socrative

Answer Key
October 24, 2023

JavaScript and HTML

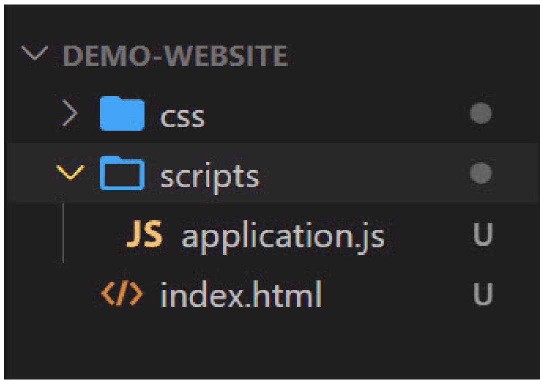
1. Given the folder structure in the image, how would you load *applicaiton.js* into index.html.

A <script source="scripts/application.js"></script>

B <script src="application.js"></script>

C <link href="scripts/application.js"></link>

✓ D <script src="scripts/application.js"></script>



i JavaScript is loaded using the **<script>** tag.

Files can be referenced using a relative path. This means that files that are in the same folder can be referenced directly like this:

<script src="application.js"></script>

But files that are in a different folder need to have the folder need to have the path to the folder included. In this image there is a **scripts** folder in the current folder where the **.js** file is located:

<script src="scripts/application.js"></script>

2. What is the DOM?

A Direct Object Manipulation

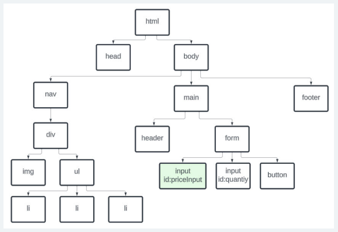
✓ B Document Object Model

C Dynamic Object Module

D Digital Object Manager

E Vin Diesel

i

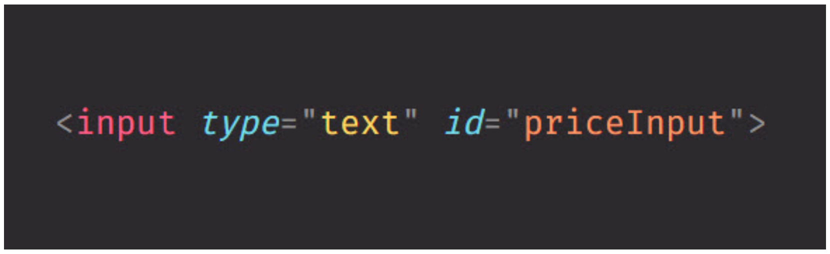


When an HTML page is loaded into a browser, the markup is parsed, interpreted, and the elements are converted into an object tree and loaded in memory.

The page is referred to as a **Document**. The HTML elements are **Objects**. And the hierarchy is the **Model**.

The DOM can be controlled by JavaScript. This means that we can access individual objects in the DOM, we can add objects to the DOM and we can remove objects from the DOM.

3.



```
<input type="text" id="priceInput">
```

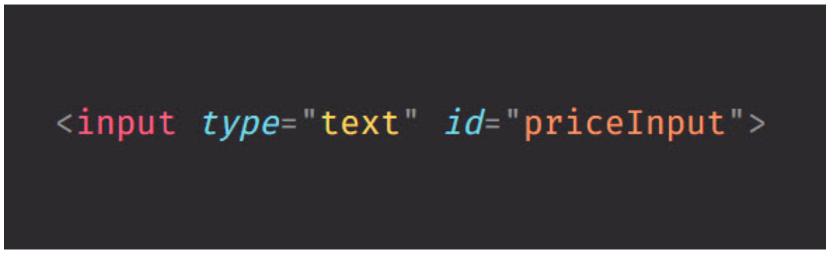
In JavaScript, how would you get access to an input object with an id of **priceInput**?

- **A** const priceInput = document.getElement("priceInput")
- **B** const priceInput = document.find("priceInput")
- ✓ **C** const priceInput = document.getElementById("priceInput")
- **D** const priceInput = getElementById("priceInput")

i To find an element in the HTML the DOM you call the **document.getElementById()** function. Each element should have a unique ID. By declaring a variable you now have access to the object and all the properties of that object.

const priceInput = document.getElementById("priceInput");

4.



```
<input type="text" id="priceInput">
```

In JavaScript, how would you get the value that the user typed into the **priceInput** textbox?

- **A** const price = priceInput.text;
- ✓ **B** const price = priceInput.value;
- **C** const price = priceInput
- **D** const price = priceInput.getValue()

i All input elements have a **value** property that allows you to interact with form elements. You can either get a value that a user entered, or set the value to display to a user.

