not ever worked with the REST APIs, I do have a, I think, Jeff Kelly recommended or approved REST API presentation.

[00:07:54.21] - Speaker 1
You can use that link there, AT2023-REST, to go and attend that and go through that process. And then select an agent. So select one that fits your workflow and then just start experimenting with it, you know, asking it to do simple things. And your process and the general process for you should be how— this should be basically how— it should be a circular pattern of continuous improvement when it comes to AI-assisted coding. So your first step is define the problem, write the problem out in plain English, type it out in Notepad or type it into any type of document.

[00:08:46.10] - Speaker 1
What is supposed to come in? What's the input? What's the output supposed to look like? What might go wrong? If you can't explain it, you know, AI can't understand it.

[00:08:57.12] - Speaker 1
So you define your problem, you create it in that, and, and that helps. Then you break it into steps. So if I'm going to do something, I want you to authenticate into my Blackboard environment, then I want you to read the file, the CSV file that was provided to you in the call. Then I want you to use REST API to go find out information from the data that you find in the first row of the file. Then if that fails, if it

completes successfully, write this output or do this next step.

[00:09:36.07] - Speaker 1
Each step should be a testable environment. Each step should be logged and each step should be just, you know, a step-by-step process. It's the only way you'll get from point A to point B. Then you're going to use those steps and definitions to create a prompt that gives a very good— I would say, I almost say detailed, but sometimes you don't put too much detail because you want the code to— you want the AI to start make decisions on how to best and efficiently complete the process, but you want to give a genuinely good contextual description of what's going on. So that's a great way.

[00:10:26.14] - Speaker 1
Then you're going to do tests and identify gaps. You can run this on your test instance, check for failures, make sure that you have a logging feature where you can have output logged somewhere. And then generally your fifth option is just to iterate. So redevelop every single time. The script crashes when a user ID is null.

[00:10:52.04] - Speaker 1
So add a check, you know, this, and have the AI agents build it and continuously evaluate. One of the great things I have is that I have I, you know, I take the logs from a process that I run multiple times and I have the system take a look at it and feed the logs and say, here are the logs, here's the logs folder, uh, go through and evaluate the logs and make a, you know, output to me recommendations of what you think as an agent just from looking through the logs we should change and give me recommendations. And then if I like those recommendations, I'm going to say apply those recommendations to the to the script or to the program. And so AI can help you make those evaluations. It allows you not to have to do the deep dives of— and allows you to collect the data, allow the AI agents to collect the data, and allow you to make decisions of what features do you need, what features do you not need.

[00:11:58.16] - Speaker 1
And sometimes AI is going to catch it, sometimes AI is not going to, and you're going to catch it. So that's an important aspect. So got a lot of great things to talk about. And so I'm going to pause here and we're going to show you some real-world experiences for AI-assisted coding workflows in the next lesson.