EPISODE 18

[0:00:00.1] NA: Welcome back to La Vie en Code, a podcast dedicated to the self-educated web developer. I'm your host, Nicole Archambault and I'm so glad that you could join me for our final episode of EdTech September. If you heard last week's episode on games and gamification, I had talked about some of the great gamification features they added to the learning platforms I was using, that really kept me more engaged than any learning platform I had ever been in before.

One of those platforms that I had great success with was Treehouse, a very popular web development E-Learning platform. Now, Treehouse is very visual heavy, which I found worked wonderfully for my own learning. I didn't get a whole lot of visuals, most of us don't in school or college. I learned so much with Treehouse as a new web developer.

That I wanted to really dig in more to why exactly I was able to learn so much with this new information delivery method I had found. Compared to when I was a hot mess back in school with my learning anxiety. This questioning led me to the world of educational technology.

Which is why I've been sharing so much of what I've learned with all of you for this entire month of September. Although I feel like I know so much more now than I did back then. I still made a lot of mistakes along the way with my E-learning experience though.

I didn't properly leverage the tools that were given to me. I admit that much and I see it now. It makes sense. I mean, as I was learning about gamified platforms recently, it kind of made me cringe when I realized the huge gap between how Treehouse was probably supposed to be and designed to be used as a learning tool. Versus how I actually used it.

That's what I'm here to talk to you about today and it's the basis of the information that I'll be sharing with you. There is a proper way to use and maximize your learning with an E-learning platform like Treehouse, Free Code Camp, Udacity, Udemy, Coursera. I mean, there's so many of them.

New tools are popping up all the time and so I'm going to make it pretty general in my tips to keep them applicable to a variety of different platforms. Just consider personally whether or not they apply to you and your current situation.

Or, if you haven't even started learning the code yet, no worries, you can actually leverage this information out the gate, starting today and really super charge your learning.

Faster learning means shorter career transitions, shorter career transitions means less energy spent trying to find a job. As well as a chance to take your learning to the next level while getting paid even sooner.

I mean, does that sound like a good goal? Because it sure does to me. You'll leave this episode today with some solid advice to apply to your E-Learning today as soon as you finish listening. Okay, let's go.

[EPISODE]

[0:03:23.1] NA: Alright. I'm going to jump right into the tips here so we can get you along and moving with your day. The first tip is to approach your education with a growth mindset not a fixed mindset. If you're not familiar with the concept of a fixed versus growth mindsets, here is kind of your 101 crash course.

A fixed mindset believes that our intelligence, our character, our creative ability are all static features and that we can't change them in any meaningful way. A learner with a fixed mindset would believe that when someone else finds success, it's just an affirmation of their inherent intelligence.

As a result, that learner avoids unknowns at all cost due to a fear of failure. The name of their game effectively becomes clinging to what they have, which can also create interpersonal relationship issues. Whereas an individual with a growth mindset as supposed to a fixed mindset, thrives on challenge and really sees failure as a pathway to learning and in the long term, success.

A learner with a growth mindset isn't afraid to stretch their existing abilities and learn by both their successes and their failures. Why do people develop fixed mindsets to begin with? It turns out there are a variety of reasons for this and I've included some links in the show notes to help you explore that idea further.

However, you should know this. People who have difficulty learning or understanding a topic generally just lack context and context is only created when we're able to relate a topic to something that we already know.

One person is not necessarily "smarter" than another. Now, the smarter person has actually just put in more effort to create context or they naturally posses the context by which to connect the idea. The process of gaming context can be easy or it can be hard for somebody. Once you gain that context, you still may not find it easy to recall the information without really deliberate practice to commit to your long term memory.

As human beings, we're capable of learning anything that we desire to learn, we're built for growth, we're constantly growing and changing. The problem really is that we're generally not aware of when we're growing or having a growth moment.

Turning information into personal growth rather than just information is a skill that needs to be practiced regularly. It's not going to be comfortable at first but it's necessary to practice alongside your E-learning.

I've made it my personal mission to become more aware of every growth moment that I experience by exercising a practice called mindfulness. When I figure something out, I take a moment to be present in the moment and savor it as if it's something that I didn't know and didn't have before.

