



CregiTech
CREGITECH LEARNING
COMMUNITY

Tech Talk 01:

**Becoming a Programmer –
Everything You Need to Know**

Becoming a Programmer – Everything You Need to Know!

Here are the highlights of the tech talk:

- What is programming about?
- Who is a programmer?
- Why is it important for you to learn how to program a computer?
- Is it difficult to learn programming?
- Do you need a laptop to learn how to program?
- How can you find the best programming path to focus on?
- What programming language should learn?
- What tools do you need as a programmer?
- How can you become a better programmer?
- SUMMARY!

Let's begin!

What is programming about?

Computer programming is a way of giving computers instructions about what they should do next. It involves coding, modelling, simulating or representing facts, data or information using pre-defined rules and semantics.

Programming is like cooking. Both tasks require a set of instructions that are to be followed with no room for errors.

Who is a programmer?

A programmer is a person who simply defines a problem, plans a solution, code the program that'll solve the problem, tests the program and finally documents the program.

He defines what he knows about the problem, the objectives of his solution, selects a programming language to use, debugs the program in stages after completion to ensure no errors are introduced and then documents the design, development and testing of the program.

Why is it important to learn how to program a computer?

With the ever changing face of computer technology and with the moving of the world to the 4th Industrial Revolution, programming just become exciting and an always challenging field you should take a visit to sometime soon.

With computer programming, you can create anything you want. Computer programming gives you the ability to digitalize your ideas. With code, you can implement any idea that comes to your mind.

Imagine being able to finally actualize that idea that has been running through your mind for ages now.....

Realistically, you might not be able to save up a great pile of cash and then give them out to someone else just to make that app for you, because it's just too expensive.

On the other hand, even if you had the money, it is very unlikely that the person you hire will implement the idea of the app exactly how you want it to be.

Here's another reason why it's important you learn to program.

The App Association in 2017 said that "there are more than 220,000 software developers job openings in the United States alone. There are more job openings than there are qualified applicants to fill them."

By 2030, the statistics is expected to increase to more than 1 million.

More to that, programming makes our lives much easier and better. Think of our mobile phones, the software we're using now (WhatsApp) and other features you enjoy because someone decided to program it.

Is it difficult to learn programming?

Well if you think learning to program is difficult, YOU'RE WRONG!

Yet, it is difficult to master programming. To be honest, you do not need to be brainy to start coding.

Motivation, Passion and Discipline are enough.

Some persons on the other hand also think programming languages are foreign and hard to read. Well, high level programming languages are not difficult!

Unlike learning a foreign language where you are faced with a different set of alphabets, unfamiliar grammar rules and a whole new set of words you've never seen, learning a programming language is much more easier.

Let's take some sample codes from 2 programming languages to help you understand better.

Here's a piece of code written with the R programming language. We're given a dataset saved in Excel format (CSV) and expected to read it into R.

```
sample <- read.csv("cregitech-community.csv")
```

Or consider another code written in Python. This code calculates the area of a triangle with inputs taken from the user.

```
a = float(input('Enter first side: '))
```

```
b = float(input('Enter first side: '))
```

```
c = float(input('Enter first side: '))
```

```
# To calculate the semi-perimeter
```

```
S = (a + b + c) / 2
```

```
# To calculate the area
```

```
area = (S * (S - a) * (S - b) * (S - c)) ** 0.5
```

```
print('The area of the triangle is %0.2f' % area)
```

Notice 3 things from the above codes:

1. The alphabets are all standard Roman alphabets and can be found on the keyboard.
2. Observe the grammar rules and syntax, they are familiar and readable.
3. Look at the words. All words there are written in English Language. No foreign words and no difficult pronunciations.

As you can see, learning the commands and functions of a programming language compared to a real world language is a lot more easier.

Do you need a laptop to learn how to program?

Well for me I'll say NO! Yet, if you want to have the best programming experience, you should consider getting a laptop.

But come to think of it, why wait to get a car when you can get a motorcycle instead and save yourself the stress of trekking long distances?

My point is getting a laptop is not compulsory! You've got a mobile device. While not start learning now so when you can get a laptop you'll be better prepared?

Well excitingly, programmers have created hundreds of free mobile code editors for different languages. These editors enable you edit and test your code.

For me, this is an awesome way to start!

How can you find the best programming path to focus on?

For a moment, first think:

Why do I want to learn to code? What creative idea do I have in mind that I'll love to implement?

Now check out following tech paths and see if your answers match any of them.

1. Mobile Developers - They build apps that work on Android and iOS devices. If you're interested in mobile devices and how they work, then this path is for you. Mobile Developers must have in-depth knowledge of how a mobile device works to make the app they work on performant.

2. Data Scientist - Data Scientists gather and process data. This processed data helps businesses solve problems. They work with Software Engineers who created some logging feature somewhere in the app or website. This data points then gets to the Data Scientist. He then gathers all of them and derives insight from them.

Data Scientists can spot new paths that lead to growth, new ways to improve current product based on insight from data and more.

If you love looking at data and trying to see patterns, then this is for you!

3. Frontend Engineer - They use HTML, CSS and JavaScript to build User Interfaces for websites. They focus on aspects that are important to the website such accessibility, security, usability and performance.

If you love working with Use Interfaces and love the web, then this is a great path for you.

4. DevOps Engineer - This is the most difficult path to explain. A DevOps as the name implies takes on the role of a Developer (Software Engineer) and an Operations Manager.

As a Developer, they want to build as many features as possible on an already existing app so they will look good to others.

