1. Why was LiveScript renamed JavaScript even though its functionality had nothing to do with Java?
   1. To answer this question, We need to take a look at JavaScript history and how it evolved with all of it’s great Libraries & Frameworks such as React.js, Angular.js, etc.…
   2. The beginning of JavaScript:
      1. In mid-1990s, it was an important time for the internet and the development of browsers e.g. “internet explorer” by Microsoft for instance. In 1995, a Netscape developer developed a new scripting language. It was originally named Mocha, soon after it was renamed LiveScript and Later, it was known as JavaScript.
      2. LiveScript is still being used but not as JavaScript since it eventually compiles to JavaScript
      3. T here is an NPM (Node Package Manager) library called LiveScript (npm install livescript), of course it is not as popular as react-router-dom, but it is an evidence that it is not dead & the last commit in it’s github repository was 2 years ago.
   3. As for the 2nd portion of the question “Even though its functionality had northing to do with Java”, I would agree, JavaScript is a scripting language and has it’s usages however, Java is another animal, it is an OOP (Object Oriented Programming) Language just like how LiveScript is functioning.
   4. JavaScript was renamed that way because the plan in the beginning to evolve the webpages development using Java. That’s why Netscape and Sun (a computer services companies) had a license agreement
   5. What is common about Java & JavaScript, that both can be used to create native Android Apps using Kotlin or React Native. Other than that, I haven’t seen much in common.
   6. A lot could be stated here in this question, but I tried to make it as comprehensive and short as possible.
2. Write a line of code to initialize a variable for a dog named Fido with a weight of 22 lbs.

Shape, rectangle

Description automatically generated with medium confidence

1. Write the HTML syntax for a blank web page with a title of “Chapter 1” for use as a wrapper for JavaScript.

Graphical user interface, text

Description automatically generated

1. Write a while loop that will scoop 3 scoops of ice cream for me. Output the results to the browser.
   1. Check Github repository (<https://github.com/AhmedAbdelRazak/RCC/tree/master/WhileLoop>)

Text

Description automatically generated

1. Write a while loop that will scoop 3 scoops of ice cream for me and will let me know when I have run out of ice cream. Output the results to an alert window.
   1. Check Github Repository (https://github.com/AhmedAbdelRazak/RCC/tree/master/AlertWindow).
2. <html>
3. <head>
4. <link rel="stylesheet" href="styles.css" />
5. <title>While Loop window alert!</title>
6. </head>
7. <body>
8. <script>
9. //Creating an unordered list element to store the array in it
10. var ulElement = document.createElement("ul");
11. document.body.appendChild(ulElement);
12. //Creating Scoop Array To iterate
13. const scoopArray = ["Scoop1", "Scoop2", "Scoop3"];
14. //loop should end after iterating through array scoopArray
15. var loopEnd = scoopArray.length;
16. //initial value
17. var i = 0;
18. while (i <= loopEnd) {
19. let liElement = document.createElement("li");
20. let text = document.createTextNode(
21. scoopArray[i]
22. ? scoopArray[i] + " Delicious Icecream ;)"
23. : "No More Ice Cream",
24. );
25. liElement.appendChild(text);
26. liElement.style.listStyle = "none";
27. liElement.style.marginTop = "10px";
28. liElement.style.animation = `show 600ms ${i}000ms cubic-bezier(0.38, 0.97, 0.56, 0.76) forwards`;
29. ulElement.appendChild(liElement);
30. i++;
31. if (liElement.textContent === "No More Ice Cream") {
32. liElement.style.color = "red";
33. }
34. if (scoopArray[i] === undefined && loopEnd === i) {
35. setTimeout(() => {
36. //launching the alert window after 3 seconds since this is what how I did the animation in CSS ;)
37. window.alert("We don't have ice cream anymore my friend!");
38. }, 3000);
39. }
40. }
41. </script>
42. </body>
43. </html>
44. Explain the fundamental differences between output to the browser, to an alert window, and to the console. When would it be best to use each function? Show me an example in code of each function.
    1. Output to the browser:
       1. Creating html element in javascript
       2. The example below is creating <li> or list tag using javascript



