

Search

David Murphy

04 June [Development](https://messapps.com/tag/development/)

**Speaking Code – A Complete Review of The Basics**

## The learning process for speaking code is no different from the learning process of any other skill or language.

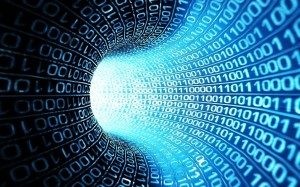
**Remember that the best way to expedite a learning process is by either re­acquainting or first**

**familiarizing yourself with the basics. Below you'll find a collection of the most frequently asked questions we hear from beginners.**

**Why is programming sometimes referred to as “Coding”?**

Because a programmer’s “source code” is like a set of instructions that the CPU ­ the Computer Processing Unit ­ cannot initially decipher. That’s why “higher” language ­ your source code ­ is oftentimes considered an “encoded” sequence of instruction for the CPU to read. The “higher” programming language that you choose ­ and there are many ­ is then translated from your source code into a “lower language.” This lower language is a form of input that your CPU can then register with the help of a compiler program.

## What's a Compiler?



A compiler is essentially the middle­man between you and the CPU that’s included in a Software Development Kit. It takes all of the source code that you've written in a higher language and transposes it into a stream of numbers that the CPU can then read ­ machine code. As a rule of thumb, a program is compiled *before* it's run

## What's an SDK?



*Source: Cometchat.com*

SDK stands for "Software Deveopment Kit," which is exactly what it sounds like: a package of software that a programmer uses in order to create programs. This kit usually comes with an integrated programming language. Each kit contains compilers, debuggers, and any other facilities needed to develop apps for a specific platform. For the most part, it's highly uncommon to simply code using an SDK; it’s typically coupled with something called an Integrated Development Environment.

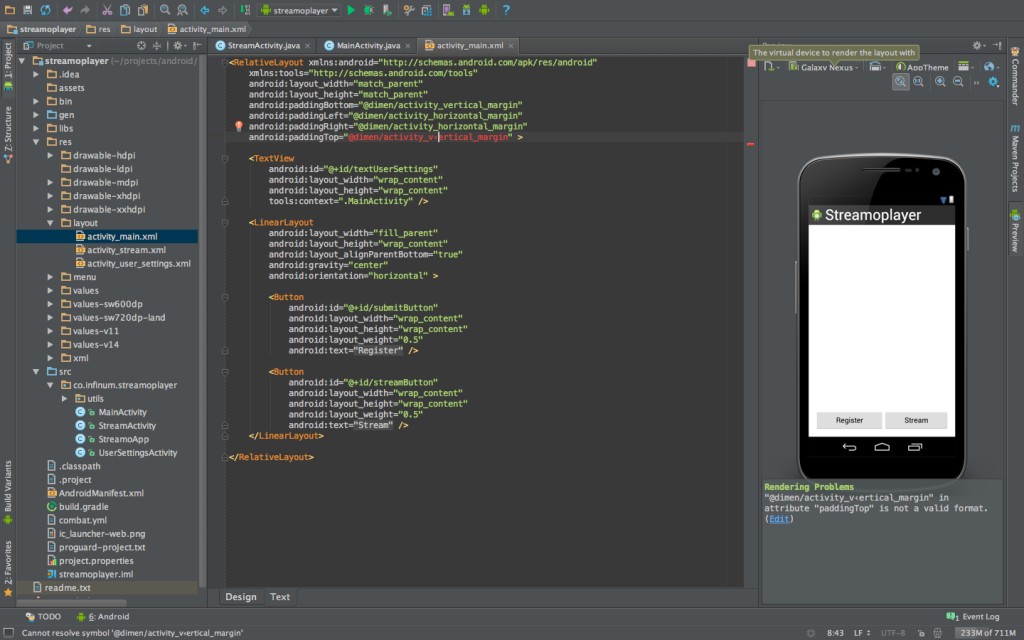
*iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii*