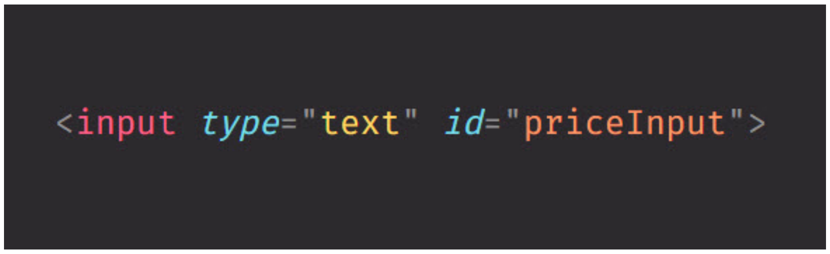
const price = priceInput.value;

Common input elements are:

<input type="text">
<input type="password">
<input type="color">
<input type="checkbox">
<input type="radio">
<input type="email">
<input type="date">
<input type="time">
<input type="number">

See more at w3schools.com/html/html_form_input_types.asp

5.



```html
<input type="text" id="priceInput">
```

If the user entered the value **45** in the text box, what is the data type of the const **price** in the following code snippet.

const price = document.getElementById("priceInput");

A number

B int

C string

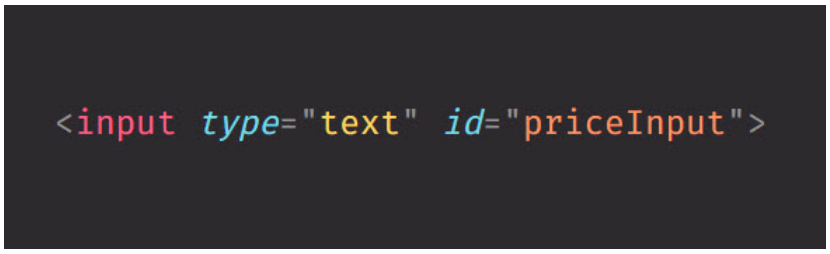
[✓ D] input

In the code snippet the const **price** points to the entire **input** element. This means that we do not just have access to the text that the user typed, but we have access to the entire textbox, including what the user typed.

If we wanted to access what the user typed, we would then have to get the value of the object...

console.log(**price.value**);

6.



```html
<input type="text" id="priceInput">
```

If the user entered the value **15** in the text box, what is the data type of the const **price** in the following code snippet.

const priceInput = document.getElementById("priceInput");
const price = priceInput.value;

A number

B int

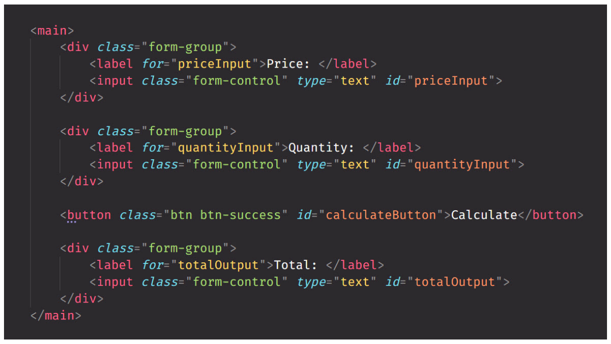
[✓ C] string

D input

The value of an input element is returned as a string. This is because the user input could be any sequence of characters. The string datatype can handle any user input.

In JavaScript you can then parse or convert the string into other datatypes, such as a number.

7.



```html
<main>
    <div class="form-group">
        <label for="priceInput">Price: </label>
        <input class="form-control" type="text" id="priceInput">
    </div>

    <div class="form-group">
        <label for="quantityInput">Quantity: </label>
        <input class="form-control" type="text" id="quantityInput">
    </div>

    <button class="btn btn-success" id="calculateButton">Calculate</button>

    <div class="form-group">
        <label for="totalOutput">Total: </label>
        <input class="form-control" type="text" id="totalOutput">
    </div>
</main>
```

Given the form pictured in the diagram. Which block code should you add to the init function to ensure that the **calculateTotal()** function gets called every time a user clicks the calculate button?

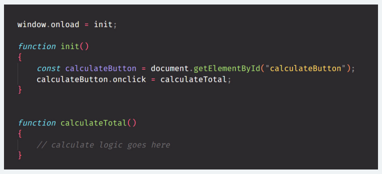
✓ **A** const calculateButton = document.getElementById("calculateButton");
calculateButton.onclick = calculateTotal;

B const calculateButton = document.getElementById("calculateButton");
calculateButton.click= calculateTotal;

C No code is needed, JavaScript will execute the function automatically.

