



CS FOR ALL

RePublic Schools

The CS for All model enables all middle-school students to access high-quality computer science so that they can pursue computer science courses in high school and beyond.

Scan QR code with your phone's camera to access this content online.



OVERVIEW

The RePublic Computer Science for All Program (CS for All) is dedicated to changing what's possible for all students by providing computer science as a stand-alone subject to every middle school student and preparing them for higher-level computer science courses and certifications. Through access to a high-quality computer science curriculum and high-quality instruction and experiences, the program prepares scholars to be successful now and in the future. Students build technical skills in programming and computer literacy as well as soft skills like leadership, teamwork, communication, problem-solving, a strong work ethic, flexibility/adaptability, and interpersonal skills. CS for All prepares scholars to seek a degree in computer science, or any other degree, and pursue a career in a computer science field.

RePublic's free vertically K-12 aligned Computer Science program is adaptable, allows for student choice, and is aligned to the AP Computer Science Principles curriculum. In the 5th-8th grade program, which is featured here, teachers get online self-paced professional development and have an option to pay for in-person training. This entry focuses on RePublic's middle-school program. RePublic also offers curricular resources for elementary and high-school students.

► [Drew's Story – A RePublic Story](#) [RePublic Schools Computer Science Overview](#)



What Makes This Model Innovative?



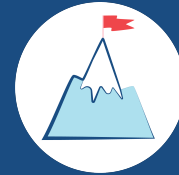
Rigorous Learning

All students experience computer science and computational thinking at each middle-school grade level, with exposure to many different coding languages.



Relevance

Students focus on problem-solving with open-ended tasks, often bringing concepts from other classes into their computer science projects.



High Expectations with Unlimited Opportunities

All students access quality computer science learning opportunities, opening the door to pursue computer science in college and career.

DESIGN

Goals

The model is designed to build technical skills and soft skills in order to pursue computer science in high school.

Computational Thinking and Computer Science

At each grade level, students progressively develop their computer science skills across all domains. Students learn how to develop meaningful artifacts in several programming languages, including HTML, CSS, JavaScript, Mobile App Design, Game Design, and Python.

Collaboration and Communication

Through the use of partner and group projects, students build and practice communication and collaboration skills.

Digital Citizenship and Computer Literacy

Students develop digital literacy skills including keyboarding, word processing, and presentation skills.

Problem-Solving and Navigating Learning

Through projects and activities, students develop problem-solving skills that can be applied across disciplines and settings.

Experience

In the CS for All model, all students engage in a vertically aligned set of stand-alone computer science courses. Units are designed to progressively move students from learning new concepts and skills to developing their own projects, games, and websites using a variety of programming languages. Lessons feature different pedagogical approaches, including direct instruction (e.g., I Do, We Do, You Do) and inquiry-based instruction (e.g., 5E Model). Each unit contains multiple cycles in which students experience introductions to new material, challenge days, project days, and presentation days.

The vertically aligned curriculum has different course units from 5th through 8th grade.

- In Year 1 (grade 5), students engage with MIT’s Scratch to learn storytelling, animation, and both basic and advanced game design.
- In Year 2 (grade 6), students interact with CodePen to learn introductory web development skills, writing HTML and CSS, as well as accessibility design to deepen their web development skills.
- In Year 3 (grade 7), students interact with Code.org's AppLab to learn design elements of mobile applications, and intermediate and advanced application development.
- In Year 4 (grade 8), students learn JavaScript to build on their web development skills, allowing for user input and engagement, and begin learning about physical computing using robotics.

Curriculum Map

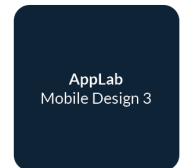
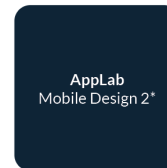
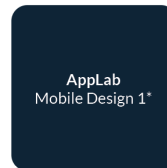
Middle School Year 1



Middle School Year 2



Middle School Year 3



Middle School Year 4



Introduction to New Material

Each unit begins with and contains multiple lessons introducing students to new material, for approximately 14 days throughout each unit. On the first day, students learn new programming skills and concepts, and then practice those skills and build on other skills on day 2. Additionally, students sign up for new applications and tools (if needed), learn definitions of key concepts, understand how they’re used in context, practice using tools and concepts, and are assessed on what they’ve learned.

