Intro to Programming

- There are three types of quotes – single, double, and back-tic
  - JavaScript gets confused with quotes inside of quotes, this can be handled:
    - single … double … double … single
    - double … single … single … double
    - single … \single … \single … single (use of escape characters)
    - double … \double … \double … double (use of escape characters)
    - or use string interpolation (template literals)

Exercise 1:

- let myExample = ‘I want to put a “quote” inside of my example’;
- let ex2 = “Let’s say I needed to put a single quote inside my string”;
- let ex3 = ‘Let\’s try that again, except differently’;
- const ex4 = “I really like Ben \& Gerry’s \”Cherry Garcia\” ice cream!”;
- let ex5 = `If you ask me, I’ll tell you the same thing every time - ${ex4} That’s all there is to it!`

Open a Browser (any page) and hit the “ctrl key + shift key + the letter I” all at the same time – this will open Chrome’s developer tools interface – just click on console and you’re ready to go (you could also use codepen).

Type each line into the console – then press enter (copy and paste doesn’t work

Then just type the variable name, and it will display the contents of the variable

Take a screenshot of the result of ex5 and submit it to D2L – there is a submission folder under the “Assessments” menu, then the “Assignment’s” submenu, and the folder is call “Wk3 – worksheet”

For the screenshot in Windows, it best to using the snipping tool and then just save and submit the image

For the screenshot on Mac:

1. Press Shift-Command-4.
2. Drag to select the area of the screen to capture. To move the entire selection, press and hold Space bar while dragging.
3. After you release your mouse or trackpad button, find the screenshot as a .png file on your desktop and submit it to D2L.

Call the file “wk3-exercise1.png or jpg”

- List of Escape characters:

Code	Outputs
\'	single quote
\"	double quote
\&	ampersand
\\	backslash
\n	new line
\r	carriage return
\t	tab
\b	backspace

        - \f   form feed
    - Concatenation & String Interpolation (String Literals)
        - Sting concatenation in JavaScript makes use of the + operator
        - The context the + operator is used tells js whether it’s addition or concatenation
        - We are covering this process here because you will see it in production, be we will be using String Interpolation (String Literals) for the remainder of the class
        - Concatenation Examples of String Building:
            - let myVar = “used to do it”;
              let myStatement = “This is “ + “how we “ + myVar + “.”;
              console.log(myStatement);
            - let myAge = 25;
              let myStatement = “Way back when I was “ + myAge + “, I had a dog!”
              console.log(myStatement);
        - String Interpolation (String Literals) Examples (same code):
            - let myVar = “do it now”;
              let myStatement = `This is how we ${myVar}.`;
              console.log(myStatement);
            - let myAge = 25;
              let myStatement = `Way back when I was ${myAge}, I had a dog!`
              console.log(myStatement);
    - Strings are objects in JavaScript and as such, come with some standard methods that we can use (we will cover functions and methods later – just think of it as some prewritten code that makes your life easier).
        - Examples of String Manipulation and Parsing:
            - let myString = “This is an example, and not a very cleaver one at that”;
            - Since I meant clever (smart), not cleaver (a knife), I need to fix it
              myString = myString.replace(“cleaver”, “clever”);
            - console.log(myString.length); // should show 53
            - console.log(myString.indexOf(“clever”)); // should show 35
            - let newString = myString.indexOf(“clever”); // should still be 35
            - newString = myString.slice(newString, newString + 6);
              I prefer:
              newString = myString.substr(newString, 6);
            - The difference is slice is (Pos, Pos) and substr is (Pos, Length)
            - both produce the same results:
              console.log(newString); // should show “clever”
            - newString = newString.toUpperCase();
            - console.log(newString); // should show “CLEVER”
            - newString = newString.toLowerCase();
            - console.log(newString); // should show “clever”
        - There are more advanced methods that we will discuss later in the course

Note: Again, run a few of these through the console (or codepen) to get a feel for it – but there is nothing to submit. There is an upcoming midterm practice quiz, followed by the midterm – there is a 100% chance that this stuff will be on both.