

Open Access Case Study

Effectiveness Case Study of Teaching Code and the LiveCode Extension

Adam Thomas, Ph.D.

Citation: Thomas, A. (2022). Effectiveness Case Study of Teaching Code and the LiveCode Extension. The Interdisciplinary Journal of Student Success.

Founding Editor-in-Chief: Masood Zangeneh, Ph.D.

Editor: Hamid R. Yazdi, Ph.D.

Editor: Mona Nouroozifar, Ph.D.

Received: 05/20/2022

Accepted: 08/10/2022

Published: 12/15/2022

Faculty of Media and Creative Arts, Humber College Institute of Technology and Advanced Learning, Canada

Corresponding author: Adam Thomas: adam.thomas@humber.ca

Abstract: This case study explores the use of the LiveCode Visual Studio Code extension to teach programming in post-secondary education.

Keywords: Accessibility, Code, Learning, Programming, Teaching, Technology.



Copyright: ©2022
Thomas, A. Licensee
CDS Press, Toronto,
Canada. This article is an
open access article
distributed under the
terms and conditions of
the Creative Commons
Attribution (CC BY)
license
(<http://creativecommons.org/licenses/by/4.0/>)

Introduction

Learning to code is becoming one of the most demanded skills, even for students who are not aspiring to become career programmers (Darby, 2021; Wang, 2017). This includes industries such as audio production, agriculture, finance, and marketing. However, learning code is difficult. In the Faculty of Media & Creative Arts (FMCA) at Humber College, students rank coding among the most difficult topics.

The LiveCode extension helps remove unneeded obstacles students face when learning code by providing class content in an accessible format, making learning more equitable across socioeconomic status and location, and improving the online learning experience (especially during the lockdowns resulting from the COVID-19 pandemic).

Background Information

When teaching code, professors often write code as students attempt to follow along on their own computer (Chen, 2019). This code is either displayed using the projector in class or streamed by sharing the professor's screen using a tool like Blackboard Collaborate or Twitch.

Figure 1

Code displayed on a screen can be hard to read when learning in person

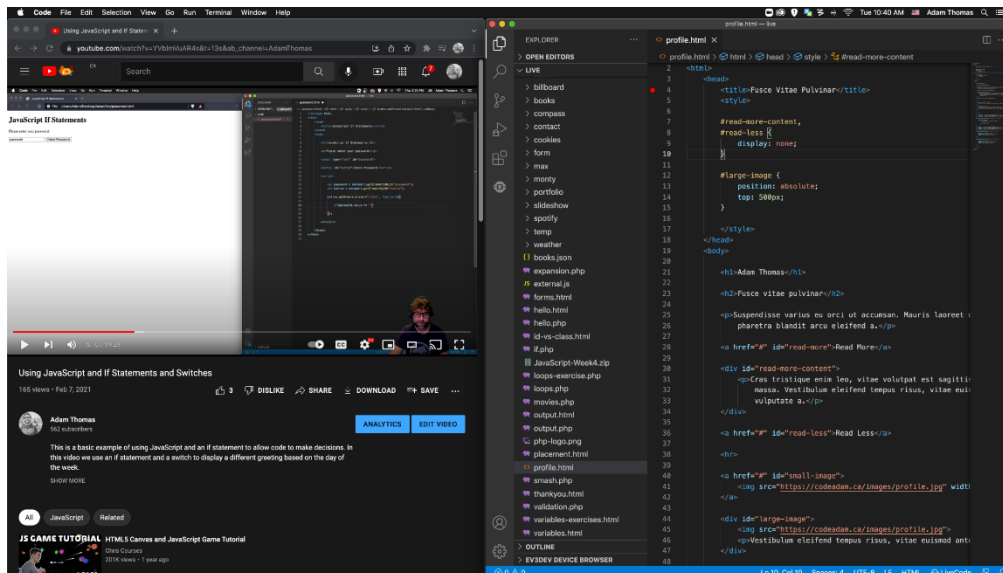


However, this method makes learning less accessible for many students and does not conform with how today's students prefer to learn (Wang, 2017; Baticulon, 2021). For example:

1. Students who are slow typers can easily fall behind and may find they do not have access to the code required to catch up.
2. Code in a video stream can be blurry; especially if the students' internet connection is not optimal due to living in remote areas with slower connections (Muilenburg, 2005; Baticulon, 2021).
3. If students cannot afford high end equipment (high definition screens, multiple screens, a second device, etc...) following along can be difficult as they do not have the screen space to both view the professor's code and their own code (Muilenburg, 2005; Baticulon, 2021).
4. Students with visual impairments, learning disabilities, or ESL (English as a Second Language) cannot utilise assistive technology (like screen readers) with code shared on the projector or streaming (Guo, 2018; Baticulon, 2021).

Figure 2

Following along remotely requires a lot of screen space



The LiveCode Extension

The LiveCode extension aims to ameliorate these barriers to learning:

1. To prevent falling behind, LiveCode allows students to select which file and/or block of code they are viewing, regardless of the professor's pace, enabling them to keep up. Students don't have to ask the professor to go back to previous code which can cause a student to feel self-conscious.

