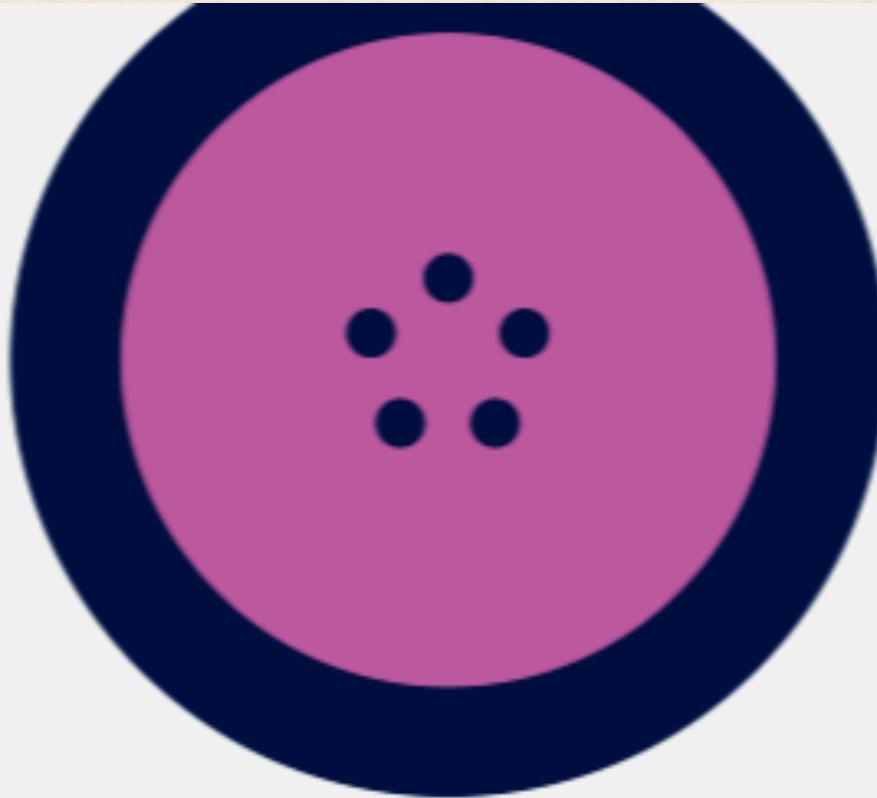


Anyways... So what my
point was is...



But who am I kidding right?



Huckleberry JUCE



ADC

[https://www.youtube.com/channel/
UCaF6fKdDrSmPDmiZc19KLnQ](https://www.youtube.com/channel/UCaF6fKdDrSmPDmiZc19KLnQ)



There are what are called high level and low level programing level languages.

Believe it or not it's the lower level ones that only Bill and Steve could conquer in their early adulthood.

But do you think Steve Jobs ever let what Bill was doing slow him down. No, he know the destination and it never mattered how the destination illuminated it's self by encouraging failure to reach for a bigger fish... a better product, a bigger dream.

Or maybe not.

Object Oriented Design

The type of programming language we need to be familiar with is **Object Oriented Programming** in C++ when we are dealing with audio. Because it's fast I'll explain in a moment.



JavaScript is not Java

Java is much like windows c# with different syntax and nuances that idk care to know.

Java is used in Android applications and may programs such as Oracle.

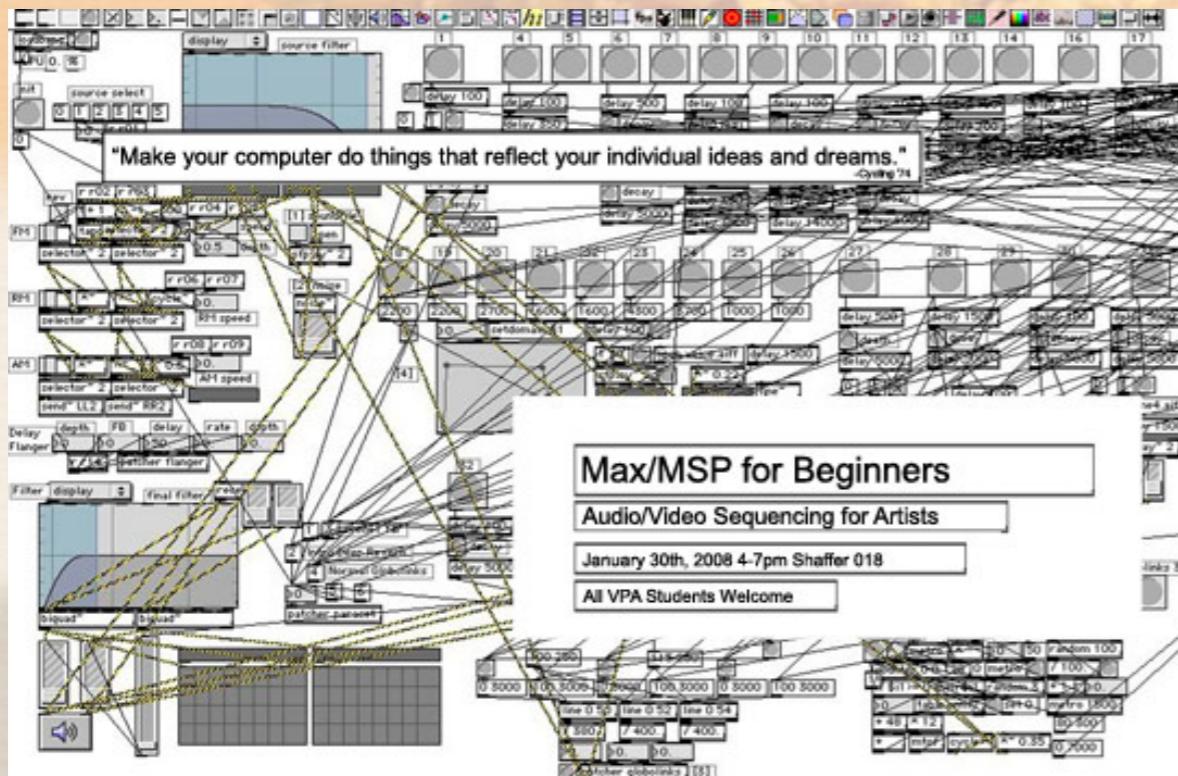
Javascript is used primarily as function control with, html is kind of your main document, and css is to keep your website from being “90’s”.

