



Hour of Code: Code the News

Advanced Lesson Plan: Conditionals

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: A conditional checks IF something is true.

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

Conditionals are the final, and most difficult part of this project.

Use the guide below to expand your students' introduction to them in this Hour of Code.

Add a graphic to your video. Use an IF statement to switch the source of the graphic.

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

Time: 60-90 minutes

10 minutes background

1 hour coding

5 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)

Cinderella's stepmother told her, "IF you finish your chores, THEN you can go to the ball." This is an example of an if-statement, or conditional. It means, "You can go to the ball ON THE CONDITION that you finish your chores."

As a class, think of other rules in your life or stories that have IF...THEN. Write them as sentences on the board.

Conditions come up in code all the time. We use them to check IF a certain thing is true, and THEN (and only then) do a particular action.

```
if (x = 1){  
    // do something  
};
```

The code for a conditional may remind you of a loop. "If" is the keyword that starts the conditional. Inside parentheses you write the thing you're checking. It has to be an equation that can be true or false (like a yes/no question). Inside the curly braces you put the code that you want to run if your equation is true. You can write as many lines of code as you want inside the curly braces, and you have to end the whole thing with a semicolon.

We have a whole set of new operators that compare numbers:

- == “Equals Equals” if two numbers are equal
- > “Greater Than” if the first number is greater than the second
- < “Less Than” if the first number is less than the second
- >= “Greater Than or Equal To” if the first number is greater than or equal to the second
- <= “Less Than or Equal To” if the first number is less than or equal to the second
- != “Not Equals” if two numbers are different

Practice using this syntax to rewrite the IF...THEN sentences from earlier in a JavaScript style.

```
if (chores == done){  
    goTo(ball);  
};
```

Code Challenge (60 mins)

Have the students complete the Hour of Code tutorial with a focus on the final two features (switching out graphics) in the “Hour of Code” project.

Sample Solution:

```
repeat(function() {  
    //switch out graphic  
    if(movie.currentTime > 6) {  
        inset.source = "star.png";  
    } else if(movie.currentTime > 3) {  
        inset.source = "meme-1.png";  
    } else {  
        inset.source = "create.png";  
    }  
}, 1);
```

Sharing (5 mins)

Take turns presenting finished projects to the class. Explain how your conditional works. What are you checking? What do you do if that’s true?

Reflection (5 mins)

- What is a conditional? (code that checks if something is true)
- How did we use conditionals today? (to reset the position IF the text went off the edge)
- Why did we put the conditional inside a loop? (to make it keep checking over and over)
- Are you impressed that your code is looking more advanced? (Pat on the back, programmer!)