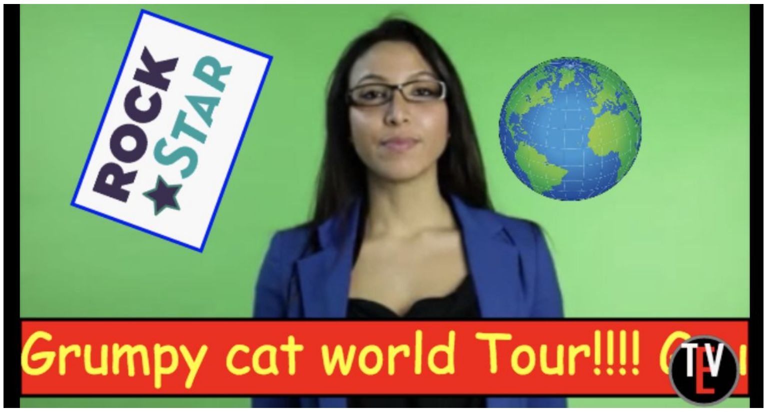
vidcode

Vidcode Lesson Plans
www.vidcode.com
info@vidcode.com



Hour of Code: Code the News

Introductory Lesson Plan

It's the HOC News! Videos and graphics are all about tech, diversity, kids, and coding. Finished projects can be uploaded to school website as the report on the HOC itself. Teachers can encourage students to use the news to report on what they learned, or a statistic for their school. "Breaking News: 400 students at Roosevelt participate in the Hour of Code"

Big Idea: You are a programmer!

Module: https://app.vidcode.io/project/hoc-news

Sample Project: https://app.vidcode.io/share/4PAqHr9pPO

Time: 60-90 minutes

10 minutes background
45-60 minutes coding
10 minutes sharing
5 minutes reflection

Standards

Common Core Standards for Mathematical Practice

- CCSS.MATH.PRACTICE.MP1 Make sense of problems and persevere in solving them.
  CCSS.MATH.PRACTICE.MP2 Reason abstractly and quantitatively.
  CCSS.MATH.PRACTICE.MP3 Construct viable arguments and critique the reasoning of others.
- CCSS.MATH.PRACTICE.MP4 Model with mathematics.
- CCSS.MATH.PRACTICE.MP5 Use appropriate tools strategically.
- CCSS.MATH.PRACTICE.MP6 Attend to precision.
- CCSS.MATH.PRACTICE.MP7 Look for and make use of structure.
  CCSS.MATH.PRACTICE.MP8 Look for and express regularity in repeated reasoning.

Next Generation Science Standards for Engineering Practices

- Practice 1 Defining problems
- Practice 2 Developing and using models
- Practice 3 Planning and carrying out investigations
- Practice 4 Analyzing and interpreting data
- Practice 5 Using mathematics and computational thinking
- Practice 6 Constructing explanations and designing solutions
- Practice 7 Engaging in argument from evidence
- Practice 8 Obtaining, evaluating, and communicating information

CSTA

- CT.L2-01 - Use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, evaluation).
- CT.L2-03 - Define an algorithm as a sequence of instructions that can be processed by a computer.
- CT.L2-04 - Evaluate ways that different algorithms may be used to solve the same problem.
- CT.L2-07 - Represent data in a variety of ways including text, sounds, pictures, and numbers.
- CT.L2-12 - Use abstraction to decompose a problem into sub problems.
- CT.L3A-01 - Use predefined functions and parameters, classes and methods to divide a complex problem into simpler parts.
- CT.L3A-11 -Describe how computation shares features with art and music by translating human intention into an artifact.
- CL.L2-01 - Apply productivity/ multimedia tools and peripherals to group collaboration and support learning throughout the curriculum.

- CL.L2-02 - Collaboratively design, develop, publish, and present products (e.g., videos, podcasts, websites) using technology resources that demonstrate and communicate curriculum concepts.
- CL.L2-03 - Collaborate with peers, experts, and others using collaborative practices such as pair programming, working in project teams, and participating in group active learning activities.
- CL.L2-04 - Exhibit dispositions necessary for collaboration: providing useful feedback, integrating feedback, understanding and accepting multiple perspectives, socialization.
- CL.L3A-01 - Work in a team to design and develop a software artifact.
- CL.L3A-02 - Use collaborative tools to communicate with project team members (e.g., discussion threads, wikis, blogs, version control, etc.).
- CL.L3A-03 - Describe how computing enhances traditional forms and enables new forms of experience, expression, communication, and collaboration.
- CL.L3B-01 - Use project collaboration tools, version control systems, and Integrated Development Environments (IDEs) while working on a collaborative software project.
- CL.L3B-03 - Evaluate programs written by others for readability and usability.
- CPP.L2-07 - Identify interdisciplinary careers that are enhanced by computer science.
- CD.L3A-04 - Compare various forms of input and output.

Background (10 mins)

JavaScript is a programming language. Since computers don’t speak human languages like English or Spanish, we use programming languages to tall to them. JavaScript is the programming language that we can use to talk directly to web pages.

Functions are the main way of getting things done in JavaScript. A function is an action that has a name. Functions are written with parentheses.

Some **functions** need one or more **arguments** inside the parentheses. This tells the function extra information about what to do.

```
text("hello", 5, 10);
```

The text function takes three arguments: text, x position and y position. The order matters!

Teacher Reference: http://www.w3schools.com/js/js_functions.asp

Code Challenge (45-60 mins)

Have students follow the “Hour of Code” tutorial. They should read the instructions on the left and carry them out. When they’ve followed all the instructions on one panel, click the button to move on to the next part.

Circulate throughout the room to troubleshoot and keep students moving through the instructions.

Sharing (10 mins)

Publish all the projects to the public Vidcode Gallery. Look at them all together. Whose stands out? Whose looks like yours?

Stand up and go talk to someone whose filter is similar to yours.Now go talk to someone whose filter is different. Talk about why you chose those filters, and what they say about your personality.

Imagine a world in which everyone made the same choices. Would that world be interesting? Diversity makes the world better

Reflection (5 mins)

- Who’s a programmer? You are!
- Does code look like you expected? Is it easy, hard, straightforward, confusing?
- What is a function? (An action that has a name.)
- What is an argument? (More information about what or how much the function should do.)
- Does the order of the arguments matter? (Yes!)