Introduction lessons are primarily taught by the teacher to the whole class with students completing their work independently. Teachers begin the lesson by engaging the class in an

online “Do Now” to activate prior knowledge either aloud in a class discussion and/or by completing a prompt online. Teachers then provide students with limited time to explore independently the concept or tool through the digital interface. Students are prompted to analyze code examples online to help them understand how the skill or concept works. After this exploration, teachers give students ample time to apply the new skill or concept independently or with a shoulder partner through a series of tasks asking them to create various tags, strings of code, or other digital outputs. The lesson then ends with the teacher prompting students to complete a check for understanding online, allowing the teacher to measure and respond to student mastery.

For example, in the 6th grade web design lesson on text tags, situated in a unit on website design using the programming languages HTML and CSS, students learn the definition of text tags in HTML, learn how text tags are written, and then practice typing tags on their own. The lesson starts with a Do Now, introducing students to the concepts of text tags in HTML and asking them to answer a few brief questions online about what they notice in the code provided. Students then have a few minutes to explore in partner pairs the definition for text tags and how text tags work, culminating in a brief whole class discussion. The students then have time to explore independently by typing their own tags online before the teacher gives them a check for understanding to test what they’ve learned about tags. [Intro to Tags Lesson Plan](#)

Explore

Element Sandwich – 6 minutes

Directions

Show scholars the structure of a tag in question one, and then give them one minute with their shoulder partner to try to type that first tag. Write the correct response on the board and show scholars where on the keyboard the slash and the angled brackets are. Next, show scholars the element sandwich and give them one minute with their shoulder partner to answer the question. This image is another way for students to think about text tags. Another real world connection is to compare text tags to a door. Just like when you open a door, you also need to close it, when you open a text tag, you also need to close that. CodePen does not automatically close tags for scholars, so it's easy to forget. If scholars learn proper syntax now, they will be able to debug their projects more efficiently in the coming lessons.

1 2 minutes - Answer questions in shoulder partners

2 4 minutes - Whole group discussion

Teacher Tips

For younger grades, it is helpful to have scholars show the difference between an opening tag and a closing tag with their hands. For example, a closing tag would look like two hands in the shape of greater than and less than symbols. An opening tag would look like one hand at a slant to represent a backslash.

Student View

Question 1: Tags tell the browser how to display text or graphics.

Everything in HTML needs to be within tags.

Element		
Opening Tag	Text	Closing Tag
<h1>	Hello World	</h1>

Type an h1 tag using the words "Computer Science is cool."

Question 2:

Element Sandwich



<bread>Lettuce and meat</bread>

<tag>Text</tag>

How is a closing tag different from an opening tag?

Exemplar Answers

<h1>Computer Science is cool</h1>

A closing tag has a backslash and an opening tag doesn't.

Challenge Days

Challenge days commonly follow the Introduction to New Material days and involve students taking a few days' worth of new ideas and combining them to tackle a set of programming challenges. This will be done either independently or in teams, depending on the lesson. Each challenge increases in difficulty and often builds up to how multiple skills work together in actual applications, websites, or other artifacts.

Challenge days begin with students completing a Do Now to review a prior concept or activate prior knowledge, usually by completing a series of questions online. Then, the teacher introduces an exemplar or rubric to the class, followed by a brief discussion. The bulk of the lesson is devoted to giving students independent opportunities to apply their understanding and use of the tool to a series of online challenges to practice using the tool, culminating in a brief check for understanding.

