Dynamic HTML

the DOM, forms & input, and event-handling

Directions

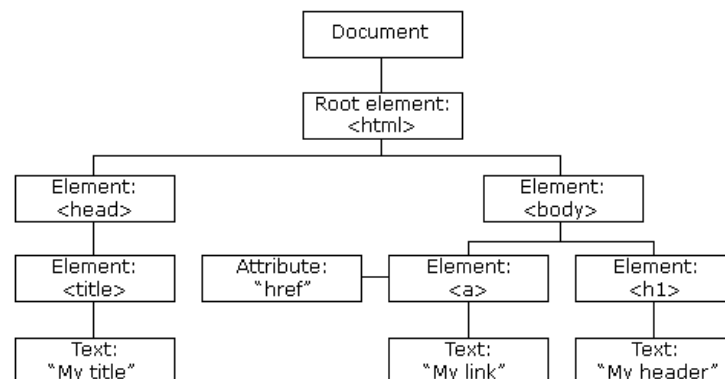- Create a web page to demonstrate dynamic page components.
- Start with this template:

```
<!DOCTYPE html>
<html>
<head>
<style>
  /* put (optional) CSS here */
</style>
<body>
  <!--  put all HTML elements here -->
  <script>
    // put all JS functions here
  </script>
</body>
</html>
```

HTML DOM & forms

Answer in HTML section

1. What is the DOM? What is the document object?
2. Answer using the diagram above (include the diagram on your web page):

      a. What is the parent node of the <body> element?

      b. What are the child nodes of the <body> element?

      c. What is the sibling of the <body> element?

3. Create a div / form with the following elements. *Note: if you use a <form>, read **this**.*

      a. text input

      b. password input

      c. textarea

      d. select menu (>=3 options)

      e. radio buttons (>= 3 options)

      f. checkboxes (>= 3 options)

      g. a submit button.

JavaScript DOM methods

Answer in script section.

1. Demonstrate the following DOM access methods:

      a. document.getElementById

      b. document.getElementsByTagName

      c. document.getElementsByClassName

      d. document.getElementsByName

2. Use the DOM to get:

      a. All the child nodes of the form

      b. The parent node of one input

      c. The previous sibling of one input

      d. The next sibling of one input

3. Use DOM methods and the form above to log or display the following when you click the button:

      a. the text in a text input or textarea

      b. the selected option in a select menu

      c. the checked radio button in a group

      d. ALL checked checkboxes in a group

4. Use the DOM to

      a. get / set the innerHTML of an element.

      b. get / set attributes of an element

      c. get / set CSS properties

5. Use the DOM to create an element and append it to the <body> or a container <div>.

Event handling

Answer in script section

1. Use **onclick** with the event object to get a DOM reference to the element that was clicked. Use the reference to access or modify an attribute of the element.
2. Create another text input and button (or utilize your form above). Assign a keyboard event handler to the text input so that when you press [enter] the button is effectively clicked.
3. Use event handling to mirror a text input in a display div as it's being typed.