Now that you have a solid foundation

of JavaScript, I want to talk a little bit

about the actual process of learning how to code

and give you some motivation.

I think this is an important aspect of a course like this,

but everyone else simply seems to overlook this topic

for some reason.

So let me share my advice on how you should learn to code

as a complete beginner in this video.

And to make this a bit more fun,

I will start by telling you how to fail

at learning how to code.

And to do that, meet John.

So John has been learning how to code for some time now,

but he is failing miserably.

He's just not going anywhere.

I guess, he is just not receiving the right advice.

So let's see what John is doing wrong so that you can learn

from his mistakes and learn how to code in no time.

So the first mistake that John made

was that he didn't have a clear goal at the beginning

of his learning journey.

He simply started by watching some courses

and reading some tutorials,

but he would just blindly copy the code

without caring how it works.

Even worse, Sometimes he would just copy and paste code

from tutorials or sites like Stack Overflow.

While learning, he didn't reinforce what he was learning

in the courses and tutorials by doing small challenges

or taking notes.

He also didn't practice coding outside of the courses

and he didn't come up with his own project ideas

to improve his skills.

Then whenever he did code a bit,

John would quickly became frustrated whenever his code

was not perfectly clean or efficient.

John also lost motivation pretty quickly

because he thought he could never know enough.

He could never know all there is to know

and this is a really big mistake.

Also, he was just always learning in isolation

without sharing his process.

Finally, after finishing a couple of courses,

he thought he was now a web developer

and could start applying to jobs,

but no one would take him because recruiters soon found out

that he couldn't even build a simple app on his own.

So that's a long list of mistakes,

but it's not too late for John.

We can fix all these mistakes, right?

So let me show you how.

So the first mistake is relatively easy to fix.

What you need to do is to set a goal for yourself.

Think about this and actually write it down

on a paper or something, make this goal specific,

measurable, realistic, and time-based.

For example, your goal might be,

I want to become a front-end web developer within one year.

From there, you can then develop a plan on how to get there.

It also helps to know exactly why you are learning to code.

For example, are you trying to switch careers

or do you want to find a better job

or are you just starting a new hobby?

Whatever it is, it's important to be aware of the reason

why you're doing this,

because this will help you stay motivated

when learning becomes more and more difficult.

To stay motivated, it's also great to imagine a big project

that you want to be able to build

by the end of your learning journey.

This has always helped me tremendously.

With this project in mind,

you just need to research the technologies

that you need to learn and then learn them one by one,

step by step in a calm way.

Next up, instead of simply copying code

from a course or tutorial,

always make sure that you really understand the codes

that you're learning and typing into your editor.

Don't move on if you don't understand.

Also, always actually type code.

Don't just copy and pasted from a tutorial

or a site like Stack Overflow.

This way, you're training your brain

to remember the syntax letter.

Reinforcing is one of the most

important learning techniques.

It means that after you learn something new,

like a new JavaScript feature or a concept,

you use it immediately to make it really stick

in your brain.

You can do this by taking notes and by challenging yourself

with small coding exercises and challenges.

For example, let's say you just watched a lecture

about functions.

So now you know what functions are

and you saw some examples.

But now as a next step, you need to reinforce this knowledge

by writing a function for your own personal environment.

For example, you might need to calculate the average

of your university grades,

or you might need to convert recipe ingredients

from ounces to grams while cooking.

So just write a function that does this task for you

and have some fun with it.

Coming up with your own ideas like this and solving them,

will absolutely boost your confidence

in your programming skills

and really take your time with this.

There's absolutely no need to complete any course

as fast as possible, all right?

Now it's absolutely normal that in the beginning,

it's hard to come up with your own challenges.

That's why I included countless coding challenges

in this course for you.

You can also use a special challenge platform

like codewars.com,

where you can find thousands of coding challenges

for different skill levels for free.

I have actually solved some of Codewars challenges

in real life with some students and I can tell you,

it's a ton of fun and really helpful.

One of John's biggest mistakes was to not practice coding

outside of the course environment.

But practicing on your own,

is the single most important thing that you have to do.

And I can't stress this enough, okay?

Coding your own projects without any course guiding you

is really not optional.

If you just follow courses and don't code on your own,

you will never ever be able to write programs on your own.

So come up with your own project ideas,

no matter how small or big and just build them,

or you can copy parts of popular sites or applications.

No matter what you do, really challenge yourself,

make it difficult and grow by overcoming obstacles.

And then now it's scary to leave the comfortable environment

of a course where you just have to follow the teacher

because you will fail and you will make mistakes,

and it will be difficult to code on your own.

But getting out of our comfort zone,

is the only way we grow.

So don't be stuck in tutorial hell,

which is exactly the state of not being able

to leave tutorials.

There is more about this whole journey

at the end of this lecture.

Now, when you practice, you should never become frustrated

with the quality of your code.

Do not get stuck by trying to write a perfect code,

which is really clean or really efficient.

So none of that matters in the beginning.

You're still learning.

So just write as much code as you can,

no matter the quality.

Really, clean and efficient code will come with time,

all right?

It's just natural progress.

Also, you can always come back and refactor

and improve code later if you want to.

Another important piece of advice I can give you

and I actually give it to myself from time to time,

is to embrace the fact that you will never know everything

there is to know in web development.

Web development is this huge, huge field.

And so it's completely normal and acceptable

that you will never know everything about web development.

So don't put too much weight on your shoulders

and create too many expectations.

So in fact, the best developers and teachers

have no problem in admitting

that they do not know everything

like the legendary Kyle Simpson,

who admits that even after 20 years,

it's still a struggle to make his code work

and understand it the next day.

So do not lose motivation if you feel the same way.

Also, do not compare yourself to top developers

who do know a lot more than you,

because, of course, they do.