For example, a 6th grade web design Challenge day lesson plan combines HTML and CSS together in one challenge, so that students can differentiate between the two and feel comfortable with both. Students start with a Do Now error-correcting activity they complete independently, and then briefly discuss as a whole group. The teacher then has students explore a problem-solving rubric for web design and answer a few questions independently before discussing them with the whole group. Students then have the bulk of the lesson to work on a series of challenge prompts in which they practice using the problem-solving rubric to identify errors and correct them in both HTML and CSS. The challenges build upon each other and lead students toward mastery of the skills needed for the final project. At the end of the lesson, students complete a check for understanding to demonstrate their learning. [Web Design Challenge Day Lesson Plan](#)

Found Pet Challenges - 37 Minutes

This challenge combines both HTML and CSS challenges so scholars can learn to switch between the two languages. A big focus on this lesson will be identifying errors, so celebrate scholars who are correcting their own errors or helping their shoulder partner correct the error. Have each student create their own project, but they should be allowed to ask their shoulder partner a question if they have one.

Student View	Exemplar Answers	Student View	Exemplar Answers
<p>You are exploring in your back yard and have come across a strange creature in your backyard. You decide you want to create a website to help the animal find its home.</p> <p><i>Students have access to an embedded CodePen Editor</i></p> <p><i>Hint:</i> You can use W3 Schools with these challenges.</p> <p>Challenge 1: Select ANY animal you want. Create an h1 tag to create a title representing this animal.</p> <p>Challenge 2: Create an h3 tag describing the location in which you found this animal.</p> <p>Challenge 3: Create a p tag with a description of their habits. Be detailed, how are they responding to captivity? Elaborate in detail in 4-6 sentences.</p> <p>Challenge 4: Create a p tag with a detailed description of their looks. This should be 2-3 sentences.</p>	<p><u>Exemplar</u></p> <p>C1: <code><h1>Dog Found</h1></code></p> <p>C2: <code><h3>The dog was found on Barnes Street.</h3></code></p> <p>C3: <code><p> The dog was found playing in a mud puddle in my back yard. She was standoffish at first, but is now extremely friendly. She limps on her right hind leg. She has eaten a lot.</p></code></p> <p>C4: <code><p>This dog has a short tan coat with black spots. She has a collar, but the ID tag has fallen off.</p></code></p>	<p>Challenge 5: Create an h4 tag with information about how to get in contact with you to retrieve their found animal.</p> <p>Challenge 6: Change the color of the background.</p> <p>Challenge 7: Change the background color of each section to a different color.</p> <p>Challenge 8: Change the font color of each section to a color that works with the background colors you have chosen.</p> <p>Bonus Challenge:</p> <p>Use the W3 schools to figure out how to create a border, change the size, color, and style of the border for your text tags.</p> <p><i>Use the Problem Solving Rubric if you get stuck.</i></p> <p>Paste the link to your website.</p>	<p>C5: <code><h4> Please contact (insert scholar's name) at (insert scholar's email) if this dog is yours.</code></p> <p>C6: <code>body{ background-color:tan; }</code></p> <p>C7&8: <code>h1{ background-color:red; font-color: white; }</code></p> <p><code>h3 { background-color:red; font-color: white; }</code></p> <p><code>p{ background-color:red; font-color: white; }</code></p>

Project Days

Project days are days spent working on a project, the primary way in which students are assessed. Every unit has two projects, one shorter project (~3 days) and one longer project (~6 days), which happen at different times throughout the unit. Projects can be independent or collaborative, depending on the content.

Project days typically include a launch day, where students are introduced to the project and its rubric, and 3–4 creation days, where students use creativity and work independently on an artifact. Some projects may have use or value outside the classroom (i.e., a computer game, website, animated story). Projects serve as unit assessments.