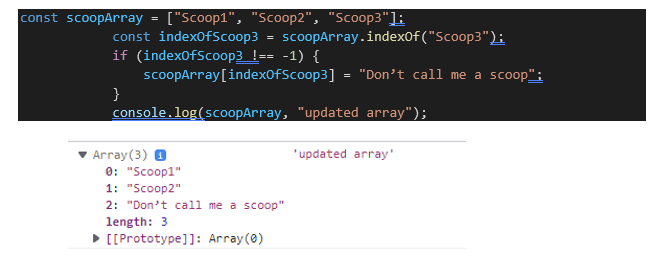
* + 1. Then we can append text to it as demonstrated in the example below



* + 1. The best scenario to use Output to the browser usually when you’re 100% sure that the variables applied are good to go. It is not recommended though to use output in the browser right away.
  1. Output to an alert window:
     1. It is recommended to use the alert window if you’re creating an app that has a delete button or filling a form, if the user tried to delete something or leave the page without completing a form, an alert window should pop.



* 1. Console logging:
     1. This is the most common and effective way to see outputs before adding it to the browser.
     2. it is usually being used to check output in a given function such as a function the renders the sum or another that renders a filtered array. A lot of examples could be written but the above 2 just to mention a few.
     3. The example below is simply a trial to substitute element 3 (index 2) in the array from “Scoop3” to “Don’t call me a scoop”:



1. What is the Document Object Model used for in JavaScript?
   1. Document Object Model also known as the “DOM”. A document object is being created when an html document gets loaded in the browser.
   2. It is simply a tree of multiple methods or attributes. Each branch of the tree ends with an event of action AKA node. DOM methods or properties allow programmers to manipulate with document’s structure, style or content e.g., onClick event listener, so programmers use the DOM to deploy an action when a user clicks on a button. E.g. onClick => submit a form to a database.
   3. Document Object properties:
      1. Document:
         1. Anchor
         2. Form
            1. Text
            2. textArea
            3. Checkbox
            4. Button
   4. Document Object Methods:
      1. getElementById() which returns element or an html tag with a given Id.
      2. getElementByClassName() which returns all elements with the same class
2. What does this code do? setTimeout (wakeUpUser, 4000);
   1. Assuming that wakeUpUser is a function, so the above code will deploy/ launch this function after 4000 ms or 4 seconds.
3. Write a line of code that will calculate the area of a circle. (Area = PI \* Radius Squared)
   1. Please check GitHub repository (<https://github.com/AhmedAbdelRazak/RCC/tree/master/CalculateCircleArea>)

<html>

    <head>

        <link rel="stylesheet" href="styles.css" />

        <title>Circle Area</title>

    </head>

    <body>

        <div

            class="formWrapper"

            style="

                margin-top: 100px;

                text-align: center;

                font-size: 1.5rem;

                font-weight: bold;

            "

        >

            <label> Please fill in the circle radius? </label>

            <br />

            <input

                type="number"

                id="radius"

                placeholder="Circle Radius"

                style="width: 50%; padding: 10px; text-align: center; font-size: 1.2rem"

            />

            <div>

                <button onclick="buttonClicked()">Get Circle Area</button>

            </div>

            <div id="finalOutput" style="display: none; margin-top: 20px">Hi</div>

        </div>

        <script>

            //a function that has a parameter of circleRadius

            function circleAreaCalculation(circleRadius) {

                return circleRadius \* circleRadius \* Math.PI;

            }

            //this is an even listener onclick from the html

            const buttonClicked = () => {

                //storing the radius in a variable from the input field of id radius

                var radius = document.getElementById("radius").value;

                var finalOutputDiv = document.getElementById("finalOutput");

                finalOutputDiv.textContent =

                    "Area of the Circle: " +

                    radius +

                    "\*3.14 = " +

                    circleAreaCalculation(radius).toFixed(2);

                finalOutputDiv.style.display = "block";

            };

            var finalOutput = document.getElementById("finalOutput");

        </script>

    </body>

</html>

1. Explain how JavaScript works and fits into the web page ecosystem.
   1. First the web page is being created using HTML and CSS elements that are already being defined to the Document Object Model AKA the DOM.
   2. After the web page loads the HTML and CSS, it executes the JavaScript code in order (top left, to bottom right)
   3. As for how it fits into the web page, JavaScript is mandatory for any dynamic web page. For instance, if you looked at the program in the previous question, you will find there is a button, This button is useless without using JavaScript since I added an event listener once the button is clicked, the radius should be calculated, not only this, after it is calculated, I want to re-style the element to change it from display:”none” to display:”block”