

Coding Pillars



Learning.com’s EasyCode Pillars for grades 6-10 builds career-ready coding skills through hands-on and interactive, project-based lessons.

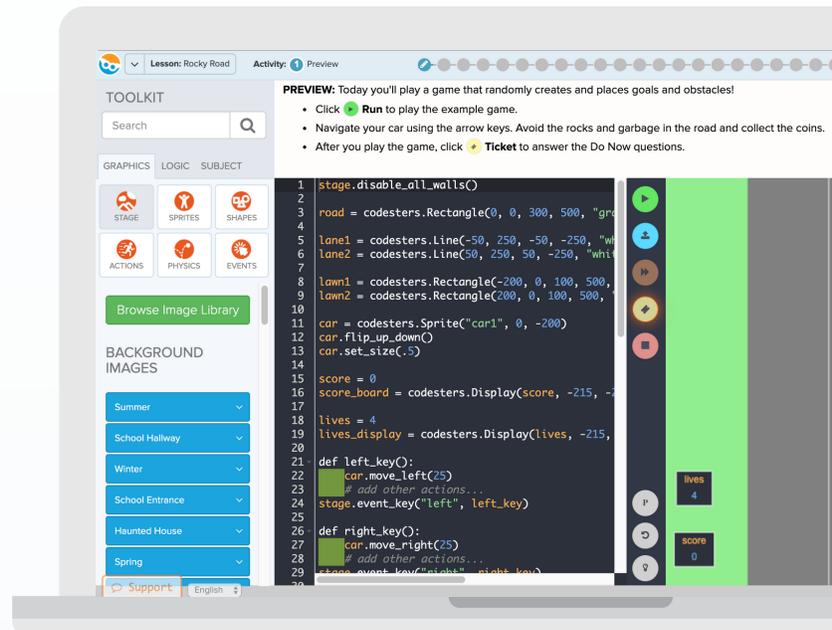
Students start as novices or as masters of our EasyCode Foundations curriculum; then, EasyCode Pillars intentionally scaffolds content through four mini-courses, so students build a strong basis with Python that they constructively build upon.

To meaningfully develop coding skills, the curriculum follows a “Build-Modify-Create” pedagogy:

- **Build:** Students gain familiarity with programming concepts by building prototype programs that they test and debug.
- **Modify:** Students deepen their programming skills by enhancing the program with new concepts.
- **Create:** Students create a new program from scratch in a capstone fashion.

Each mini-course within Pillars leverages this “Build-Modify-Create” process to help students practice, deepen understanding, and master critical concepts in Python.

In conjunction with Codesters, Learning.com’s EasyCode Pillars instructional content includes a series of hands-on and scaffolded projects that deepen student knowledge and skills in Python so they learn to approach problems computationally, think in terms of algorithms and automation, and design programs focused on the user experience.



For a comprehensive look at EasyCode Pillars through the Learning.com platform - including the built-in gradebook, dynamic progress monitoring, and district-wide reporting - contact us for a live demo, sales@learning.com.

ABOUT LEARNING.COM

Learning.com’s digital literacy curriculum enables schools to develop students’ technology skills throughout core instruction.

sales@learning.com • 800.580.4640 • [learning.com](https://www.learning.com)

MINI-COURSE 1

Intro to Codesters

Big Idea: Interactive programs are comprised of elements that control everything from visuals to reactions based on user input.

Skills

- Systematically test and refine programs using a range of test cases.

Key Topics

- The Stage
- Background Images
- Sprite Motions and Actions
- Shapes and Drawings
- Click Events
- Keyboard Events



MINI-COURSE 2

Intro to Python Part I

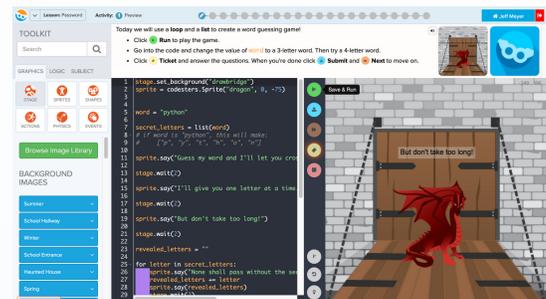
Big Idea: Simplify the coding process for a program by breaking it down into smaller problems or steps.