Rather than just you know, rushing on to learn the next thing. Awareness of and gratitude for growth also releases dopamine which results in happiness. Learners have to have a growth mindset in order to get to the point where they can begin putting new information in their head.

[0:07:11.1] NA: Believing that every obstacle that you encounter is you know, a result of your own personal limitation is harmful and it's not going to get you very far in the web development industry to be brutally honest.

Those beliefs are only going to inhibit your ability to grow because you sub consciously seize asking yourselves the questions and seeking the answers to those questions that lead to an improved version of yourself.

If you learn before this mind shift to a growth mindset occurs, you may find a lot of anxiety and avoidance in actually applying the skills that you're learning and building projects to add to your portfolio. You may just even be going through the motions without really seeing yourself combining all of those skills into a job role.

Because you don't truly believe that you're actually capable of getting there. This is a big part of why so many web development E-learning students fly through their curriculum and then you find yourselves at a big project and have no idea what to do.

I'll raise my hand there too, that was 100% me at virtually every step and now I figured out how to move past it. Furthermore, if you never find yourself with a big, daunting project that test your limits, if your E-learning platform doesn't include something like that. You may just never get yourself to the point of building and creating it all.

Which would drastically hinder your education and the time to get a job offer. These are all things to consider. This is why applying the things that we learn is so important. Application is exposure and repeated exposure leads to long term memory improvement and behavioral changes.

In short, before you begin learning with an E-learning platform, ensure that you're first committed to growth and your brain is onboard. If you take absolutely nothing else from this episode, that's why I wanted to put it first.
Start by taking 100% responsibility for your education and ensure that you're not wasting your time by reading more about fixed and growth mindsets in the episode show notes at

lavieencode.net/18. Alright, moving on, that's a really deep one so I wanted to really expand on that.

Tip number two is to choose the right platform. This one's pretty self-explanatory but I mostly want you to understand what I mean by right. As we know, there are lots of E-learning platforms for web development out there to choose from.

By platform, I mean, an entire app, you know, a solution, a packaged solution that will help you deliver results in your web development education. My first point of advice here in choosing the right platform, is to go by your learning style.

Be sure to pick a platform that doesn't make assumptions about what works for your preferred learning styles. If you learn better with image and video heavy resources, choose a platform that's primarily visual or video heavy like Treehouse.

IF you're distracted by images and you just want text or audio, go with a book, an audio book or a podcast. If you know you'll want to explore after learning the basics, use some guided tutorials to apply the skills that you're learning and build projects out of your portfolio.

The next point that I want to focus on is don't do simple. What I mean by that is that you should really work to avoid platforms that refer to the process of learning web development as easy or simple or anything related to that.

Why? Not only is learning web development not easy, but it's actually quite difficult, statements like something as easy or simple assume a lot about your background. Using an E-learning tool only provides you as much as you put in. You'll be required to put in a lot of work to achieve your goals.

You want a platform that both recognizes the fact that web development isn't easy and accommodates it with relevant elements of educational technology, that will help aid you in the process of learning.

There are even Chrome extensions out there that will scan a web development tutorial for such assumptive phrasing like “easy” or “simple” and it came as a result, funny enough of a free code camp founder Quincy Larson.

He made a call to his students to create a tool to help learners identify those types of resources more quickly. He linked to some of them in this great article that he created, *One Does Not Simply Learn To Code* which is a great read in general.

[0:12:20.4] NA: you can check that out in the show notes as well at lavieencode.net/18. Alright, next point in choosing the right platform is to pick something that teaches java script well.

Now, this is opinionated obviously. I strongly suggest that you have one go to platform that includes at least the basics of html, CSS and java script. Most platforms will probably go well beyond that but unless you’re at that point, I don’t recommend worrying about it early on.

The reason for recommending java script is this. Java script is incredibly popular, it is also an excellent starting point for both front end and back end developers. As well as people just wanting to learn some programmatic thinking and basics of data structures for their none programming job.

A platform that teaches java script effectively will actually also effectively teach you all of the following. Developer mindset including approach, problem solving, data structures, control flow statements like loops. I mean, the list goes on.

