

Rutgers University, Center for Effective School Practices

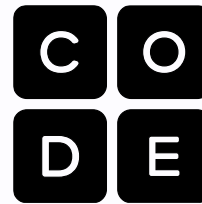


Evaluating & Selecting Computer Science Curricula

Making Informed, Equitable Decisions



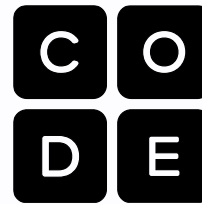
Resource Overload!



CodeHS



Resource Overload!



CodeHS



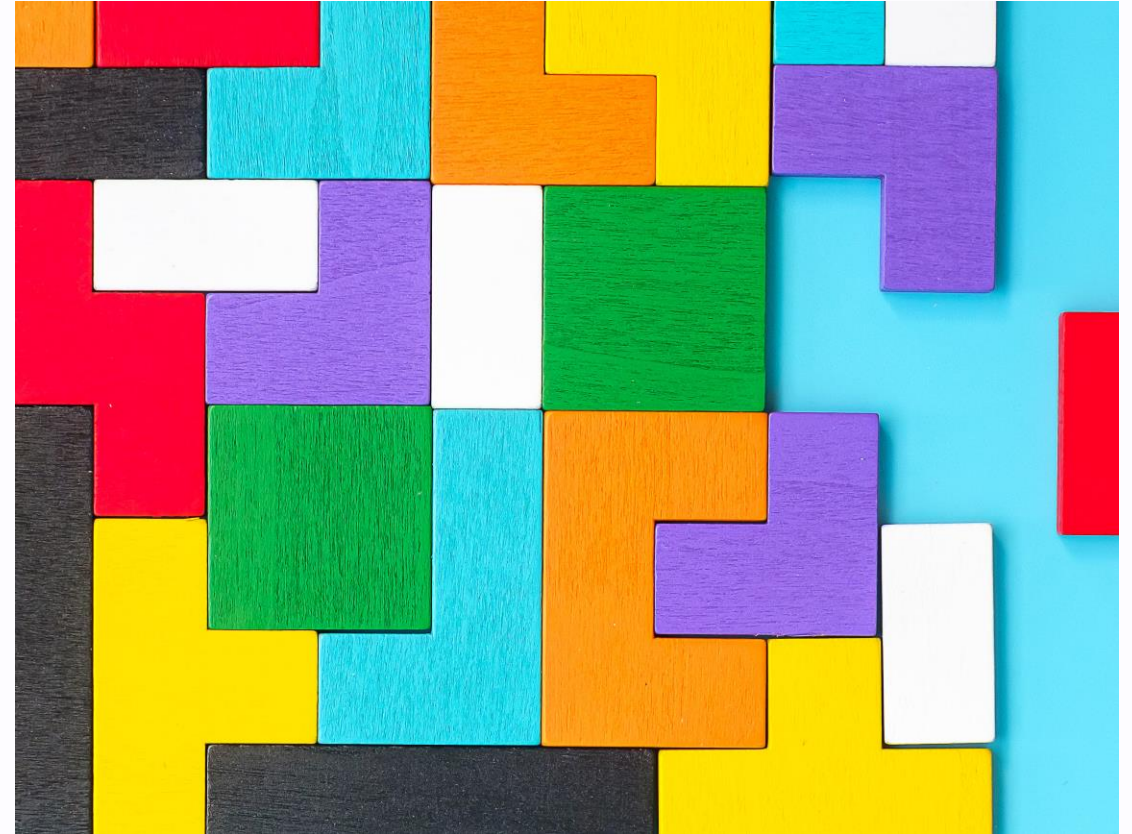
Danger!

Decision Paralysis, Arbitrary Choices

What can we do?

Use **computational thinking** skills!

Decompose our problem,
then *systematically* solve it.



Think **broadly** about curricular resources, but **specifically** about your needs.

- Full-course curricula
- Units or project plans
- Daily lesson plans
- Specific worksheets & activities
- Slides
- Videos
- Guest speakers
- Discussion topics

Replace something.

Changing a lesson, unit, or an entire course.

Supplement something.

Providing additional context or extensions on existing topics.

Add something new.

Adding a new topic, lesson, or project to a course.

Think **broadly** about curricular resources, but **specifically** about your needs.