Skills

- Create clearly named variables that represent different data types and perform operations on their values.
- Create procedures with parameters to organize code and make it easier to reuse.
- Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.
- Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

Key Topics

- Variables
- Loops
- If-Statements
- Lists
- Functions
- Events



ABOUT LEARNING.COM

Learning.com's digital literacy curriculum enables schools to develop students' technology skills throughout core instruction.

sales@learning.com • 800.580.4640 • [learning.com](https://www.learning.com)

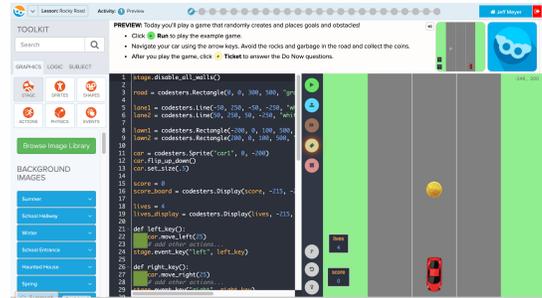
MINI-COURSE 3

Intro to Python Part II

Big Idea: Programs are meant to be tested and refined, whether based on user feedback, performance, or general enhancements.

Skills

- Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
- Create procedures with parameters to organize code and make it easier to reuse.
- Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.



Key Topics

- Functions
- Indexes
- Events
- Data Types
- Program Design
- Game Mechanics

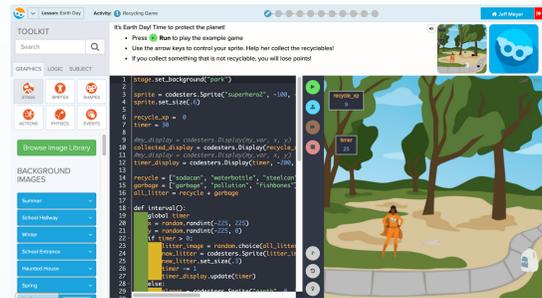
MINI-COURSE 4

Game Design

Big Idea: To design programs, focus on the needs of the end user, define the problem being solved, think creatively about solutions, build and test prototypes, and improve solutions based on these design principles.

Skills

- Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
- Incorporate existing code, media, and libraries into original programs, and give attribution.
- Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.



Key Topics

- Taking Turns
- Moving Objects
- Collisions
- Random Numbers
- Avoiding Hazards
- Getting Points

ABOUT LEARNING.COM

Learning.com's digital literacy curriculum enables schools to develop students' technology skills throughout core instruction.

sales@learning.com • 800.580.4640 • [learning.com](https://www.learning.com)

SUMMATIVE**21st-Century Skills Practice**

EasyCode Pillars promotes the “4Cs of 21st Century Skills”: Communication, Critical Thinking, Collaboration and Creativity. The 4Cs help students build “executive function skills,” including planning, organizing and strategizing. Combined, these skills empower students to become independent, life-long learners and promote the development of social-emotional intelligence.

Communication

- Document programs in order to make them easier to follow, test, and debug.

Critical Thinking

- Systematically test and refine programs using a range of test cases.
- Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

Collaboration

- Systematically design and develop programs for broad audiences by incorporating feedback from users.

Creativity

- Evaluate and refine computational artifacts to make them more usable and accessible.

ABOUT LEARNING.COM

Learning.com’s digital literacy curriculum enables schools to develop students’ technology skills throughout core instruction.

sales@learning.com • 800.580.4640 • [learning.com](https://www.learning.com)