

back to [ERHS CS](#)

Learning to Program by Creating Games, Apps and Programs

See how easy it is to learn to program with an interactive, graphical user interface and a more English-like language. You still have to write code (like Python, C++ or Java) but can be more creative, and productive. You can create useful programs, games and even apps while learning the underlying computer concepts, commands and logic.

Why Learn to Program?: for [Students](#) - for [Teachers](#) - for [Everyone](#)

This website will help you get started with LiveCode and writing your game or app. You will also learn to write some really cool programs. It is for you to get started learning to program. It is designed for you to learn by doing. So get your hands dirty and do all the examples yourself. You will quickly feel at home with writing computer programs and understanding how it works. Soon you will be designing your own games and programs to put on your computers, tablets and smartphones. You will be amazed at what you can accomplish.

Thank you for using this website, enjoy...

p.s. If you have any suggestions, changes or ideas, email me at cypruszko@hotmail.com.

IMPORTANT: As of July 1, 2017, I will no longer have write access to this website. My new website is being built and can be found at LearnToLivecode.com

Note: *This site is continually being updated. So please come back often. (and don't forget to REFRESH this page upon returning)*

Background

LiveCode - about, info. and student examples:

- **About** - what it is like, what you can do with it: [LiveCode](#), [What is it?](#) *** **RECOMMENDED** ***
- **LiveCode?** - why it is a better language choice than Python/Java: [Why LiveCode?](#) *** **RECOMMENDED** ***
- **LiveCode for C, C++ and Java Developers:** - [Developers](#)
- **Main LiveCode Sites** can be found at: [LiveCode.org](#) and [LiveCode.com](#)
- **Resources** (Lessons, Scripts, Tutorials, ...) found at: [Resources](#) and [LiveCode](#) - *** **BOTH RECOMMENDED** ***
- **Examples of Programs** (some with code) can be found at: [Games/Utils](#), [Programs](#), [Apps](#), [Cards](#), and lesser ones at [Educational Games](#), [Edu Games 2](#), [Mazes](#).
- **Hear from students** (Videos) - as they demo their apps [2016-17](#) and [2015-16](#) - *** **BOTH RECOMMENDED** ***

Presentations:

[CG2017 Create Your Own Class Games and Apps](#)
[CG2017 You Can Teach Apps and Games](#)
[PUWT2016 You Can Teach Programming Apps and Games](#)

Getting Started: Installing it and understanding the interface

1. **Installation** - installing it on your own computer, and about the different versions ****MUST READ****
2. **Internals** - how LiveCode works, understanding its design (objects, properties, messages)
- *3. **The Screens** - the parts of LiveCode (Edit bar, Tool palette...) that you will be using ****MUST READ****
- *4. **The Tools** - the tools and objects (Edit/Run, buttons, graphics...) that you will be using ****MUST READ****

Other resources for learning: [BYU Lessons](#) - [LiveCode's Lessons](#) - [Dictionary](#) .

Lessons:

Legend

Colors used in the lessons

- red** = recommended lesson #s as a minimum
- blue/purple** = browser links to other webpages
- green** = LiveCode commands/messages
- orange** = code samples and scripts

Step 1 - Basics - First Programs (at a minimum, just do the numbers in red)

0 - Getting Familiar with the Interface and the Environment:

0.0 - Starting with the Basics - an intro/review of the LiveCode tools and palettes in the IDE (Integrated Development

1 - Getting Started Programming - simple commands:

- 1.0 - First Program** - writing your first program
 - 1.1 - Standalone Program** - creating a Standalone - to give to your friends
 - 1.2 - Asking Questions** (Answer, Ask, mouseUp, message box) - interacting with the user, asking questions
 - 1.3 - Making Decisions** (If ... then ...end if, Answer...with) - choices, different answers for different users
 - 1.4 - Backgrounds** (Import) - adding color, patterns and photos to your program
- Practice - [Interactive Programs](#) (Ask, Answer, If) - make fun programs for practice

2 - Getting Info From The User - more commands, working with data:

- 2.1 - Input/Output** (Get, Put, text fields) - getting info to/from the screens
 - 2.2 - Objects** (short, long and names) - objects and how to identify them, advanced techniques
 - 2.3 - Variables** (Add, Subtract, Put, text fields) - using variables to save information
- Practice - [Magic Math Program](#) (Put, text fields) - simple example of using text fields
- Practice - [Useful Calculation Programs](#) (Put, text fields) - make useful programs for computer or cellphone
- Practice - [Memory Game](#) (target,short/long name,msg path) - shortcuts and tricks to use
- 2.4 - Variables** - Part 2 (Set, Get, local, global, custom properties) - more on variables, custom properties

3 - More Interactivity - more screens, actions:

- 3.1 - Cards, Cards and More Cards** (Go, Next, Previous) - adding screens (cards, levels, pages, etc) to your program
- Practice - [Greeting Cards](#), [Multi-Pages](#) - make an interactive greeting card, book, a slide show and more...
- 3.2 - Changing Properties** (Set, Show, Hide) - making your programs more interactive and dynamic
- Practice - [More Greeting Cards](#) (Play, Move, Set, Show, Hide) - make fun cards that talk, sing and do things

4 - Wrap-up of Basics:

[Variables](#) - naming, using, special variables...

[Playing Music](#) - playing music and sound effects

Vocabulary and Review - [LiveCode Vocabulary](#) - Review and Preview of LiveCode commands and terms

[How-To ...](#) - A collection of useful How-To's and chunks of code to do different things (*just starting...*)

[Problems?](#) - common mistakes and their solutions

Step 2 - What would you like to learn to do? ...

Books - Creating interactive books, drawing programs, etc (*new...work in progress*)

Game Programming - Creating and writing your own games (*mature...minor changes in progress*)

App Programming - Writing Apps for mobile devices in LiveCode. (*new/current...work in progress*)

General Programming - Writing programs for everyday problems and uses. (*just starting...*)

Specific programs and projects to do for fun ...

[Animal Wizard](#) - It knows every animal. If you stump it, it remembers the animal for next time.

[Artificial Intelligence](#) - Create your own Chatbot, Virtual Assistant, Siri © or Google Now ©.

[Data/BigData](#) - How popular is your name? How many people in the US have the same name?

[DigitalMagic](#) - Add people/remove people from photographs, change backgrounds with code (*almost done...*)

soon [Cryptography](#) - Secret Codes, (*work in progress...*)

soon [Facial Recognition](#) - Recognize faces, tells when you need a haircut (*coming...*)

soon [Chat Room](#) - Build your own chat room, web server, mail client, or completely new Internet protocol (*coming...*)

Comments

You do not have permission to add comments.