## What's an IDE?

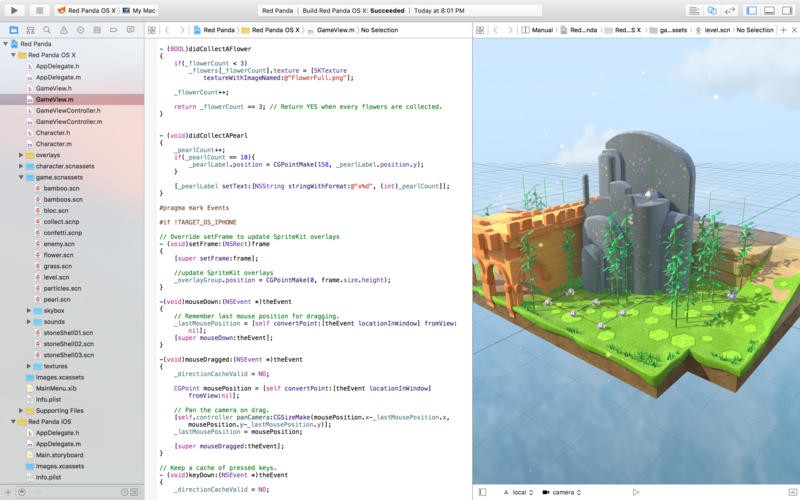
An IDE is an Integrated Development Environment that *contains* an SDK. Most IDEs are tailored to function around a particular SDK. An IDE’s purpose is to make coding with an SDK more manageable. What an IDE is to a programmer is essentially what an instrument is to a musician: although all instruments can play music, certain instruments allow a musician to express him or herself through a more desired/preferred format. Although all IDEs can be used while coding, one IDE and its included SDK might be preferred or even necessary for creating a program that is only compatible with a particular platform.

## Android and iOS IDEs

[**Android Studio**](https://developer.android.com/develop/index.html)



## [Xcode](https://developer.apple.com/xcode/)



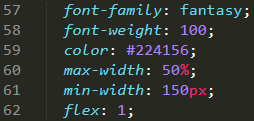
Java and Objective­C are both programming languages used for coding Android and iOS apps respectively. Developers interested in developing apps for iOS devices should first familiarize themselves with the IDE called Xcode, which uses the iOS­compatible language of Objective­C. As for Android devices, Eclipse and Android Studio are both the go­to IDEs for coding in the Android­ compatible language of Java. This also means that Android devices*cannot*run apps that are coded in Objective­C, and an iOS device*cannot*run apps that are coded in Java. There are, however, a few up­and­coming IDEs which are considered cross­platform compatible, such as [*Xamarin*](https://www.xamarin.com/).

## What is HTML and CSS?

Both HTML (Hyper Text Markup Language) and CSS (Cascading Style Sheets) are programming languages that function in website development. HTML is used to structure a website with links, images, texts, videos and other “elements”; CSS is used in tandem with HTML to further stylize the layout of the webpage by re­arranging the colors, fonts, and positioning of HTML elements.

## What are the three main control structures of programming?

1. **Sequential Execution:** Instructions in your source code that are compiled and executed in the same order as they appear. When ending a line of code, instead of using a ‘period’ mark like you normally might in English, you’ll be using a semicolon. After each semicolon, it is standard procedure to start the next instruction one line away from where the previous line ended. Check out the image below and notice how each instruction that's ended with a semicolon is followed by another on the next line. The instruction on line 57 begins at “font­family” and ends at the next semicolon, and a new instruction is created on line 58 in a similar fashion.

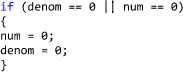


*\*Written in CSS*

Some languages don’t assign the semicolon with this purpose, such as HTML and Python, but most do: Xcode, C++, Java, CSS, etc.