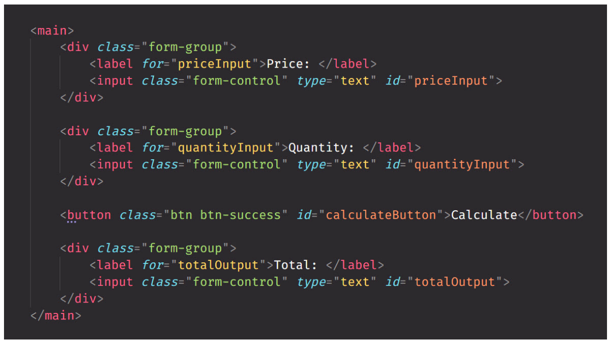
D document.getElementById("calculateButton").value = calculateTotal;

i



```javascript
window.onload = init;

function init()
{
    const calculateButton = document.getElementById("calculateButton");
    calculateButton.onclick = calculateTotal;
}


function calculateTotal()
{
    // calculate logic goes here
}
```

8.



```html
<main>
    <div class="form-group">
        <label for="priceInput">Price: </label>
        <input class="form-control" type="text" id="priceInput">
    </div>

    <div class="form-group">
        <label for="quantityInput">Quantity: </label>
        <input class="form-control" type="text" id="quantityInput">
    </div>

    <button class="btn btn-success" id="calculateButton">Calculate</button>

    <div class="form-group">
        <label for="totalOutput">Total: </label>
        <input class="form-control" type="text" id="totalOutput">
    </div>
</main>
```

Given the form pictured in the diagram.

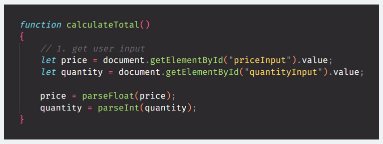
Which of the following code snippets do you need to add to the **calculateTotal** function get the values that user input?

Ensure that all values are numeric.

(Select all that apply)

- ✓ **A** let price = document.getElementById("priceInput").value;
- **B** let quantity= document.getElementById("quantityInput");
- ✓ **C** price = parseFloat(price);
- ✓ **D** let quantity= document.getElementById("quantityInput").value;
- ✓ **E** quantity = parseInt(quantity);
- **F** let price = document.getElementById("priceInput");

i

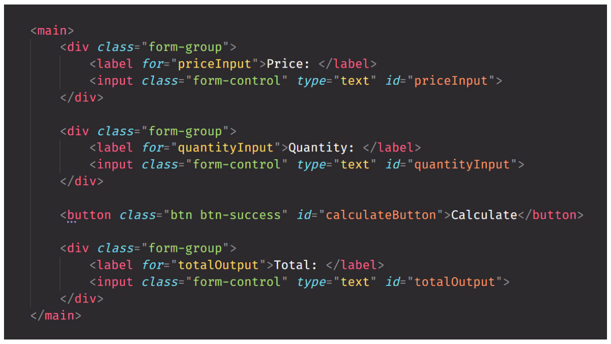


```javascript
function calculateTotal()
{
    // 1. get user input
    let price = document.getElementById("priceInput").value;
    let quantity = document.getElementById("quantityInput").value;

    price = parseFloat(price);
    quantity = parseInt(quantity);
}
```

Step 1 - get a reference to the input elements by using document.getElementById()

Step 2 - convert the user input values into numeric values

9.



```html
<main>
    <div class="form-group">
        <label for="priceInput">Price: </label>
        <input class="form-control" type="text" id="priceInput">
    </div>

    <div class="form-group">
        <label for="quantityInput">Quantity: </label>
        <input class="form-control" type="text" id="quantityInput">
    </div>

    <button class="btn btn-success" id="calculateButton">Calculate</button>

    <div class="form-group">
        <label for="totalOutput">Total: </label>
        <input class="form-control" type="text" id="totalOutput">
    </div>
</main>
```

Given the form pictured in the diagram.

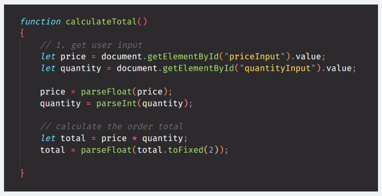
After getting the user input into variables named price and quantity, which code snippets would you need to calculate the order total?

Ensure that the answer is appropriate for financial calculation.