- Full-course curricula
- Units or project plans
- Daily lesson plans
- Specific worksheets & activities
- Slides
- Videos
- Guest speakers
- Discussion topics

Replace something.

Changing a lesson, unit, or an entire course.

Supplement something.

Providing additional context or extensions on existing topics.

Add something new.

Adding a new topic, lesson, or project to a course.

Takeaway: Identify your need.

Approach resources with specific, solvable goals in mind.

Using curricular resources is a process.

Searching

- Identifying a need
- Stating resource goals (type, level, format, topic)
- Locate potential matches



Evaluating & Selecting

- Assessing feasibility
- Exploring content relevance
- Determining contextual relevance



Adapting & Implementing

- Adapting structure, content, and pedagogies
- Piloting, assessing, and modifying – repeat!



Using curricular resources is a process.

Searching

- Identifying a need
- Stating resource goals (type, level, format, topic)
- Locate potential matches



our focus! Evaluating & Selecting

- Assessing feasibility
- Exploring content relevance
- Determining contextual relevance



Adapting & Implementing

- Adapting structure, content, and pedagogies
- Piloting, assessing, and modifying – repeat!



Content: Relevance Factors



Difficulty of content



Familiarity with content



Alignment with
objectives/standards



Enduring
understandings

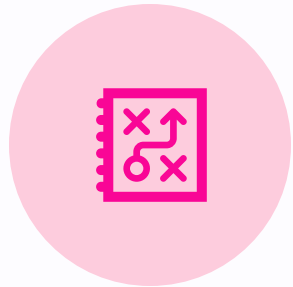
Context: Relevance Factors



Backwards & forwards connections



Examples and analogies used



Instructional methods used



Activities used

Feasibility: Logistic Factors



Materials &
Equipment



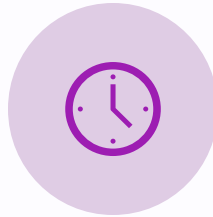
Device
Compatibility



Lesson & Class
Duration



Language



Prep Time



Grades &
Assessments



Cost

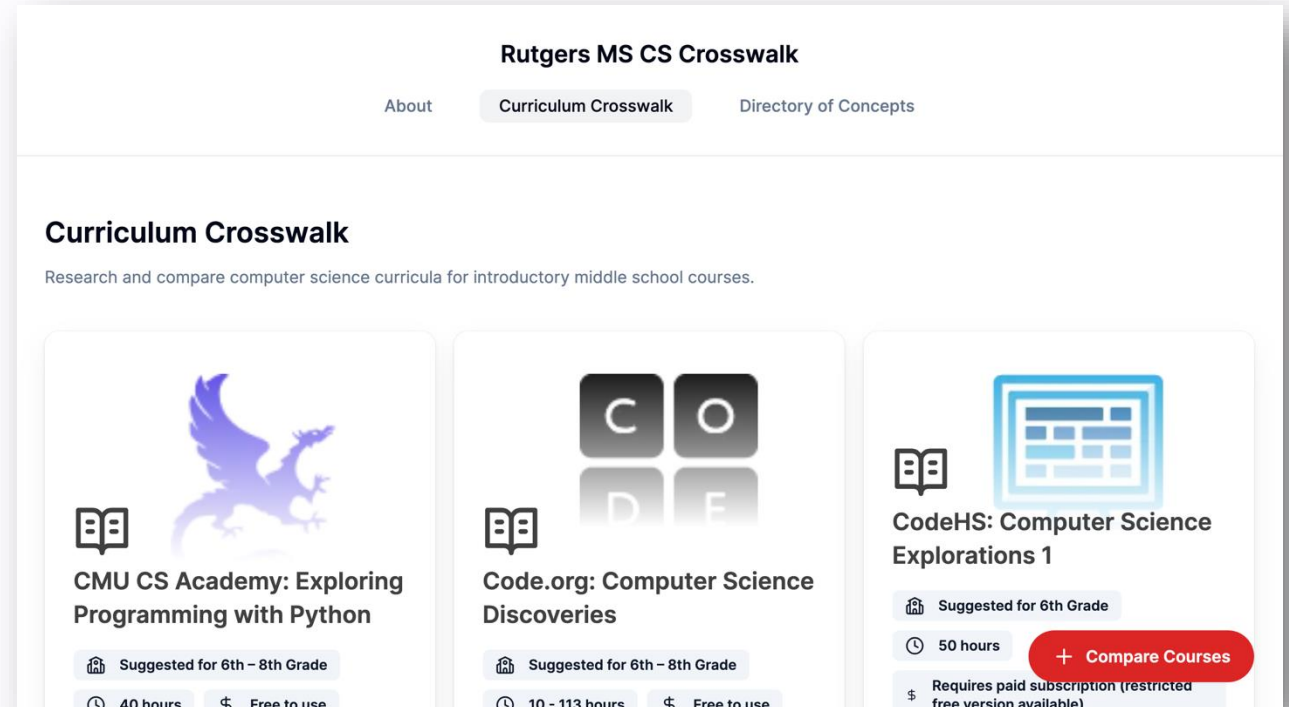
Highlighted Resource

The Middle School CS Crosswalk

Free online resource that provides detailed information on eight middle school CS curricular options.



go.rutgers.edu/CSCrosswalk



Spotlight:

Design & Interactivity

From the beginning of the course, students are learning programming concepts through completing interactive artifacts like games or animated stories. This gives them high-quality products from their work to be proud of, opposed to simple programs they might create in other courses. However, students often work only with a piece of the puzzle and do not create every product they submit in its entirety.

What makes it great:

With introductory coding knowledge, it is difficult to create visually appealing and engaging products. By allowing students to program small pieces of larger programs, they are able to produce more flashy and interactive products from their work that they can be proud of and share with others.



Not an endorsement!

Expert Take

