```
print('Hello, World!')
```

Thank you for purchasing the lesson.
We will be getting in contact with you shortly!

For now, I have something awesome for you…
I have helped many of my students with this e-mail,
hopefully, you would find it beneficial for yourself…

Use this to start mentally preparing yourself to code and for the lesson.

In this lesson, we'll be talking about the most important step that you need to take before you even think about starting to code: **mental preparation**.

No, I'm not talking about positive thinking—you already know that.

What I am talking about is how you **need** to train your brain to start thinking correctly before you start coding.

The success of your programming career can depend, in a large part, on how well you do this.

But, I don't want to put too much pressure on you.

Too many people never end up starting to program because they are stuck in small things like deciding what computer programming language to start with.

By far, it would be better to choose a less-than-optimal programming language for your career, than it would be to not pick one and never get started in the first place.

1. Semantics… NOT Syntax.

The best way to head-start your career is to pick a programming language and then stick with it.

Let me give you an example—one that you'll hear from me a lot—that helps illustrate this point.

Imagine if someone came up to you who did not know any human languages and asked you (Assume they were able to express their ideas to you somehow).

"What language do you think I should learn first?"

What would your advice be to them?
Would you tell that person that any one language is best?
Would you tell that person to learn a little English, then a little Spanish, and then a little Chinese?
Would you tell him, or her, to learn Japanese because the translators

are getting paid a lot?

No, of course not!

Your response to that no-language-speaking-human-being—who you could somehow communicate with—would be to pick one language and **stick with it.**
You would also tell this person to **NOT** switch the language until they got very good at it.

The reason why as a human we need to learn a language is not so we can live in a luxurious city.
We learn a language so we can **express ideas.** THAT is the idea behind learning a language.
computer programming is just that, a way to **express and digitize your ideas**.

That is how you **need** to think about computer programming. Pick a language and then stay focused.
Start with a simple language and focus on learning the ideas behind computer programming.

2. Programming is Hard

I am not going to sit here and lie to you… **Programming is hard**.
Learning anything for the first time is hard. It’s like learning your very first language.

Learning how to program is like a toddler learning how to walk. They try to walk, they wobble instead and fall hard on their faces.

You will try to take an online or college course that will ask you things like,
“Create a class that has methods to assign different bytecodes to different primitive data types. Then instantiate an instance of that class by using your constructor method… Do not forget to follow the style guidelines either”.

In the realm of computer programming, **you** will be the toddler.
Instead of the pavement hitting you in the face,
it will be your disappointment and failures slapping you across the face.

This will naturally lead you to conclusions such as but not limited to:

- *I am not good enough*.
- *Programming isn’t for me*.
- *Only if I had a better math background*.
- *Perhaps I should switch to another language.*
- *I will come back to programming later.*
- *I should drop this class, accounting is more fun anyways.*

That’s the bad news.
Luckily for you, there is also **good news**.
There is a solution to this predicament. I am going to share that with you now.

Want to learn what that dirty little secret is? Then read on.

3. Play: The Universal Learning Technique

Think about all the things in your life that you are truly good at. Were these things shoved down your throat? Were these things forced upon you against your will and you just became really good at them through pure discipline? I’m imagining that the answer to that is going to be a resounding **“NO!”. **If there isn’t anything that you can think of that you are good at… Perhaps you never truly liked anything enough.

Why do you think that is the case that you only get good at things that you associate with **playing**?
Well, the reason is that — dare I say it — you **enjoy** doing and learning those things.
"Learning and enjoyment can go together!? [**INSERT GASP HERE**]!".

Think about the #1 athletes in their sports or the #1 leaders in their industries… They are in that position because they truly enjoy what they do. They **THEN** (after enjoying what they do) develop systems of discipline to keep themselves in check and protect themselves from natural up and down spikes in motivation. Motivation is awesome but it is most certainly a limited resource.

There is only one common denominator between top businessmen and athletes and that is… **Play**.

They don’t wait for deadlines to get things done.
They are not waiting for someone to tell them what to do.
They don’t wait for some homework assignment or some curriculum to fall into their lap.

Why? Because they ENJOY what they do.

Gary Vaynerchuk was 12 selling baseball cards and now he’s 40 with a net worth of $10,000,000.

Michael Phelps used to absolutely love swimming when he was young.

I personally absolutely love chess and now I rank in the top 1% of all competitive chess players.

What does this mean for you? It means that if you want to become a programmer or if you want to get good at programming, you need to stop thinking about it as something you HAVE to do to get a job but instead start thinking of it as something **FUN you get to do**, **which HAPPENS to also land you a job**!

See the difference?

What do you like doing more: Homework for a class that you hate or a class that you are indifferent about, or for a class that you absolutely love?

This is the kind of mental preparation and neurological rewiring of your brainyou** need to do before you can even think about starting to code**.

But how do I do this Qazi? – I am glad you asked.
Stop trying to ONLY follow a curriculum.

But I don't know where to start.
That's why I said ONLY in capital letters. A curriculum is good, someone took a lot of time to curate something for you so you can learn it all in once place. However, don't confuse passive learning with active learning.

Most people passively follow a curriculum without actually doing anything. I want you to focus and stop on each step. I want you to make that step your own. I want you to step out of the contrived examples that your teacher gives you for where he thinks these concepts could be applied.

I want you to start thinking about how you could apply these concepts to something familiar to YOU. Make it relevant to you. Use each new concept you learn and **play** with it.

For example… Let's say you learn about the **random** module in the programming language called Python.
(I am assuming you don't know what module means but that's okay).

They teach you by saying *random module: This module implements pseudo-random number generators for various distributions*.

Does that make a lot of sense to you? Is it very relatable to you? Probably not.

How about create a game that makes funny sentences?
Create a list of adjectives where you randomly choose things from.
[ugly, cute-not-so-cute, snotty, intelligent]
Then make sentences where each adjective is randomly chosen based on user input like:

Tom is ugly
Bob is snotty
Julian is intelligent
Jill is cute-not-so-cute.

Now you have made it a heck of a lot more relatable to you and it will be more memorable because it’s lame and funny.

If you learn a new concept and have an idea that you want to implement, put your tutorial on pause. Go and explore that idea right away. Keep going as far as it takes you. Then and only then come back to the tutorial and continue.

That’s the secret to defeating hard. Make it play, explore, strive to fail, and ultimately you will overcome any obstacle.

The Plan for today

With that said, here is the plan for today:

1. Watch this video I made: *<a> W</a>hat Programming Language Should I Learn First?*

2. Go learn about different languages and why they could be beneficial to you.

3. Actually select one language today so you'll be ready for the next lesson where we will go over some programming concepts.

   ```
   (Remember lessons go out every week on Monday.)
   ```

I strongly recommend starting with a simple language like Python.

This way you do not have to deal with both conceptual difficulty (of learning computer programming concepts) along with syntactical difficulty (remembering to put semicolons, squiggly brackets, and a lot of other complicated stuff) at the same time.

You can get started now watching these tutorials: Learn Python for Beginners.

Talk to you in the next lesson, when I'll show you how to actually start programming.

–Qazi
http://cleverprogrammer.com

If you need convincing on why you need to program… Here is why:

Top 4 Reasons Why You MUST Learn Computer Programming

If you want to see what programming language should you learn first:

What Programming Language Should I Learn First?

Finally, if you want an excellent book to start programming in Python:

Python Programming: An Introduction to Computer Science

P.S – That book will also be helpful for your lesson – But you don't ***need*** *the book.*