Many of them have been at this for over 10 years

or maybe even 20 years,

but nobody was born an expert and even experts

are still learning all the time.

Now here, it's actually pretty important

that you have a well-defined goal for yourself,

because if you do,

you can just focus on what you need to learn

in order to achieve that goal.

Everything else is unnecessary, it's just nice.

For example, if you don't need a backend in your app,

then you don't need to learn Node.js or PHP, okay?

The next mistake that is hopefully easy to fix,

is to not learn in isolation.

One great way to do that is to learn together

with other people, no matter if in person or online.

Whenever you learn something new,

explain it to them because explaining things,

forces you to really understand them first

and repeating a concept will then really

make it stick in your brain.

That's why many people say that the best way of learning

is to teach and I actually share that opinion too.

Another great technique that many people like,

is to share your goals publicly,

for example, on a social network,

to make yourself accountable.

You can also share your progress on social media

to motivate yourself and others.

For example, on Twitter,

there are some popular hashtags that people use,

like #100DaysOfCode, #CodeNewbie, and #weddev.

If you use these, you'll get feedback

and a boost of motivation from other people

that are on the same journey of learning to code.

It's a really nice and friendly community,

at least on Twitter, which is what I know and use.

So for sure, you should try it out.

And finally, the absolute biggest mistake that John made

was that, after he went through a couple

of web development courses,

he thought that he was now a web developer,

ready to take a job.

Now, this is probably the biggest misconception

that people have.

Just take a couple of courses and you're done,

but no, that is just not how it works.

So I'm sorry to bring you the bad news,

but courses like this one can only take you so far.

They will give you an amazing foundation, of course,

but they're just the beginning of your journey.

You need everything else that we just talked

about in this lecture in order to become

a successful web developer.

So let's talk a little bit more about this

and the journey of actually becoming a job-ready developer.

So here is a very true graph that shows you

how your confidence in your skills

changes as you become more and more competent

as a web developer.

And this is useful because it allows us to plan a strategy

for your learning journey.

So when you start, you are in the,

everything is awesome phase.

You are studying one or more or tutorials,

understanding how the code works

and taking a lot of challenges

just like I mentioned earlier.

Now you're not really competent yet,

but are starting to get confident

and are pumped to keep going.

So this is probably where you are right now

and will be throughout this course

or any other course that you take.

But then as you start practicing on your own,

you reach the cliff of confusion.

Here you realize that coding might be a lot harder

than you thought when you no longer can just follow a course

with a predefined path.

So your confidence is gonna start to drop a little bit

and you might realize you can't build anything meaningful

on your own yet.

But don't lose motivation here,

just keep writing lots and lots of code on your own

no matter how much time you need to spend fixing bugs

and searching for solution.

It's all part of the journey and it's still gonna be fun.

Slowly, you will become better at writing code,

so you will become more competent,

but your confidence might drop even further

because now you start to realize just how much more

you will need to practice until it can build a real

application on your own.

You are now in the so-called pit of despair, but believe me,

it's not as bad as it sounds.

You might start to feel lost,

but a great solution for this is to start learning

with other people like other beginners

or even other developers who might mentor

and help you along the way.

To leave the pit of despair,

you must keep challenging yourself.

You cannot stop here, okay?

So keep building bigger and bigger projects,

run into lots of problems and keep fixing them.

You really need to have this mindset of fixing problems

in order to become a web developer.

It's what we do all day long basically.

At this point, it's also important to keep in mind

that learning how to code is a really hard thing to do.

So don't beat yourself up for finding it too difficult

and for being stuck all the time.

It is difficult indeed, but it's gonna be worth it.

And trust me, we've all been in that place

and everyone who successfully works in the industry today,

just start out the same as you right now.

So really, congratulations for even starting this path.

It's already a great achievement in itself.

Now you just need to continue building stuff.

If you want to know about me personally,

I learned to code by starting projects

that were actually way too difficult for me at the time,

but that made them real challenges

that made me grow tremendously.

For example, I tried and failed

to a football simulation game with Visual Basic.

Then, I build a Content Management System in PHP

with really bad quality code, but it worked

and I actually sold it to a client

and made good money.

I also accepted some client projects for websites

that I didn't know how to deliver,

but I figured it all out in the end

and it made me grow so much.

And I'm sure that the same is gonna happen to you too.

Anyway, back to the journey here,

at a certain point, you will start to see the light.

So at this point, you are getting quite good at writing code

and your confidence will finally start to grow again.

You are now in the back to awesome phase.

Now you can either start learning new technologies

or you can start rounding up your skillset

with best coding practices and tools

that you will need for jobs,

for example, like version control, testing,

task runners, et cetera.

If your goal is to find a new job,

you're probably starting to be ready to apply for jobs.

You need many things for that, like a really good portfolio,

but that is a topic for another day.

Just be aware that even when you're working,

you will never stop learning.

So you're never really done learning how to code.

Web development changes really fast

and so you will need to be constantly adapting

to new technologies.

All right, and that is basically the typical journey

from start to job ready.

Now please, don't be scared by what I just told you.

I'm telling you all these things not to scare you,

but to prepare you because I really want you to succeed

on this path and I want you to achieve real success,

not like fake success.

There are just too many people out there

who will tell you that coding is easy

and that you will magically become a web developer

in two months if only you take their course,

but this is simply not true.

So I'm not gonna sit here and tell you

that coding is easy.

It's not easy.

But with the right attitude and the guidelines

I showed you in this lecture, you can totally do it.

And I mean that because I've seen it happening

time and time again.

So many people have learned how to code

from complete scratch, switched careers,

found better jobs and improved their lives dramatically.

And I mean many of my students and also many other people

that I know personally, it's real,

and you can achieve the same thing.

Just keep going, keep coding and keep building.