This 6th grade web design project gives students the opportunity to tell a historical figure's story through the creation and design of a website. Students begin the project by receiving an overview of the project and rubric and participating in a quick whole group discussion on both. Students then have extended time to individually brainstorm ideas online with a series of prompts and potentially get individual feedback from the teacher on their ideas. On the creation days, students start with a Do Now, and then work alone or with the help of peers on selecting colors, choosing appropriate tags, adding images, and including facts about their figure for others to see in their finished project. They can seek help and feedback from the instructor and largely work independently. The teacher introduces a tip each day to support students as they complete their projects. [Web Design Project and Pitch Day Lesson Plans](#)

Project

Day 4 - Creation & Reflection Day

Directions

This Do Now reminds scholars how to add a border to an element. Model for scholars if they forgot.

For the rest of this day, scholars should be working on their projects. They should be allowed to seek help from peers and the instructor. The instructor should spend this time helping scholars and challenging scholars who are finished to add more to their websites. Make sure that by the end of class scholars have completed their entire website.

1 5 minutes - Do Now

2 30 minutes - Work time

3 10 minutes - Reflections

Student View

Question 1: What is the correct CSS code to add a thin, solid, green border around the h1 tag below?
<h1>How to Make Spaghetti</h1>

Question 1: List the sequence of steps you follow when creating your Biography Website below.

Question 2: What was rubric item that you struggled to complete? How did you overcome this struggle?

Question 3: What is one thing you learned, that you didn't already know, about your Historical Figure?

Exemplar Answers

```
h1 {  
border-style: solid;  
border-color: green;  
border-width: 3px;  
}
```

Answers will vary.

Answers will vary.

Answers will vary.

Teacher Tip

Scholars will continue building their websites. By the end of day 4, scholars should have everything complete on their project, focusing on:

- UI elements: colors, images, borders
- Syntax Indentation and Spacing in your HTML document

If students finish early, challenge them to add another image or a link to their website. They can also edit their CSS, changing colors and fonts to add to the aesthetics of their website.

Presentations Days

Units end with a day where students present their final projects, whether independently, with a partner, in small groups, or with a group to the whole class. The goal is to prepare students for pitching their ideas and to collaboratively share each student's ideas and creations. Typically, students are placed in small groups with a structured protocol for presenting and giving feedback to presenters based on the project rubric, and they take turns presenting to their peers.

In this 6th grade presentation lesson plan, the teacher starts off by providing instructions and expectations for the pitch day. Students are then placed in groups of four and take turns pitching their websites and providing feedback to each other. The structure for pitch days commonly allows for 3 minutes to pitch their projects to their teammates, 2 minutes for teammates to critique these projects on the rubric presented at the beginning of the project and provide feedback, and 2 minutes to share feedback with teammates; each time, the presenter takes notes. With the remaining time, students can implement the feedback they were given or elect a member of their group to pitch their project to the whole class. The lesson plan provides structured time for the pitch and feedback, guiding questions, and a rubric, so that students can run their own pitch sessions without input from the teacher. [📄 Web Design Project and Pitch Day Lesson Plans](#)

Project

Day 5 - Pitch Day

Directions

On this day, students should be placed into groups of 4. They will each take turns pitching their websites and providing feedback.

- 3 minutes to pitch their projects to their teammates.
- 2 minutes for teammates to grade these projects on a rubric and provide feedback
- 2 minutes to share feedback with teammates and the presenter to take notes
- This should take 30 minutes, with 5 minutes to explain instructions and expectations.

For the remaining 10 minutes, you can either have scholars implement feedback they were given, or scholars can elect a member of their group to pitch their project to the whole class, completing as many groups as you can.

1

5 minutes - Instructions

2

40 minutes - Pitches and Feedback

Say

Today we will have the opportunity to review and give feedback on each other's' websites! Everyone in your group will present for three minutes. Following this, your team will have two minutes to ask questions and write down your feedback in the platform.

Then each person will share their feedback with the presenter, who will take notes in the note section of the platform.

Student View

Question 1: What do you like about their website?

Question 2: What would you change about their website?

Question 3: Rubric Grade

Exemplar Answers

Answers will vary.