[0:13:35.8] NA: All of that will be transferable to any other programming language that you encounter in the future. Treehouse and Free Code Camp have excellent java script programs with Free Code Camp being a good deal more hands off with their challenges after introducing you to the basics.

I use bother personally but started with Treehouse and migrated to Free Code Camp once I found myself wanting a little bit more independence and freedom. When you choose the right platform for your needs and not just what you think is the hottest out there.

You'll feel better supported in t hose critical first months. If you want to learn more about what makes a great E-learning resource, check out episode eight. 10 Things You Need To Know Before Learning To Code. Let's move on to tip three now.

Know that not every instructor is created equal. Think about your favorite teacher ever, that should bring up some warm fuzziness. What did they do in particular that helped you learn. Now, recognize that not every teacher is going to be your favorite teacher.

A lot of them know content but are unable to teach. As I've quoted before, teachers can't geek and geeks can't teach. We're a small group that sit at the center here. People who develop and teach E-learning courses are as varied as the topics that they cover.

Find an instructor who emulates those traits that you best connect with. Even if it's a YouTube video. The person behind the video is what matters and how they deliver the content in a way that you can understand it.

If you respond well to humor for example, find somebody who is funny and if the instructor isn't a good fit for you, you can always find another resource. Just don't give up as a result of not feeling like your instructor is a good fit for you.

There are plenty of other options out there, reach out, you know? Don't isolate, find somebody that can suggest something for you to move on to, that's a better fit for you. Tip number four.

Ask yourself questions while learning. Inquiry based learning is super powerful. We learn most effectively when we ask our own questions and find our own answers and questions can bubble up from other questions, information you encounter, the opinions of others, anything, really. Before attempting to learn a topic, consider how the topic relates to other things you already know about.

While learning, continue to ask yourself questions to help you make the connection until it finally culminates in that wonderful "aha!" moment for you. Here are some questions that tend to lead me down the proverbial rabbit hole. I ask myself, what do I need to know to be able to

understand this topic. What do I know that's already close enough to this concept to contextualize it.

Try Googling that term and the term that you're looking to understand. Someone may have already written on it, maybe making an analogy, making a connection, you just want to find something that's close enough. Questions will help you get there.

You also probably want to know how much information do you need to know on this topic to get a job in your desired field. Not every resource out there is going to be aware of your goals unless you're working directly with them.

You need to have in your mind what it is that you need to know. Otherwise, you might just end up learning a lot of information that you don't need to know in order to get a job in your desired field.

Ask yourself too, "Does this particular E-learning platform include enough deliberate practice on the topic that I'm learning to help me reinforce this new behavior or knowledge?" Asking and answering those questions early on helps you to stay mindful in the moment.

And identify the moment your learning isn't optimal any longer. Tip number five is to allow yourself to make mistakes. Mistakes may not always be allowed in traditional classrooms especially when there's grading involved but that's not the case here.

For a lot of people, this is really unfamiliar concept like how are you supposed to progress if you make mistakes. We've heard that practice makes perfect since we were little kids. But often times, people forget that practice itself is not perfect.

When we make repeated mistakes while being aware of the mistake and the desired outcome that we wanted, our chances of both deeper understanding and improvement are increased over time. When you make mistakes, remember that other new web developers aren't necessarily doing it right either.

Also, you have an opportunity to share your experience with the mistake and help others in the process. You know, you could even use that opportunity to create a blog or a podcast and build a valuable asset for your future job search. Allow yourself to make those mistakes, they're going to help you in the long run.

Tip number six is to create and maintain a routine. Consistency is really key when you're using any kind of e-learning platform. However, it's way more difficult to be self-disciplined when we're not required to be present in a classroom. Routines keep you thinking about your coding education every day. To the point where learning is no longer a conscious decision.

Many gamified e-learning platforms out there track activity streaks, which can provide significant incentive to become and stay consistent with your learning. If you miss a few days, don't be a damn perfectionist, I say that as a damn perfectionist.

Just pick it up and get back up to it. Learn from the experience. Why you dropped off in the first place and address them or you're bound to repeat that cycle. You know, ask yourself, "Did things just get busy at home or work?"