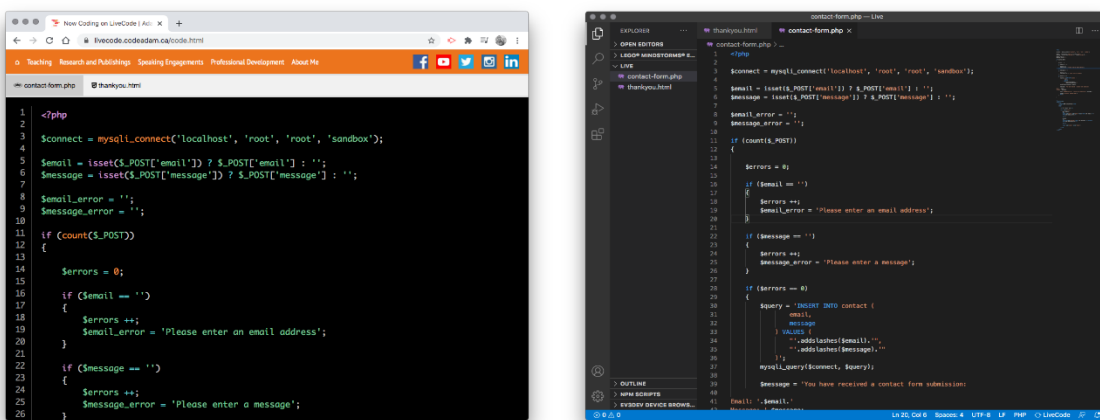
2. While one megabyte (MB) of streaming video in a standard resolution provides about three seconds of video, 1 MB of streaming code in plain-text provides over 20,000 lines of code. Streaming code in plain-text requires approximately 0.005% of the bandwidth required by streaming the same code through video. This ensures class content is delivered smoothly even for students with less reliable internet connections.

3. By streaming code in plain-text, students can use a smaller portion of their screen and still effectively follow along.

4. Code provided in plain-text is much more accessible: students can zoom in and the text remains clear, it can be converted to audio or braille using assistive technology, and students can copy and paste blocks of code from the examples during a live lesson.

Figure 3

LiveCode Student View (Left) and Professor's Code (Right)



Existing Solutions

There are some existing tools for sharing code, but they require professors and students to use specific programs, install additional plugins, and/or register for online accounts.

LiveCode, on the other hand, is designed to be easy to implement. Professors simply need to install a small extension for their code editor (a 30 second process) and provide a link to their students. Students then open the provided link in any browser, and they are ready to follow along. It works on any device, with any browser, on any operating system, and without needing to install any software or register for any online accounts.

Impact

During the first class of our introduction to programming courses we walk students through using the LiveCode extension. The following is a selection of student reactions during one such demo:

“This is a game changer!”

“Can my other instructors use this?”

~ Anonymous, Fall 2020 - Progressive Programming Techniques

“Cool, Thank you for [making] this.”

“How did you connect [your code] to a website?”

~ Anonymous, Winter 2021 - Creative Coding

In Winter 2021, a basic Stop/Start/Continue survey (Burden, 2016) was distributed to six coding classes. The students were not specially questioned about the LiveCode extension, but the LiveCode extension was a common topic mentioned as positive feedback. The following is a portion of this feedback:

“[LiveCode] is a brilliant idea that I’ve seen no other teacher apply.”

“The LiveCode tool is really helpful because I am slow at coding.”

~ Anonymous, Winter 2021 - Design Practice 2

“LiveCode is a perfect idea!”

“LiveCode is very useful, I’m so happy for this tool :)”

~ Anonymous, Winter 2021 - Creative Coding

The inclusion of LiveCode has dramatically improved the delivery of course content for both in-person and online coding classes. It has received particular praise from recent international students, and in one case, a student studying from a remote location in the Yukon.

Implementation

LiveCode was developed in the Fall of 2020 after students during the first COVID-19 lockdown expressed frustration with following coding

exercises while learning online. It has been used by three professors in over ten different coding courses, with over 500 total students. Professors have easily integrated the extension into their classroom after only being provided a ten-minute demo. It does not require professors to make any changes to their existing course delivery.

LiveCode currently supports HTML5, CSS3, JavaScript, Node.js, Markdown, and PHP, and is currently compatible with Visual Studio Code (VS Code; the most popular code editor used by both faculty and students in the FMCA). It is available for download in the Microsoft Visual Studio Code Marketplace and currently has approximately 2,000 downloads.

