



Vidcode Curriculum and CSTA Alignment

| Unit and Code Focus | Standards |
|---|---|
| <p>1. Intro to Javascript</p> <p>Sequence, functions & arguments, objects & properties</p> | <p>CSTA 1B-AP-09 Create programs that use variables to store and modify data.</p> <p>CSTA 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>CSTA 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>CSTA 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>CSTA 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> |
| <p>2. Arrays</p> <p>Variables, arrays, objects, properties</p> | <p>CSTA 1B-AP-09 Create programs that use variables to store and modify data.</p> <p>CSTA 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>CSTA 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>CSTA 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>CSTA 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>CSTA 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> |
| <p>3. Loops and Animations</p> <p>Loops, randomness</p> | <p>CSTA 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>CSTA 1B-AP-09 Create programs that use variables to store and modify data.</p> <p>CSTA 1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>CSTA 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> |

| | |
|---|--|
| | <p>CSTA 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> |
| | <p>CSTA 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> |
| | <p>CSTA 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> |
| | <p>CSTA 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> |
| | <p>CSTA 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> |
| | <p>CSTA 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> |
| <p>4. Conditional Logic and Special Effects</p> <p>Conditionals, operators</p> | <p>CSTA 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> |
| | <p>CSTA 1B-AP-09 Create programs that use variables to store and modify data.</p> |
| | <p>CSTA 1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> |
| | <p>CSTA 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> |
| | <p>CSTA 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> |
| | <p>CSTA 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.</p> |
| | <p>CSTA 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> |
| | <p>CSTA 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> |
| | <p>CSTA 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> |

| | |
|--|--|
| | <p>CSTA 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> |
| <p>5. Interactivity</p> <p>Event listeners, logical operators</p> | <p>CSTA 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>CSTA 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>CSTA 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>CSTA 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>CSTA 2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>CSTA 2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> |
| <p>6. Algorithms and Art</p> <p>Loops, functions</p> | <p>CSTA 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>CSTA 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>CSTA 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>CSTA 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>CSTA 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>CSTA 2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>CSTA 2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> |
| <p>7. Word Wizardry</p> <p>String manipulation, parameters</p> | <p>CSTA 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.</p> <p>CSTA 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>CSTA 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> |



| | |
|--|--|
| | <p>CSTA 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>CSTA 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>CSTA 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>CSTA 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>CSTA 2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>CSTA 2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> |
| <p>8. Choose Your Own Codeventure!</p> <p>Simulations, combining control structures</p> | <p>CSTA 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.</p> <p>CSTA 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>CSTA 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p> <p>CSTA 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.</p> <p>CSTA 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>CSTA 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.</p> <p>CSTA 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>CSTA 2-AP-17 Systematically test and refine programs using a range of test cases.</p> <p>CSTA 2-AP-19 Document programs in order to make them easier to follow, test, and debug.</p> |
| <p>9. Make it Click!</p> <p>Objects, object constructors, properties, methods</p> | <p>CSTA 2-AP-17 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> <p>CSTA 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.</p> |

| | |
|--|--|
| | <p>CSTA 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.</p> |
| | <p>CSTA 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p> |
| | <p>CSTA 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.</p> |
| <p>10. Make it Move! Objects, arrays, variables</p> | <p>CSTA 2-AP-17 Incorporate existing code, media, and libraries into original programs, and give attribution.</p> |
| | <p>CSTA 3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.</p> |
| | <p>CSTA 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.</p> |
| | <p>CSTA 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.</p> |
| | <p>CSTA 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p> |
| | <p>CSTA 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.</p> |
| <p>11. Make your Mark! Nested loops and conditionals, data structures</p> | <p>CSTA 3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables.</p> |
| | <p>CSTA 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions.</p> |
| | <p>CSTA 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.</p> |
| | <p>CSTA 3A-AP-18 Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.</p> |
| | <p>CSTA 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.</p> |



| | |
|---|---|
| | CSTA 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. |
| 12. Make Up Your Mind! Decision trees, heuristics | CSTA 2-AP-17 Incorporate existing code, media, and libraries into original programs, and give attribution. |
| | CSTA 3A-AP-14 Use lists to simplify solutions, generalizing computational problems instead of repeated use of simple variables. |
| | CSTA 3A-AP-16 Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue by using events to initiate instructions. |
| | CSTA 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects. |
| | CSTA 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible. |
| | CSTA 3A-AP-23 Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs. |