Teacher Tip

Scholars may struggle with presenting to their peers. Push presenters around best practices of presenting: using an audible tone, making eye contact, using an energetic voice, describing the best parts of the project. Push observers to write down questions to ask after the presentations. It may also be helpful to discuss presentation norms with scholars.

Supporting Structures

Any school can implement CS for All into its existing school design, but it must make shifts in curriculums, schedules, and more to enable access to CS for *All* students.

Teachers use the CS for All curriculum and assessments that are vertically aligned with and scaffold to the AP Computer Science Principles course in high school.

Each project-based course is vertically aligned to the AP Computer Science Principles (AP CSP) course. All lessons relate directly to the five Big Ideas from the AP CSP framework, and all coursework incorporates components of analysis, presentation, and critique. [AP CSP Conceptual Framework](#) The programming languages, technical skills, and projects are intended to prepare students for the AP CSP exam and the curriculum is designed to be accessible and inspiring to students who may not initially have any background skills or prior motivation.



CURRICULUM, INSTRUCTION, & ASSESSMENT

RePublic's CS for All curriculum leverages existing online resources such as MIT's Scratch, CodePen, Code.org's App Lab, LEGO MindStorms, and Georgia Tech's EarSketch, to build engaging and interactive instructional units. [Scratch](#) [CodePen](#) [App Lab](#) [Lego MindStorms](#) [EarSketch](#)

The curriculum is focused heavily on training teachers to master the content and lessons to take advantage of teachers who now have a deep understanding of the material.

Assessments are aligned to the AP Computer Science Principles (AP CSP) course. Aligning the assessments in middle schools helps to prepare students for the create and exam portions of the course. Three different types of assessment, all aligned to the AP CSP exam, are used to collect data, close gaps, and track mastery and growth throughout the year.

- Diagnostic and End of Course exams: AP CSP aligned assessments to show student mastery and growth through the school year.
- Unit Quizzes: Occur four times per unit to show student mastery, allowing teachers to analyze data and close gaps.
- Unit Projects: AP CSP Performance Task aligned projects to show student mastery of programming and problem-solving skills.



ADULT ROLES, HIRING, & LEARNING

While RePublic's professional development is optional, it emphasizes the importance of teachers engaging in ongoing development.

Successful computer science classes are dependent on teachers implementing effective teaching practices and utilizing computer science-specific systems. Teachers must master practices behind entire classes of students using devices without losing focus on the generally

pedagogical practices like whole group discussion, rigorous questioning, and lesson timing.

To support and train multiple computer science teachers at each grade level, RePublic has focused on creating detailed curriculum documents that can be internalized by teachers with various content backgrounds and skills.

Teachers do not need computer science expertise to teach the lessons. CS for All includes content challenges to support teacher understandings. [CS Content Challenge for Web Design](#) However, states have different credential requirements for computer science teachers. [State Credential Policy Tracking](#) RePublic also offers customized professional development for teachers and consulting for school leaders (see Supports Offered to learn more).

Schools implementing the CS for All model must dedicate time in the student schedule for a stand-alone computer science course.



SCHEDULE & USE OF TIME

RePublic recommends that students in grades 5–8 take computer science as a stand-alone course 2–3 days a week, though some schools use the materials for fewer than 2 days a week. The units in this model can be rearranged to fit the school’s stated outcomes.

Timing is a key consideration in the curriculum. Some schools offer one computer science course each quarter or semester, swapping out different cohorts of students each iteration. Other schools have a group of students who take the class all year. While all schools and districts have fluctuating schedules, each unit is 32 instructional days at 50 minutes per day. Districts usually have an average of 38 days, so the curriculum has removed a few days for state testing, weather, and other scheduling obstacles.



FAMILY & COMMUNITY PARTNERSHIPS

Schools should build partnerships with local organizations to increase students’ exposure to real-world programming languages and projects from professionals who use computer science on a daily basis.