1. **Conditional Statements:** A more specialized instruction, known informally as the “if­then” statement:*if*a certain*condition*is met,*then* the included set of provided instructions will follow. In the image below, the phrase between the first parenthesis “(denom

== 0 || num == 0)” is the conditional statement which, in this case, is testing whether or not the variable “denom” or the variable “num” is holding a value that is equal to 0. If it is, then the instruction below the conditional statement which is positioned between the two brackets are run, and the two variables are, again, set to the value of 0.



*\*Written in C++*

Note: variables are assigned in C++ like so: "int denom = 0". "int" designates the variable "denom" as an integer, and "= 0" sets the variable "denom"'s value to 0. The variable can be changed at any time once it's created by typing the variable's name and resetting its value, such as the conditional statement does above: "denom = 0;".

1. **Looping:** Another specialized instruction, a looping statement is a sequence of instructions that is repeated until the specified condition is met. In the image below, the conditional statement is checking if the value inside of the variable ‘i’ is less than the value inside the variable ‘n.’ For every time the condition is NOT met, the loop will continue to repeat itself, starting with the conditional statement. That is, until the value inside of the variable “i” is greater than the value inside the variable ‘n’ ('n' is given a value at an earlier line in the source code), the loop will continue to run, and the instruction enclosed in brackets “cout << a[i] << “ “;” (which displays a number into a seperate console window) will run once every time the loop’s condition is NOT satisfied. The phrase “i++” means that the value inside of the variable “i” is being increased by one whole integer each time the loop is run. Eventually, the value of the variable “i” will be greater than that of the variable “n,” and the instruction listed between the brackets will stop being triggered.

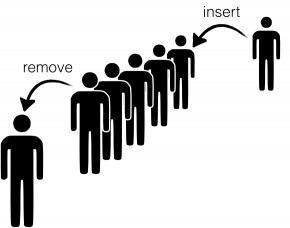


*\*Written in C++*

## What is a Queue vs a Stack (FIFO vs LIFO)?

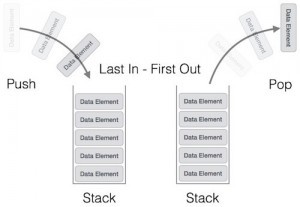
Both queues and stacks are ordered arrays (lists), meaning that the order of their contained variables is significant. Ordered arrays are useful tools to have for problem­solving at all levels of coding.

* **Queue:** an array of variables in which the first variable (a phone­number, for instance) that is inputted into the array (a list of phone numbers) will also be the first variable that the queue will output when it's commanded to do so. The image below illustrates that concept through the similar notion of waiting at a line at the check­out isle: the first person on the line will be the first person to get their groceries checked out.



*Source:* [*ucsd.edu*](https://messapps.com/academy/articles/speaking-code-complete-review-basics/ucsd.edu)

* **Stack:** an array (a list) of variables (phone numbers) in which the last variable to be inputted into the array is also the first variable that will be outputted. The image below illustrates how the entrance of the array is also its exit.



*Source:* [*tutorialspoint.com*](https://messapps.com/academy/articles/speaking-code-complete-review-basics/tutorialspoint.com)

0

You might also like this articles



Dominick Wojtas

19 December

# [“An idea of a life time comes once a month”](https://messapps.com/academy/articles/idea-life-time-comes-month/)

As many entrepreneurs will tell you, great ideas do not fall out of the sky and into your lap. An intimidating

Dominick Wojtas

25 July

# [Mobile App Design Guide: Sketch Vs. Photoshop](https://messapps.com/academy/articles/mobile-app-design-guide-sketch-vs-photoshop/)

The debate between Sketch and Photoshop is blazing stronger than ever. Photoshop has been the most

start to the app developing process is coming up with the actual idea. If pressure to make it ‘great’ from its conception is removed, then it allows for a wider velocity of ideas to surface.



0

commonly used tool when making mockups for mobile and web design for years. However, it seems that the days of Photoshop’s reign in the industry are no more. Sketch has been bubbling quietly for some time now but with the latest release of Sketch 3, I wouldn’t be surprised if Adobe is panicking about losing some of its market share.

0