Did you get discouraged or frustrated, are you not committing enough time to your education or are you committing too much time and burning out. Are you getting bored perhaps? Set one particular time each day, ideally when you'll work on your web development education and stick with it.

Your mind will start automatically reminding you and pulling you toward that. By announcing it to other people, others will actually take notice and leave you alone too during that time. Also, be sure to have a backup plan for each part of your routine too.

You know, what happens if you get stuck? Are you going to walk away and never come back? What happens if something comes up and you need to shift your time to another time. What will ensure that you still get it done for the day?

When will you consider your education complete for the day and actually give yourself some rest to switch to the defuse mode and process it all? Back up plans make it easier to get back to what you were doing if you get distracted or you fall off.

Distractions happen. This is a valuable skill that I recommend you practice. Tip number seven is to be aware of and prepared for the beginner hand-holding to end. This is a tough one and I'll repeat it. The beginner hand-holding is going to end and you need to be aware of and prepared for it.

So a lot of new E-learning students don't realize coming into web development for the first time that there's a drop off that happens with web development and general programming education, unlike many other industries out there. The programs out there are generally best equipped for helping students learn the basic principles of web development but not as well equipped for the more complex topics.

So once this happens, you may need to deviate away from the platform in order to find what you need and to avoid all the confusion that comes along with facing this drop off and the emotional anguish that can happen, you need to be aware of it and observe when it begins to happen. You may feel more anxious, you may feel unsure of yourself compared to the confidence that you had before.

You may become really easily distracted as your attention and your focus is just wondering as a result of becoming more disengaged. You might begin questioning yourself whether or not web development is actually a good fit for you or you're a good programmer and when you think about returning to a project that you are working on, I really experience this deeply and frequently.

You might find yourself thinking of literary anything else to do instead. So all of this is totally natural but you need to know how to work around it in order to successfully leverage your E-learning platform in general. They all have an area that it's going to just drop off. So this gap it typically occurs along the advance beginner threshold.

So right after you've gotten through the basics and you're getting pushed off, you're being pushed out of that beginner nest technically and it needs to be supplemented. Your education needs to be supplemented at that point with other resources. Sometimes it even requires an entirely different approach than the approach that you were taking to learn those basics.

So just be aware of the face that this happens, be prepared for it, don't blame yourself when it does happen and all of this really ties into my next point too which is tip number eight. To allow yourself to explore and find the information that you need.

[BREAK]

If you're not understanding a topic once it starts to get difficult especially you are going to get frustrated. This is where E-learning strengths actually lie though. Take advantage of the forums that are offered, let other people explain. Identify to people what you do and don't understand specifically about the problem. They will always appreciate this.

Psychologically, it's like reaching for a grip on a climbing wall but just not quite giving it enough power to make that connection. Allow yourself to explore some other options too as far as materials. Videos, books, online forums, chats with other developers, that's totally valid. Something will click but you again need to allow yourself to explore and not keep yourself constrained to that one platform.

Just because you don't feel confident going out to find other resources, people can hook you up. So think about the things that you didn't understand, even a couple of months ago, that you do now. Just work hard to make that connection and then reinforce it. Tip number nine is to take frequent breaks, this should be a happy point for you because it's pretty self-explanatory.

But a lot of students have difficulty getting back to working once they take a break, as I mentioned before, so they avoid taking breaks all together. We need breaks in order to learn and this is why: our brains actually cycle through focus and defuse modes. When we are actively learning and absorbing information we're in the focus mode.

But the defuse mode occurs while we're resting, sleeping or just generally focusing on other things that aren't that topic or learning actively. So be sure to take the breaks that you need and focus on other things because that's like filling your funnel up and then giving it a chance to drain out before you put more in otherwise it is going to overflow.

You are not going to remember thing and you are going to get frustrated. If you are a procrastinator or you are easily distracted like I am and is such not likely to come back at all after a break. Serially practice it and this is what I mean. I had some great success using the Pomodoro Technique, technically using an app called Be Focused Pro. I will link that in the show notes as well.

So I set it to 15 minute increments with two minute breaks in between. I found that more often than not with the 15 minute and two minute set up, I wanted to continue working through that two minute break. Eventually, I increased it to a five minute break I think in 20 minutes of working instead of 15 and I mostly wanted to work through the break still but at this point, I had to remind myself that I need those breaks.

