

ScriptEd Student Outcomes

Foundations Curriculum (Year One)

ScriptEd equips students in under-resourced schools with the fundamental coding skills and professional experiences that together create access to careers in technology. Students who complete ScriptEd’s Foundations curriculum will achieve mastery of the skills listed below. These skills and concepts are matched with the standards developed by the Computer Science Teacher’s Association, which can be found at <https://csta.acm.org/Curriculum/sub/K12Standards.html>.

Outcome	Concept & Standards	Skill / Indicator
A. Creating and Organizing Static Web Pages (Students who demonstrate mastery of these skills will have achieved proficiency in creating a web page using HTML markup and CSS.)	HTML Pages CPP.L2-03: CPP.L3A-01	Student can employ HTML body / head tags.
		Student can assign tags with an ID or class to differentiate it from other tags of the same name.
		Student can identify and compare the separation of structure and style in the creation of web pages.
	Page Content CPP.L2-03: CPP.L3A-01	Student can use appropriate tags to display text, images, forms, titles, tables, buttons, and links on a webpage.
		Student can employ a div tag to define a section in an HTML page
		Student can use the nesting structure of web pages to group elements and control page flow.
	Styling CPP.L2-03: CPP.L3A-01	Student can use CSS selectors to style elements by tag, ID and/or class.
		Student can use external libraries like Bootstrap to make a webpage responsive.
	Wire Framing CPP.L2-03: CPP.L3A-01	Student can construct a wire frame for a web page using HTML.

**B.
Using a Programming
Language (JavaScript)
to Solve
Programming
Problems and Create
Interactive Web
Applications**

(Students who demonstrate mastery of these skills will have achieved proficiency in solving programming problems using the JavaScript programming language. Students will be able to use these skills to create interactive web pages.)

<p>Variables CPP.L2-05 I</p>	Student can create, name, and assign values to variables.	
	Student can explain variable creation, reassignment, and scope.	
	<p>Math Operators CPP.L2-05 I</p>	Student can employ math functions such as “pow”, “rand”, “sqrt”, “ceiling”, and “floor” to manipulate numbers.
		Student can use JavaScript operator precedence to predict the effect of an expression.
		Student can recognize augmented assignment operators (i+=1) and can decompose them to simpler syntax.
		Student can generate and control the range of random numbers to produce complex behavior in interactive pages.
		Student can debug “null” , “NaN” or “undefined” inputs to calculations when solving programming problems.
	<p>JavaScript Types CPP.L2-05 I</p>	Student can identify differences between numbers, strings, and arrays, and can infer when it is appropriate to use each type.
	<p>Boolean/ Comparisons CPP.L2-05 I</p>	Student can name and appropriately implement boolean values to solve programming problems.
		Student can employ conditional statements (“if”, “else”, “else if”) to solve programming problems.
		Student can use logical operators (&& and and !) within “if” statements.
	<p>Strings CPP.L2-05 I</p>	Student can use string functions to find the length of a string, split a string, concatenate a string, cast a string to a number, find the index of a character, and make strings uppercase and lowercase.
		Student can compare strings, accounting for case and whitespace, and use the results in conditional statements.
		Student can find all the instances of a character in a string.
	<p>Loops and Control Flow CPP.L2-05 I</p>	Student can use “while” and “for” loops and avoid infinite loops.
		Student can use “break,” and “return” statements.
<p>Arrays CPP.L2-05 I</p>	Student can create arrays: student can assign each element of an array to a specific index; student can access elements of the array and change them; student can find the length of an array.	
	Student can loop through an array in order to repeat operations across each element of an array.	
<p>Event-Driven</p>	Student can trigger JavaScript behavior from user interaction.	

	Behavior CPP.L2-05 I	Student can modify page contents using JavaScript and/or a JavaScript library.
	Functions CPP.L2-05 I	Student can create functions, name them appropriately, and use them to solve programming problems.
	External Libraries CL.L3B-01	Student can access external code libraries to utilize the pre-written functions within said library
C. Collaborative Computing (Students who demonstrate mastery of these skills will have achieved proficiency in collaboratively designing, developing, and publishing projects using technology resources, in keeping with industry standards).	Version Control CL.L3A-02 CL.L2-01 CL.L3B-01	Student can use the command line to stage files for commit, write commit messages and push their code to a remote repository; student can use git to collaborate and can resolve merge conflicts.
		Student can send and receive pull requests using the GUI found on GitHub
	Teamwork & Collaboration CL.L3A-01 CL.L3B-01 CL.L3B-03 CL.L2-02 CL.L2-04	Student can divide responsibilities for a project; student can integrate others' code with his or her own.
		Student can plan a project with teammates (using wireframes, specifications /requirements, and pseudocode).
		Student can maintain a design by contract relationship with other team members; student can write clear code; student can explain code by including clarifying comments in their code.
D. Career Readiness	Professionalism	Student has a professional e-mail address and uses professional language in e-mails.
		Student can appropriately use industry-standard vocabulary when speaking about technology and software development.
		Student demonstrates a high level of professionalism when asking questions and seeking out answers.
	Industry Exposure	Student can identify a range of career opportunities in technology.
		Student can name at least 3 industry mentors who they can contact for questions, advice or recommendations.
Independence in Problem Solving	Student utilizes a repertoire of resources (search engines, professional message boards, comments and documentation, personal notes) to answer questions, debug programs and solve problems.	

ScriptEd Student Outcomes in Milestones: Foundations Curriculum (Year One)

Student Portfolio – *all students who complete ScriptEd's Foundations curriculum will have a portfolio that demonstrates mastery of foundational computer programming skills, aligned with the ScriptEd Foundations rubric. The portfolio will include the following:*

- **A portfolio webpage** built using HTML and CSS and hosted on GitHub pages. Page includes text, images, links, and is styled using fonts, colors, and arrangement of elements on the page ([student example](#)).
- **4 – 6 interactive web projects** that incorporate HTML, CSS and JavaScript. Each class chooses and selects different projects based on student interest. These projects may include: [Color Changer](#), [Grading App](#), [Around The World](#), [MadLibs](#), [Calculator](#), among others.

Mentorship and Exposure to Technology Careers & Culture -- *all students who complete ScriptEd's Foundations curriculum will have achieved the following:*

- **Exposure to technology career paths and culture** via two or more field trips to major NYC technology companies. During these field trips, students visit companies, learn about its products, services and culture, and engage in question and answer sessions with software developers. Companies visited include Google, Facebook, Tumblr, Soundcloud, and more.
- **Personal relationships with 3-4 mentors in the technology industry** who students can contact for questions or advice. The mentors are the ScriptEd volunteers who teach students throughout the year, which results in meaningful, well-established relationships.
- **Experience using industry-standard technology tools**, including GitHub, Bash, IDEs, text editors, and web based developer tools.

Internship Placements – *All students who complete ScriptEd's summer internship program will have achieved the following:*

- **5 weeks of paid professional experience coding** as a member of a software development team.
- **A completed professional project** that is useful to the internship partner.
- **Professional contacts** who provide students with advice, mentorship and references.
- **4 or more publicly posted blog posts** detailing their internship experiences.
- **A professional resume and profile** detailing their work experience and programming skills.
- **Extensive training and experience in professionalism and self-presentation** in the workplace.