There is a new c++ library called, “Oboe” available on GitHub developed by those working at Google’s Android and the Audio Developer Network.

It can be compiled in Android studio along with it’s Java to create a, “Multi-Threaded” Audio application, with far less latency.

So which came first the Chicken or the Sandwich?

[https://www.youtube.com/watch?
v=4OQVBX2vx_g&t=4s](https://www.youtube.com/watch?v=4OQVBX2vx_g&t=4s)



WE
GOT
SOUL

<https://soul.dev/>



Using the FAUST DSP language and the JIT compiler with JUCE

[https://www.youtube.com/watch?
v=INlqCIEOhak](https://www.youtube.com/watch?v=INlqCIEOhak)

Android Studio



<https://developer.android.com/studio>

Your First Real Coding Project

Turn any website into an android app!

Source code -

<http://www.codebind.com/android-tutorials-and-examples/convert-website-android-application-using-android-studio/>

Tutorial Video -

<https://www.youtube.com/watch?v=a5dlmqM9Oo8>

About that Latency....

<https://juce.com/maq>

IDE Xcode for Mac

[https://apps.apple.com/us/app/xcode/id497799835?
mt=12](https://apps.apple.com/us/app/xcode/id497799835?mt=12)

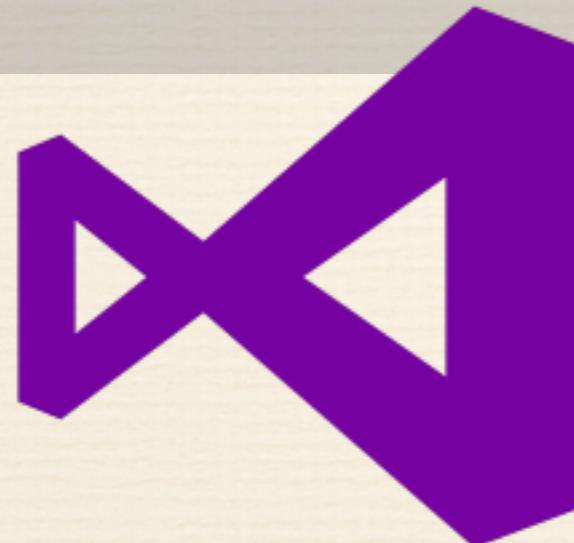
or



Windows/Mac

IDE Microsoft Visual Studio Community -

<https://visualstudio.microsoft.com/>



What is really cool, is we can
debug and test our code in
Ableton Live.

In order to truly check to make sure that an audio
application is functioning properly there's defiantly
no better tool!

Digital Audio Theory

Digital Audio Theory

You see it's actually a very simple concept that we tend to overwhelm ourselves with.

Maybe it's the triple threat of letters put together to create a word that fits in a phrase, inside a sentence, of a paragraph to convey information that is also going through our own personal interface so that we can try to interpret a particular message.

Analog signals = **Continuous Time Domain**

And

Discrete Time Domain = Which would be a **digital signal** sending 0's and 1's.

Like

01101101 01010110 01010100 101010101 01010101 01010
1030101 0001001 0010010 0100101

And remember your computer only processes one thing at a time.

For now let's just keep things simple and understand that Analog sound is a constant data signal being feed into the computer. Meaning that "Audio" is Harder on you're computer than Video due to the fact that video only needs to be played at 24fps to create the phenomena of continuous motion. And in todays world, on our way to 8k monitoring the video and audio will be fighting the rest of the hardware even more.