So I took them and it was admittedly harder to refocus the longer the break was. So I think about five minutes is sweet spot for me and then if I have to recover it takes a lot more energy from like a 30 minute break or something.

My brain just no longer fights me the way that it once did though when I am trying to get back to coding and it's a result of direct practice. I do have one other piece of advice on the topic of breaks, no TV or other heavily engaging activities or conversations unless you really want to challenge in a fight getting back into it.

Another stage piece of advice is to take a break while you are doing well at whatever you are doing as oppose to when you hit an obstacle. When you know that you are enjoying coding you're more likely to want to come back to it. This is just basic human psychology and a little brain hack for people who think too much, like I do.

So tip number 11 is to use visual learning controls wisely. Each of the controls on the videos that we watch on some of the more visual heavy learning platforms like Treehouse, play a certain

role in E-learning compared to just watching a movie or a TV show. The pause functionality on videos allows you to stop whenever you need to.

You can walk away, you can take a break, you can just stretch, you can take a deep breath, you all we can't do this in traditional classrooms just that one button is one of the most powerful features of educational technology and E-learning in general. Close captions also are not just available for accessibility but they also allow students who are both visual and auditory to process a video in not one, but two, different ways.

Now you need to be careful with this and your mileage may vary because some people really cannot process two different forms even if it is the same information. So if you know that then don't use the close captioning or you can turn off the sound and turn on the close captioning so that you just read it. So speed controls even can be used to speed up or slow down the lecture's pace.

And that can help us move more quickly through familiar content as well as slow down newer content for better understanding or like I said before, you can turn off the audio entirely and just simply read the close captioning if you're distracted by audio. So when you think of these features as tools that you can wield to control your learning experience and environment, you can really power through a lot of material in a very short period of time.

More importantly though, you'll retain the material better and you'll be able to apply it to the "next levels" of your E-learning experience. Tip number 12 is to watch or read the material multiple times. While it's not exactly deliberate practice, repeated exposing yourself to the same material plays a very important role in improving long term memory.

But don't confuse this with wrote memorization. The goal in this is not to simple memorize the information but to develop a deep understanding by gaining a bit more context with each pass. But how do you know when it is time to watch a video or listen to a podcast episode or read a chapter again?

Well after I encounter material for the first time, I first try to recap it in my mind afterwards. If I can't fully explain the material including being able to recall or remember it, if I am just fishing

and not coming up with it even though I watched it and went, “um-hum” it’s time to re-watch or revisit. In the second pass, I’m then subconsciously looking for the material that I couldn’t recall before.

Which results in kind of one of those “aha! moments” when you encounter it again. That “aha! moment” seems to trigger your brain back into focus mode and really work to form deep context this time because you’ve identified it as something that you don’t know but should. If it doesn’t happen that time, try it again and then pause and maybe branch off.

To find some other resources like I mentioned in the previous tips, to explain the topic in a way that you’ll be able to understand and deeply contextualize. In short, don’t think that just because you encountered information once that you’re fully retaining or even understanding it. Tip number 13 and we are almost done, there’s only 15 here, is don’t spend too much time on practice problems.

By practice problems, I mean those little end of lesson or mid-lesson questions and challenges. Don’t get me wrong here, they’re super important and practice problems are actually a great way to commit material that you understand to long term memory, for easier access in the future but they can also give a false sense of accomplishment.

Much like with the things we learned in secondary, and you know high school, earning a high score on a test does not necessarily mean that you can apply the information on that test and that you are learning in the real world. And that’s actually exactly what we are trying to do here, isn’t it? We’re trying to get a job in the real world with the skills that we are learning.

So be mindful of when you are spending too much time artificially inflating your sense of progress, by simply encountering lots of information without using it. I recommend splitting your time evenly at first at least between practice and application. This does seem to be a bit of a golden ratio and it gives equal focus to learning skills and on using those skills for maximum output.

The sooner that you get used to firing up your code editor, your terminal, your browser, wherever you write code to write actual code and apply what you know, the sooner you’ll have

projects to show for your experience which results in a valuable portfolio. We all want that valuable portfolio but why do we avoid it then? It's craziness.