Schools are encouraged to pursue partnerships with their local chapter of Blacks in Technology so that students may work with a teacher who is also a professional in a computer science field. [Blacks in Technology](#)



Schools must provide enough devices, whether 1:1 or in a computer lab, allowing each student to have a computer and access to various coding programs.

TECHNOLOGY & INFRASTRUCTURE

Students learn block-based coding in Scratch and learn how to program in JavaScript using Code.org's App Lab. This allows students to have several years of working in the App Lab before developing their create task in the AP Computer Science Principles course. For web development and stand-alone JavaScript units, students use CodePen.

IMPLEMENTATION

Supports Offered

[RePublic Schools](#)' Computer Science for All offers the following supports to help implement its approach.

CS for All Curriculum

Free



RePublic offers the CS for All curriculum for free to all teachers and schools interested in teaching computer science to middle schoolers. The curriculum includes teacher-facing content challenges to help teachers master the skills embedded in unit projects prior to teaching them to students.

[Get in Touch](#)

Leader Professional Development

Free, Cost Associated



RePublic offers professional development sessions for leaders to develop their teachers' content knowledge and ability to teach computer science. RePublic offers detailed agendas with linked resources and activities for leaders wanting a turnkey solution to providing their teachers with support. Leaders pursuing live and/or customized training with RePublic pay a fee.

[Get in Touch](#)

Customized Support

Free, Cost Associated



Computer science specialists at RePublic also partner with school leaders to think through pathways and options that work for their schools. Light support includes thought partnership and customer support. Leaders wanting more in-depth, customized support pay a fee.

[Get in Touch](#)

Reach

150

Schools

51,000

Students

200+

Educators

6

States

Impact

At RePublic Schools, the flagship school where the CS for All model was designed and implemented, students are seeing strong success.

- 85% of the RePublic High School class of 2019 enrolled in college in fall 2019, compared to 60.9% of the Metro Nashville Public Schools class of 2019, and the national average of 66% for the class of 2019.
- 70% of all high-school students at RePublic passed the AP CSP exam in 2020; 60% of these students experienced CS for All in middle school.
- 67% of RePublic high-school students chose the CS or AP pathway in high school in the 2023–23 school year; 72% of these students experienced CS for All in middle school.

Contact

Allison Arth

Director of Computer Science

aarth@republiccharterschools.org

RESOURCES



Drew's Story – A RePublic Story

A RePublic student's story about her experience with computer science.



RePublic Computer Science Overview

An overview of RePublic's CS for All model and resources.

Explore
Element Sandwich - 6 minutes

Directions
Show students the structure of a tag in question one, and then give them one minute with their shoulder partner to try to type the first tag. Show the correct responses on the board and show students where on the keyboard the < and the > symbols are. Next, show students the element sandwich and give students five minutes with their shoulder partner to answer the question. This image is another way for students to think about the tag structure: real world connections. It is to compare the tag to a door. Just like when you open a door, you also need to close it. When you open a tag, you also need to close that. CodeItm does not automatically close tags for scholars, so it's easy to forget. If scholars learn proper syntax now, they will be able to debug their projects more efficiently in the coming lessons.

1 2 minutes - Answer questions in shoulder partners

Student View
Question 1: Tags tell the browser how to display text or graphics. Everything in HTML needs to be within tags.

Exemplar Answers
all Computer Science is cool!</h1>

Question 2: Element Sandwich

A closing tag has a backward slash and an opening tag doesn't.

Intro to Tags Lesson Plan

A sample 6th grade Introduction to New Material lesson plan.

Found Pet Challenges - 37 Minutes

This challenge combines both HTML and CSS challenges as scholars can learn to switch between the two languages. A big focus on this lesson will be identifying errors, so moderate scholars who are correcting their own errors or helping their shoulder partner correct the error, have each student create their own project, but they should be allowed to ask their shoulder partner a question if they have one.

Student View

Exemplar Answers

Challenge 1: Create an h1 tag with information about how to get in contact with you to receive their found animal.

Challenge 2: Change the color of the background.

