A Short Introduction to EIPM

What is EIPM?

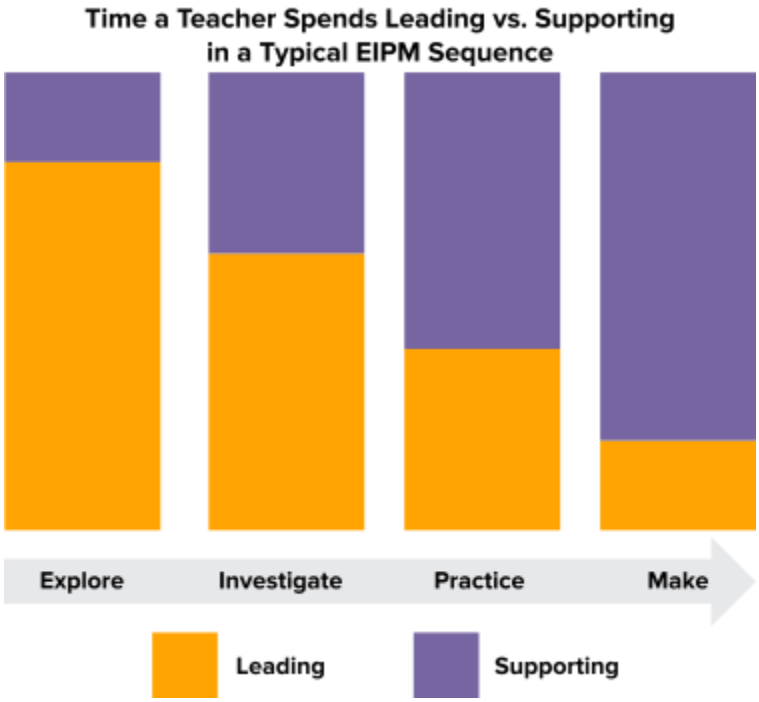
EIPM is a structured approach to teaching programming in Code.org's CSP Curriculum. It's designed to meet the needs of diverse learners, encourage collaboration, support independent creation, and clarify the role of the teacher throughout the learning process. Each letter represents a different type of lesson (E - Explore, I - Investigate, P - Practice, M - Make) which are taught in sequence for each major programming construct. A typical programming unit will feature 2 or 3 EIPM sequences, followed by a unit project and a multiple choice assessment. The visual below shows that structure for Unit 5 - Lists, Loops and Traversals, with each square being a separate lesson.

Why did you develop EIPM?

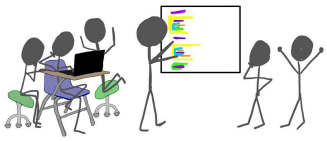
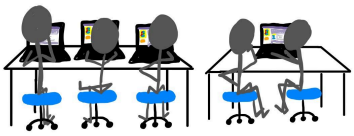
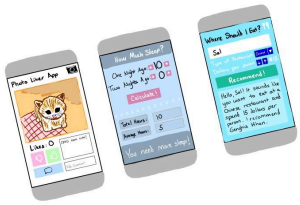
The biggest motivator in developing EIPM was direct feedback from students and teachers. While they had many positive things to say about our programming units, we also heard some consistent concerns. Due to the way our programming units are designed, students were often working in isolation, teachers lacked clarity on their role, and classrooms would arrive at the end of the unit lacking the skills to take on major independent projects. We took this feedback, consulted research and experts in the field of computing education, and created a pathway for introducing programming concepts that is more collaborative, clarifies the teacher's role, and more intentionally prepares students to create programming projects independently.

The Teacher's Role in the Classroom

The EIPM structure provides more clarity on the way students are expected to learn in each lesson and your role in facilitating that learning. In the Explore and Investigate lessons you will spend relatively more time leading group activities and class-wide discussions. In the Practice and Make lessons you will spend more time circulating the room to provide support to individuals and small groups. This predictable transition clarifies your role in any individual lesson and encourages students to work with greater independence as they develop mastery over a new concept. The graphic to the right shows the gradual transition from leading students through activities to supporting students working more independently.



A Summary of Each Lesson Type

	Overview	Goal
Explore 	Students explore the new concept through a teacher-led hands-on group activity. • Typically uses physical manipulatives • Teacher leads with support of slides and activity guides	Students begin to develop a shared mental model and understand the main ideas of the new concept.
Investigate 	Students investigate two or three sample programs that use the new concept. • Close-reading of working programs • Teacher-led discussions • Tasks to modify apps	Students become comfortable reading and modifying programs that use the new concept.
Practice 	Students practice using the new concept through a scaffolded series of programming activities. • Students work independently or in pairs • Teacher introduces debugging practices at the beginning of the Activity and circulates the room during the lesson to provide targeted support.	Students gain confidence in writing and debugging programs that use the new concept.
Make 	Students make a target app for which they are given the screen elements but little to no starter code. • Students are provided high-level steps to break down the project • Teacher supports students by directing them towards notes, previous work, and debugging strategies practiced in earlier lessons.	Students are able to independently decide when and how to use the new concept in the context of a larger project.