Future Development

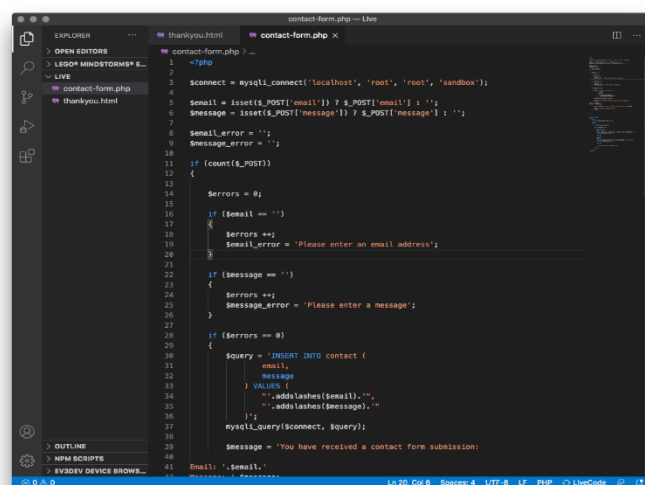
The following new features have been requested by students and/or faculty:

Colour Blind Mode

When using a program like VS Code, text is colour coded to make it easier to read (see below). However, some of these colour themes cause portions of the code to “disappear” to a student with a particular form of colour blindness. I have had students inquire about missing code on a slide when the colour theme was not inclusive to red-green or blue-yellow colour blindness. LiveCode allows students to choose a colour theme that best suits their needs.

Figure 4

A Sample of a Visual Code Colour Theme



Multiple Faculty

Currently the application will only facilitate one faculty member at a time. Faculty currently check in with others to make sure it's available when they need it. With some small additions the tool could be used by multiple faculty members simultaneously. Students would simply choose which faculty they are "following" when starting a session.

Download Class Files

Currently code is made available during the lecture and for approximately 30 minutes after the lecture has concluded. Students have requested both (a) a quick link to quickly download all files from a class, and (b) an option to retrieve class files from previous classes. LiveCode could easily be integrated with GitHub to archive this.

Additional IDEs

The LiveCode extension was initially created for use with Visual Studio Code. The extension could easily be adapted to work with other commonly used IDEs.

Conclusion

The LiveCode Visual Studio Code extension has provided an improved and more accessible learning experience to students learning code both in person and online. Students have been quick to remind professors to turn it on if they forgot. After integrating LiveCode into their regular classroom experience, professors have continued to use it, and students often mentioned the extension in a positive light when given the opportunity to provide course feedback (Stop/Start/Continue, Humber SFQs, etc.).

Funding

N/A

Declaration of conflict of interest

No conflict of interest.

Ethics Approval and informed consent

N/A

References

- Blackwell, A. F. (2002, June). What is programming? Proceedings of the 14th workshop of the Psychology of Programming Interest Group, Brunel University, Middlesex, UK. pp. 204-218.
- Chen, C. H., & Guo, P. J. (2019, June). *Improv: Teaching programming at scale via live coding*. In Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale (pp. 1-10).
- Baticulon, R. E., Sy, J. J., Alberto, N. R. I., Baron, M. B. C., Mabulay, R. E. C., Rizada, L. G. T., ... & Reyes, J. C. B. (2021). Barriers to online learning in the time of COVID-19: A national survey of medical students in the Philippines. *Medical Science Educator*, 31(2), 615–626.
- Burden, P. (2016). *A comparison of the usefulness for teaching improvement of qualitative feedback data gained from Student Evaluation of Teaching surveys (SETs) and the "Stop, Start, Continue"(SSC) method*.
- Darby, F. (2021, June). *Why coding is still the most important job skill of the future*. Online Courses. Retrieved February 1, 2022, from <https://www.upskilled.edu.au/skillstalk/coding-most-important-job-skill-of-the-future>
- Guo, P. J. (2018, April). *Non-native English speakers learning computer programming: Barriers, desires, and design opportunities*. In Proceedings of the 2018 CHI conference on human factors in computing systems (pp. 1-14).
- Muilenburg, L. Y., & Berge, Z. L. (2005). Student barriers to online learning: A factor analytic study. *Distance Education*, 26(1), 29–48.
- Wang, X. M., Hwang, G. J., Liang, Z. Y., & Wang, H. Y. (2017). Enhancing students' computer programming performances, critical thinking awareness and attitudes towards programming: An online peer-assessment attempt. *Journal of Educational Technology & Society*, 20(4), 58–68.

Key Terms:

The following definitions may be helpful to non-programmers before reading the remainder of this paper:

IDE (Integrated Development Environment): An application used to write computer Code.

Plain-Text: Content on a computer that can be highlighted, copied and pasted, magnified, and easily read by assistive technology. According to the Web Content Accessibility Guidelines (WCAG), plain-text content is the most accessible format of content delivery. The WCAG recommends that all non plain-text content (images, audio, and video) be accompanied by a plain-text alternative (image descriptions and captions).

Visual Studio Code (VS Code): A Microsoft tool used to write computer code.

Visual Studio Code Extension: An add-on that enhances your experience in Visual Studio Code by adding new features.

Programming: “A process of translating from the language convenient to human beings to the language convenient to the computer” (Blackwell, 2002, p.1)

Additional LiveCode Resources:

LiveCode information page:

<https://livecode.codeadam.ca/>

LiveCode demo:

https://www.youtube.com/watch?v=XcD_rj48wtY

LiveCode download on the MicroSoft Visual Studio Code Marketplace:

<https://marketplace.visualstudio.com/items?itemName=CodeAdamCa.livecode>