You'll also be able to run into real world problems with your code and need to communicate with other developers in order to figure it out. That's a huge skill, it teaches you how to work with other developers to solve your own code problems. You'll also learn to gain confidence because one of the anxiety points that I most frequently hear from others, learning web development, is that they're not able to confidently use the skills that they are learning.

Right? Sound familiar? So be part of that small percentage of web development students that are really hitting that golden ratio, of skill practice and application. Moving on, tip number 14 is to intentionally design your primary study space. Now you can put up motivational quotes, images to help remind you why you're learning these new skills.

We tend to forget what we are learning for while we're learning. When you are in the middle of it, you forget why you're doing all of this activity. You start going through the motions and that's harmful to learning itself. So make your space comfortable, make it inspiring, make it motivating. You are not in a classroom anymore with people all around you in a desk that a thousand other butts have been in.

This is your space and you can create and curate your own space. That's where you're going to control your future. So don't study where you sleep either or you will definitely find more difficulty sleeping there. Your brain will begin to associate it with problem solving activities and not sleeping. I did a lot of Treehouse in bed and I found that it was very difficult for me to get to sleep afterwards for months.

So please, heed my warning there. Finally, tip number 15 is to use any and all social resources offered by your E-learning platform. Remember that you decided to take your learning into your own hands so it is going to require a slightly different approach than what you were doing before.

Isolating yourself in your education may work for a little while but it's not sustainable. E-learning platforms that you use may not have a lot of social tools embedded in them and if that's the

case, you'll need to find your own. It should be a consideration though when you are looking for a platform. You need somewhere to be able to connect and really talk to other developers. That are going through what you are.

So drive yourself to get the most out of the social resources that are available to you that you can. That includes forums, it includes Slack or Gitter channels, it includes Facebook groups direct communications obviously with other more experienced or newbie web developers. You know I learned solo for a long time and I neglected the social parts. Because I think I was just afraid that I wasn't up to par with other people.

It ultimately really hurt my education because I wasn't getting the perspectives and learning from other people. So you'll encounter problems that you just can't solve on your own and you need some help with and although you know, I didn't bite anybody's head off for offering code advice, I also had to get used to accepting help as well.

And learning to apply other people's advice to my own situations. Make sure that you ask questions without fear of being considered ignorant because if you don't ask you risk not understanding and that's going to hurt you far more than a couple of trolls and meanies on the internet and it will only create more anxiety for you when you feel like you're behind. So I recommend that you get comfortable with talking about your learning struggles openly and you start talking about them early on.

[OUTRO]

[0:39:22.1] NA: Thank you so much for joining me for EdTech September. I hope I'll be able to make this a regular annual theme month, especially since September is my birthday month. It's on September 30th. I really, really enjoyed putting these episodes together and there should be some really great future guests from the educational technology industry as well that I just wasn't able to fit in for this particular month.

This was just a first go, so I know it will get even better. I love to hear your feedback on how you enjoyed this month's episodes, good or bad and hopefully good though. You can drop me a comment in the Facebook comment section at the bottom of the episode page located at

lavieencode.net/18. You can also reach out on Twitter @lavie_encode and be sure to like and follow the La Vie en Code Facebook page at Facebook.com/lavieencodeblog.

If you enjoyed the podcast as much I enjoyed creating it, please be sure to leave a five star review on your preferred podcast player. Next week, I am going to shift it up and share some of my personal experience, specifically my story of going from, oh wow, unemployed newbie web developer to employed web developer to self-employed freelance web developer to tech entrepreneur.

So I find a lot of folks out there don't actually know what I'm doing. They just see me out here doing the most, they don't know what my deal is, so I am going to do a bit of a re-introduction to Nicole 2.0 for my newer podcast listeners especially and discuss why I decided to pivot away from programming temporarily to become a tech entrepreneur and why I'm not creating my first online course.

My journey has really been kind of an interesting one that's for sure. So I am trying to learn to be fearless right now, so my hope is that by sharing these experience with you all it's the first step in the right direction and so you'll definitely want to catch episode 19 next week. We're almost to 20 oh my goodness. Until next time my friends, peace, love and code.

[END]