This course is designed with the understanding that teachers may not have a background in Computer Science or may be teaching Computer Science for the first time. MakeCode Intro to CS provides simulated environments that allow students to test and debug their code without the need for physical hardware, which proves invaluable when access to such devices is restricted. Utilizing a block-based, drag-and-drop proprietary language interface, this curriculum minimizes the focus on syntax, making it a good option for students with little or no programming background. The curriculum covers fundamental programming concepts applicable to any language, providing students with a solid foundation for future CS courses. For teachers interested in integrating physical computing into their CS curriculum, MakeCode Intro to CS offers a seamless option that doesn't necessitate prior knowledge or experience in Computer Science. This course option also serves as an excellent bridge between coding and hardware.



Not an endorsement!

Applications of Computing

Curriculum	CMU CS Academy	CodeHS	Code.org	Harvard Creative Computing	Google CS First	Project STEM
Storytelling		✓	✓			✓
Data Analysis/Science			✓			
Careers in Computing		✓	✓			
Impacts/Daily Life		✓				

Applications of Computing

Curriculum	CMU CS Academy	CodeHS	Code.org	Harvard Creative Computing	Google CS First	Project STEM
Storytelling		4.1, 4.3	Unit 5		N/A	2, 5
Data Analysis/Science			Unit 5		N/A	
Careers in Computing		3.7	Covered Throughout		N/A	
Impacts/Daily Life		3.1			N/A	

Not an endorsement!

Tips & Closing Remarks

Be picky!

Resources are abundant – you can afford to be selective. Having a clear understanding of what you're looking for will be your GPS!

Adapt, then use!

Select resources either that work, or that you can *make* work. Decouple resources from their curricula. Customize activities, lessons, examples - make them yours!

It's a process!

Information overload is **real**! Focus on what is in your control, and look for warning signs of hasty decisions. Don't be afraid to ignore the noise.

Thank you!

Evaluating & Selecting Computer Science Curricula
Making Informed, Equitable Decisions

We ❤️ connections; let's stay in touch

w: cesp.rutgers.edu

e: cesp@gse.rutgers.edu

p: (732) 564-9100 x23



Vivian Allen
associate director

David Amiel
research associate

Cindy Blitz
executive director

Our Team @ Rutgers University!
Center for Effective School Practices



Thank you!

Evaluating & Selecting Computer Science Curricula
Making Informed, Equitable Decisions

We ❤️ connections; let's stay in touch

w: cesp.rutgers.edu

e: cesp@gse.rutgers.edu

p: (732) 564-9100 x23



Vivian Allen
associate director

David Amiel
research associate

Cindy Blitz
executive director

Our Team @ Rutgers University!
Center for Effective School Practices

See me for goodies!