Life is Tech!®

Back-to-School Checklist for Your Coding Curriculum

Learning to code benefits students of all ages. Here's a checklist to help you with your computer science/STEM curriculum, whether you're just getting started or reevaluating your current plan.

	Review the existing resources, tools and programs that you have available:
☐	Standards–What standards are required for each grade level? Do you need to provide standard-aligned lesson plans?
☐	Instruction–What will be the frequency and duration of instruction that is most appropriate for your students?
☐	Coding knowledge–Do you have any previous coding experience or a computer science background? If not, choose a program that enables any teacher to facilitate and that provides support when you need it.
☐	Teaching style–For more open-ended instruction, choose a program with scaffolding and support so that kids don't get stuck.
☐	Learning style–How do your students like to learn? This can determine the most appropriate activities.
☐	Programs/Platforms–What kind of coding curriculum do you have available? Do you have a choice of programs? What is your budget?

	Teaching coding–there isn't one "right" way to teach These are coding basics to cover:
☐	Vocabulary reference–Learning coding is like learning a new language. Provide a list of terms that is easy to access.
☐	Digital literacy–Provide guidance on logins and internet safety
☐	Sequencing–Concept that is used in ELA and other subjects so can use that to help students understand it in the context of coding
☐	Computational thinking–Show students how they can break problems down into manageable chunks. Four key skills in computational thinking: decomposition, pattern recognition, pattern abstraction, and algorithm design
☐	Debugging–How to solve a problem
☐	Branching/conditions–Use real-world examples to keep students interested
☐	Help students develop a coding mindset to build creativity and resilience–it's ok for them to make mistakes and try again.

	Build a coding curriculum to maximize engagement:
☐	Include games you can do with and without a computer
☐	Make coding entertaining with interactive material
☐	Break projects into micro-steps and think of analogies for each step
☐	Build hints and “checkpoints” into coding games or lessons beforehand.
☐	Make use of peer-to-peer helpers–More advanced students can further test their knowledge by helping other students.
☐	Use project-based learning (PBL) for a more hands-on experience. Make real-world connections to reinforce coding concepts

Disney Codeillusion is Built for Educators, Designed to Inspire
With dedicated 1:1 support, a fully mapped, standard-aligned curriculum, and robust professional resources to ease training and onboarding, you can finally focus on what’s most important: creating meaningful connections with your students. DIsney Codeillusion provides differentiated instruction, with a virtual mentor so students don’t get stuck. Turn-key implementation means that even teachers with no coding experience can facilitate classes.

With Disney Codeillusion, students will learn four different languages:

- HTML
- CSS
- JavaScript
- Processing

They will then use these coding skills to create their own websites, design media art, build games, and more. Bring the magic of Disney Codeillusion to your students. Schedule a custom demo to learn more.

https://codeillusion.io/pages/coding-for-schools