

 fewd-sample-lesson.md

Raw



Lesson 07 - Introduction To Programming

LEARNING OBJECTIVES

- Practice programmatic thinking by writing pseudo code to solve a basic problem.
- Define web site behavior and the practical uses of JavaScript.
- Predict DOM output / changes by reading JS code.

SCHEDULE

Time	Topic	GA ICLs	Comments
75 min	Introduction to Programming	Thermostat Rock Paper Scissors	
10 min	Introduction To JS	What JS Can Do	A discussion about JS functionality. Based on the JS video students watched for homework .
60 min	Reading JS	Color Switcher	An introduction to JavaScript s syntax
30 min	Lab Time	Traffic Light	

LESSON PLANNING NOTES

Below you will find notes on each section from the proposed schedule above. These notes are meant to help you plan for a great class.

Introduction To Programming

Time: 75 min

Many students know programming as a buzz word, and don't realize programming is about solving problems, and then giving computers steps to solve those problems. This section is meant to clarify what it means to program.

Thermostat

Class Discussion	No solution file
Time	30 min
Topics	pseudo code, conditionals, loops, variables
Description	Pseudo code for an application that would monitor the room temperature and adjust it so the room remains at a certain temperature.
Notes	White board and walk students through writing a program that controls a thermostat. Highlight variables, conditionals, and loops. Again, the purpose of this exercise is to introduce students to what it means to program. We've chosen a thermostat; feel free to change if you like.

Sample Pseudo Code For A Thermostat.

```

get target_temperature
target_temperature = 72
repeat forever,
    current_temperature = get_sensor_reading
    if target_temperature > (current_temperature+5),
        turn_on_heater
    if target_temperature <= current_temperature,
        turn_off_heater

```

Rock Paper Scissors

Exercise	No solution file
Time	30 min
Topics	pseudo code, programmatic thinking
Description	Students write pseudo code on their own to program a computer to play the game rock paper scissors.
Notes	It is best for students to do this in groups of 3 - 4 Encourage students to use the thermostat syntax for clues. They should write each line of instruction onto a post it and start to put the post its in order to form the program. If time permits, ask the students to walk around and view what other groups came up with. Again there is no solution to this exercise. This exercise is to get students thinking programatically.

Introduction To JavaScript**Time: 20 min**

This section is meant to demonstrate the capabilities of Javascript, and get students excited about the next couple of lessons. We often say JS is the behavior of a web site, but students don't actually know what that means. Lets break it down for them.

- Students were asked to watch GA's Front Row video: "[What Can I Do With JavaScript](#) chapter 1."

What JS Can Do!

Class Discussion	No solution file
Time	15 min
Topics	JS, interactivity, server side scripting

Description	Class discussion about JS capabilities.
Notes	Choose a couple of sites that you've built that use JS events (i.e. clicking, scrolling). Have your students visit the site and ask if they can identify what is powered by JS. Ask students What can JS DO? Lead the conversation around some of the basics and not so basic things that are powered by JS (events are basic to understand, node and working with data is advanced). This is a good time to frame what will be covered in JS.

Reading JS

Time: 60 min

We are purposefully omitting jQuery for students introduction to JavaScript. All function names in these stock exercises are meant to be very descriptive. The goal is to have students begin to build their understanding of how code should be read (i.e left to right, top to bottom, and sometimes jumps to functions)

- We want to teach students how to read code. This is based on the assumption that when you are a child you learn to listen and read before you learn to speak and write. We learned to speak JS with discussion, videos, and pseudo code activities. Now lets teach students how to read. Specifically we want students to understand control flow.
- We are not focused on syntax, just the flow of the program. This example uses very descriptive function names so that it is easy to follow. You should mention how to link to a JavaScript file.

Color Switcher

Time: 45 min

Code Along / Walk Through	Color Switcher Solution File
Topics	html, css, js, functions
Description	Walk students through the flow of a JS program.
Notes	These exercises are sometimes easier to showcase in codepen, or to show sublimetext juxtaposed to rendered output in the browser.

Instructional Design Notes

- Allow students to interact with output and start to build an understanding about how JS code interacts with HTML and CSS.
 - For example, in this exercise students can click on a color, see the background change, and then view the code to begin to infer how it all works together.
- The instructor's job during this walk through is to highlight the flow of control in the program without using heavy jargon. Here is an example:
 - JS "listens / observes" for when the user clicks on a list item with the id "x" (.onclick and select element by id).
 - It then calls this block of code. (functions)
 - This block tells JavaScript to, "change the background color to..."
 - Now ask questions to check for understanding (e.g. What would I do if I wanted to change the yellow button to red and change the background to red when its clicked)?
- Let students know that there are many ways to write a program, and we will learn how to make our code more efficient as we go.

Lab Time

Time: 60 min**Traffic Light**

Exercise	Traffic Light In Codepen
Time	30 min
Topics	reading code, functions, .onclick
Description	Students read code and make small changes. The goal is to continue to build on the mental model of how JS works with HTML and CSS.
Notes	The yellow button changes the bulb to purple and the green light does not work. Students are to change the code so that the traffic light works correctly. We are going to ask students to read code and make small changes. Share the Codepen link. Code pen is ideal for this example because it offers immediate feedback. This is essential when students are beginning to build an understanding of how JS code works.

Instructional Design Notes[Link to slides](#)

click icon for slides.