Let's break this fish down
Shall we...

Sampling Rate - ?

Sampling Rate - Number of samples collected each second, which can be used to represent a sound wave digitally.

The Nyquist-Shannon Theorem:
In order to reconstruct a signal, sample it at a rate greater than twice it's frequency range.

Human Hearing is somewhere around
20Hz - 22khz or 22.5khz.

In order to duplicate that digitally and
to hear it back as high quality audio, we
need to double this to 44.100khz.

This will leave us with no frequency
loss. Disregarding how you may feel
about 96k or w/e. Because at this point
you are oversampling and creating
something called fold-back which isn't
important today.

What is important is that in 1 second 44,100 samples are being collected and broken down into block of 512.

This ratio is important, I'll leave it at that. To summarize that constant audio sample is being ran on what is called a Thread. And in order for an application to stop ever 512 samples to check for information such as new incoming data or processes that need to be performed. This is why its hard on your device to play high quality video along with high quality Audio... This \$2,300 MacBook or any device for that matter is only able to push the boundaries of physical limitations so far. Surprisingly us humans were actually designed quite well.

Audio Plugin Architettura

The “GUI” or “Graphical User Interface,” part of an application is what you can see. Let me ask you this? Have you ever pressed stop on a device and it took a moment for the audio to quit? This is due to the fighting between the GUI and the AUDIO Thread. You see you have a GUI Thread, Audio in(from Host), Midi in(from Host) and this all gets feed into the audio engine. Which is also a thread at the same time, sort of seaming the audio back together in a round about way to allow for high fidelity 44.1khz audio. This is why every device with audio hasn’t been the greatest sounds produced. We would sacrifice the quality of our audio in order to play it back at a lower sample rate which will be an audible difference, “and not a good one!”

So why do we want to use JUCE so damn bad and not all the other things you've heard come in and out the door?

Many reasons.

One being that JUCE is cross-platform!
It is a really difficult learning curb to know how to create the same object from one system to another.
This is one reason that is quite critical. It is much nicer to be able to do it once and know that you won't have a mess across the line up.

Roli and Midi 2.0

<https://juce.com/discover/stories/introducing-midi-2-0>

<https://www.midi.org/articles-old/details-about-midi-2-0-midi-ci-profiles-and-property-exchange>



Sippin on Gen & Juce -

**[https://cycling74.com/tutorials/
gen~-for-beginners-part-1-a-place-
to-start](https://cycling74.com/tutorials/gen~-for-beginners-part-1-a-place-to-start)**

Tutorial Video

**[https://www.youtube.com/watch?
v=f94Go5Qafg8](https://www.youtube.com/watch?v=f94Go5Qafg8)**

Okay... but how the f?:{ do I learn how to code so easily and st8 to the source. Because that kinda matters if I wanna do any of this shit... ?

Very easy once you've been the wrong way
Danial Son, yet sometimes not always as easy
to take a so calledAnyways, let's make sure
we drink our daily Juce....
but first we have a small grocery list.

Get Juce - Projucer !!!!

<https://shop.juce.com/get-juce/download>

Learning Juce

<https://juce.com/learn/tutorials>

[https://www.kadenze.com/courses/
intro-to-audio-plugin-development/info](https://www.kadenze.com/courses/intro-to-audio-plugin-development/info)

<https://theaudioprogrammer.com/>

Udemy - Best course for learning
Web Design and Development

[https://www.udemy.com/share/
1013gGAksTeVxaRX4=/](https://www.udemy.com/share/1013gGAksTeVxaRX4=/)

Create a GitHub Account
[https://github.com/join?
source=header-home](https://github.com/join?source=header-home)

learn to code you said...?

Good old Mike Dane can help you understand the differences between the syntax not so much the logic.

<https://www.mikedane.com/programming-languages/c++/>

Best app ever though is called

<https://www.sololearn.com/>

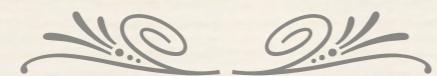
HTML5 Audio -

<https://www.html5tutorial.info/html5-audio.php>

Integrated Code Editor for Web applications

<https://atom.io/>

Ohh and btw please
allow me to introduce
you to YOURSELF.



*It's for your own good. I'm not an
asshole, “at least intentionally, “most-
the-time” n shit.”*

https:// www.kadenze.com/ partners

“I feel as if our program could benefit not only being connected to the rest of the industry, but to allow ourselves the opportunity to continuously recreate new solutions to upcoming hurdles, as we grow and make our way to the top. Because this is an industry that doesn’t pause and wait. It’s important stay current on events.” - Paul Weber

To learn more about Real-Time Audio visit.

<http://www.rossbencina.com/code/real-time-audio-programming-101-time-waits-for-nothing>

*“Thank you for your time!” .. ohh and you’re quite
welcome BTW.*

(317) 709-8132

pauweber@iu.edu

<https://www.linkedin.com/in/paulmweber/>

—Paul Weber