Yet, as an Operation Engineer, they focus more on stability. This means that they don't want to release too many features as these features could come with unpredicted outcomes, which results in instability, thereby threatening their goal.

It takes years of experience though to become a DevOps Engineer because the vast amount of knowledge or tools they'll need to learn.

If you love learning new things (the idea of knowing every aspect of your tech stack should inspire you) and if you would like to become the person everyone runs to whenever something goes wrong, then you should think of becoming a DevOps Engineer.

5. Backend Engineer - They focus on databases, scripting and the architecture of the website. Code written by a Backend Engineer helps to communicate the database information to the browser.

Their major roles include:

To understand the goals of the website and come up with effective solutions,

Store data and also ensure that it is displayed to that user who has access to it,

Responsible for development of payment processing systems like accepting data, securely storing that info, and making charges to that payment amongst a host of other activities.

If you can take on any of these roles, then this path is right for you.

What programming language should I learn?

I'll be answering this question based on the programming paths mentioned earlier.

1. **Mobile Developer:** For those who'll love to become mobile app developers, you should check out these languages and pick anyone that suits you.

#Flutter: Created by Google, Flutter is used to develop applications for Android and iOS.

#Kotlin: Created by JetBrains, Kotlin is a cross-platform and general-purpose programming language.

#Swift: Created by Apple, Swift is a fantastic way to write softwares, whether it's for phones, desktops, servers or anything else that runs code. It is age, fast and interactive and friendly to new programmers.

2. **Frontend Engineer:** If you'll love to become a frontend developer, then **HTML, CSS, JavaScript and Bootstrap** are languages you need to learn.

3. **Data Scientists:** For those who love Data Science, check out these 2 languages and see which one you'll love to use.

#R: Created by the R Core Team, R is both a programming language and a free software environment for statistical computing and graphics supported by the R foundation for statistical computing.

#Python: Designed by Guido Van Rossum, Python is an easy to learn, powerful programming language. It is efficient, runs everywhere and is friendly.

4. DevOps Engineer: For those who'll love to be a DevOps Engineer, you should start by learning how your own code gets released into a production environment.

This can be done by building your own website and deploying it yourself.

5. Backend Engineer: If you'll love to become a Backend Engineer, learn to code with **PHP, MySQL, .Net** or any other database management programming languages.

What tools do I need to start programming?

First, you'll need to download the necessary code editors/ IDEs. Some IDEs can read, edit and execute files from different languages. For example, the Sublime Text Editor can be used to read, edit and execute HTML, CSS, PHP, JavaScript, Bootstrap and other programming files. In other cases, IDEs like R can only run codes from their specific programming languages.

Secondly, it is necessary for you to join online developers communities.

One of such communities is GitHub. It is one of the world's learning software development platform. There you can host and review code, manage projects and build softwares along with 36million other developers.

Another awesome community you should definitely join is StackOverflow. It is a question and answer site for professional as well as enthusiasts programmers. With StackOverflow, you get answers to your toughest coding questions, share knowledge with your consumers in private and even be able to land your next dream.

Finally, the Cregitech Coding Community to be launched in September is another awesome community of developers you should be part of. Being part of this Community gives you the opportunity to meet, interact, work on projects, participate and win competitions and most importantly build solutions!

How can you become a better programmer?

The truth is NOBODY can learn everything about a programming language. Even professional programmers still come across errors, they still make mistakes. But you can grow. Here are some awesome tips I believe can help:

1. Learn, Unlearn, Relearn - To succeed in learning to program, you need to be open to new knowledge and ready to apply the acquired skills in practice.
2. Participate in competitions - Coding competitions are awesome opportunities to build yourself as a programmer. Even if you don't win, the knowledge lasts forever.
3. Attend programming hackathon and meetups - They are awesome opportunities to meet other developers physically.

4. Complete exercises - Check online for coding questions and try solving them.
5. Build something - Knowledge without use is wasted! Put the knowledge you've gotten into use, even if it's small. Remember Jerusalem was not built in one day or one month either, it took a lot of time.

SUMMARY:

Programming is not as difficult as painted! It helps you develop your structured and creative thinking abilities.

Learning to program teaches you PERSISTENCE!

Learning to program a computer will be valuable to you REGARDLESS of whether or not you go on to make a career out of it.

The fact that you're learning to program doesn't mean you now have a responsibility of creating the next Twitter, Netflix or Facebook. We just need you to solve problems that affect your life!

There is a programming path for everyone. If none of the paths mentioned doesn't fit you in, don't worry there are more paths.

I just had to limit them to 5 because of time. Be sure to ask for more if you want to.

You can create virtually anything that comes to your mind with programming.

Don't let those awesome ideas of yours go down the drains! Make them work, make them happen.

All of us at Cregitech are glad to help you out if you'll need more help on this!

Thanks for sticking with us, we hope you enjoyed our first tech talk?

You're free to ask your questions now!

#Team_Cregitech

If you haven't, be sure to join us on:

Facebook @Cregitech

LinkedIn @Cregitech

Instagram @Cregitech

WhatsApp: bit.ly/cregitech-community

DON'T ALSO FORGET TO SPREAD THE WORD.....YOUR FRIENDS, FAMILY MEMBERS, EVERYONE ON YOUR LIST NEEDS TO HEAR ABOUT THIS!

P.S: Please the Official Cregitech website is currently down for now. This is in preparation for our summer coding courses coming up in September.

Please do not visit or refer others to the website for now!