Challenge 3: Change the font color of each section to a different color.

Challenge 4: Change the font color of each section to a color that matches with the background colors you have chosen.

Bonus Challenge:

Web Design Challenge Day Lesson Plan

A sample 6th grade Challenge day lesson plan.

Project
Day 4: Creation Day

Directions
This Do Now reminds scholars how to write a comment. Focus on the syntax of the comment. Model for scholars how to add a comment if they forget.

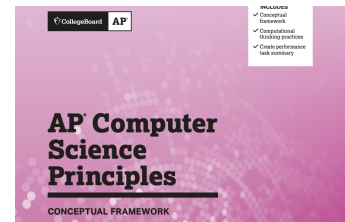
Question 1: Write a comment for this HTML.

Exemplar Answers
Assessors will vary. Sample: <!-- Hello to Make Spaghetti! -->

1 5 minutes - Do Now

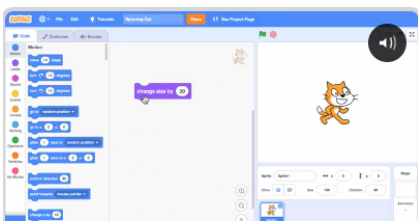
Web Design Project and Pitch Day Lesson Plan

Sample 6th grade Project and Pitch Day lesson plans.



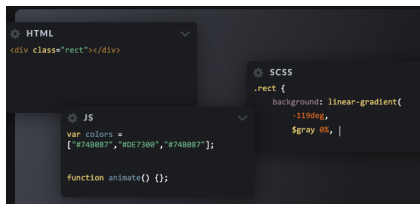
AP CSP Conceptual Framework

College Board's AP Computer Science Principles framework and computational thinking practices.



Scratch

Coding language for digital stories, games, and animations.



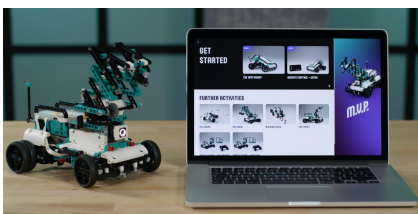
CodePen

Social development environment for HTML, CSS, and JavaScript code.



App Lab

Programming environment to make simple apps.



LEGO MindStorms



EarSketch

Web Design 1 Challenges - HTML Tags, CSS Properties, and Attributes

Challenge 1: Intro to HTML
Challenge Description: Recreate the following websites to test your skills and use of resources using HTML.
Skills Covered: HTML, text tags.

Missing Dog

Description: Don't forget to include the alt attribute for the missing dog website.

Planning Space:

Challenge 2: Intro to CSS
Challenge Description: Recreate the following websites to test your skills and use of resources using HTML and CSS.
Skills Covered: CSS properties
Hint: Use w3schools.com to find the necessary tags and

Planning Space:

Programmable robots utilizing Lego blocks.

Music composition and production coding platform.

[CS Content Challenge for Web Design](#)

Content challenges to support teacher understandings of computer science.

State Tracking 18 Policies PUBLIC

18 Policies | State Plan | Standards | Funding | Certification | Performance | State ID | Request ID | Goal | Impact ID | Draft/Approved

Copy and share these 18 policies | [https://transcend.org/CS_tracking_18_policies.html.pdf](#)

Table with 18 columns: State, State Plan, Standards, Funding, Certification, Performance, State ID, Request ID, Goal, Impact ID, Draft/Approved. Rows list various states including AL, AK, AZ, AR, CA, CO, CT, DC, DE, HI, IL, IN, IA, KS, KY, LA, MA, MD, MI, MN, MO, MS, MT, NE, NH, NJ, NY, NC, ND, OH, OK, OR, PA, RI, SC, SD, TN, TX, UT, VA, VT, WA, WI, WY.



[State Credential Policy Tracking](#)

Spreadsheet of computer science credential requirements by state.

[Blacks in Technology](#)

Organization focused on increasing Black workers in technology.