(Select all that apply)

- **A** let total = price + quantity;
- **B** let total = price - quantity;
- ✓ **C** let total = price * quantity;
- ✓ **D** total = parseFloat(total.toFixed(2));
- **E** total = Math.round(total);

i

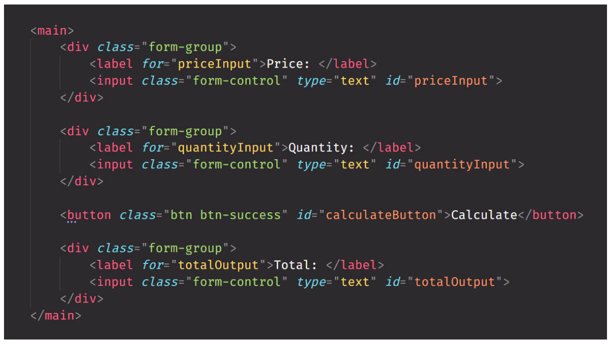


```javascript
function calculateTotal()
{
    // 1. get user input
    let price = document.getElementById("priceInput").value;
    let quantity = document.getElementById("quantityInput").value;

    price = parseFloat(price);
    quantity = parseInt(quantity);

    // calculate the order total
    let total = price * quantity;
    total = parseFloat(total.toFixed(2));

}
```

Step 1 - perform the calculation

Step 2 - round the answer to 2 decimal places.

10.



```
<main>
    <div class="form-group">
        <label for="priceInput">Price: </label>
        <input class="form-control" type="text" id="priceInput">
    </div>

    <div class="form-group">
        <label for="quantityInput">Quantity: </label>
        <input class="form-control" type="text" id="quantityInput">
    </div>

    <button class="btn btn-success" id="calculateButton">Calculate</button>

    <div class="form-group">
        <label for="totalOutput">Total: </label>
        <input class="form-control" type="text" id="totalOutput">
    </div>
</main>
```

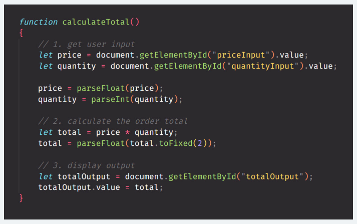
Given the form pictured in the diagram.

After calculating the order total, what code snippets should you add to display the order total to the user?

(Select all that apply)

- ✓ **A** let totalOutput= document.getElementById("totalOutput");
- **B** let totalOutput= document.getElementById("totalOutput").value;
- ✓ **C** totalOutput.value = total;
- **D** total = totalOutput.value;
- **E** totalOutput.text = total;

i



```
function calculateTotal()
{
    // 1. get user input
    let price = document.getElementById("priceInput").value;
    let quantity = document.getElementById("quantityInput").value;

    price = parseFloat(price);
    quantity = parseInt(quantity);

    // 2. calculate the order total
    let total = price * quantity;
    total = parseFloat(total.toFixed(2));

    // 3. display output
    let totalOutput = document.getElementById("totalOutput");
    totalOutput.value = total;
}
```

Step 1 - get a reference to the totalOutput element by using document.getElementById()

Step 2 - set the text (value) of the textbox

11. How can you change the text of non-form elements, like <div>, <span>, <h1>, and <p>?

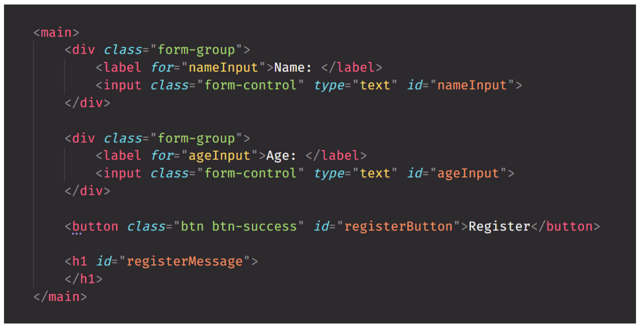
(Select all that apply)

- **A** element.value = "some text";
- ✓ **B** element.innerText= "some text";
- **C** element.text= "some text";
- ✓ **D** element.innerHTML= "some text";

i In JavaScript one way to change the contents of regular HTML elements is by changing the text in the element. Two of the properties that can be used for this are .innerText and .innerHTML.

.innerHTML is considered unsafe because a hacker could exploit it by executing malicious javascript.

12.



```
<main>
    <div class="form-group">
        <label for="nameInput">Name: </label>
        <input class="form-control" type="text" id="nameInput">
    </div>

    <div class="form-group">
        <label for="ageInput">Age: </label>
        <input class="form-control" type="text" id="ageInput">
    </div>

    <button class="btn btn-success" id="registerButton">Register</button>

    <h1 id="registerMessage">
    </h1>
</main>
```

The HTML has been provided for a registration form (shown in diagram).

A user must enter their name and age.

Write the code that you would add to the register() function, and display a message to the **h1** element with the id **registerMessage**.

* If a user is at least 18 years old display the message **"Welcome <name>"**
* If a users is under 18 display the message **"You must be at least 18 years old to register"**

Assume that the following code has already been provided. You just need to provide the code for the **register()** function

----

```
window.onload = init;

function init()
{
    const registerButton = document.getElementById("registerButton");
    registerButton.onclick = register;
}

function register()
{
    // add your code here
}
```