Front End Web Development

9/7/2022

can code communities

ALBANY CAN CODE_

JavaScript Variables

JavaScript variables are containers for storing data values.

In this example, x, y, and z, are variables, declared with the var keyword:

```
var x = 5;

var y = 6;

var z = x + y;
```

Semicolons ;

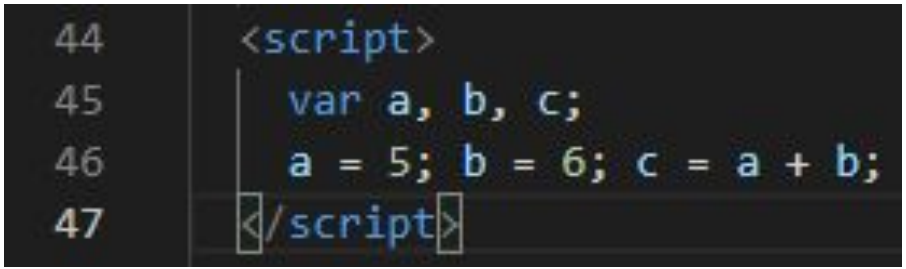
Semicolons separate JavaScript statements.

Add a semicolon at the end of each executable statement:

```
37    <script>
38      var a, b, c;     // Declare 3 variables
39      a = 5;           // Assign the value 5 to a
40      b = 6;           // Assign the value 6 to b
41      c = a + b;       // Assign the sum of a and b to c
42    </script>
```

Semicolons ;

When separated by semicolons, multiple statements on one line are allowed:



Case Sensitivity

All JavaScript identifiers are case sensitive.

The variables lastName and lastname, are two different variables:

```
49  <script>
50      var lastname, lastName;
51      lastName = "Doe";
52      lastname = "Peterson";
53  </script>
```

JavaScript Statements

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

Most JavaScript programs contain many JavaScript statements.

The statements are executed, one by one, in the same order as they are written.

JavaScript programs (and JavaScript statements) are often called JavaScript code.

JavaScript Programs

A computer program is a list of "instructions" to be "executed" by a computer.

In a programming language, these programming instructions are called statements.

A JavaScript program is a list of programming statements.

In HTML, JavaScript programs are executed by the web browser.

JavaScript Functions

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

Why do we use Functions?

You can reuse code: Define the code once, and use it many times.

You can use the same code many times with different arguments, to produce different results.

JavaScript Function Syntax

A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ()

The parentheses may include parameter names separated by commas:

(parameter1, parameter2, ...)

The code to be executed, by the function, is placed inside curly brackets: {}

```
function name(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

Function Invocation

The code inside the function will execute when "something" invokes (calls) the function:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Function Return

When JavaScript reaches a return statement, the function will stop executing.

If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a return value. The return value is "returned" back to the "caller":

```
56  <p id="demo"></p>
57
58  <script>
59    function myFunction(p1, p2) {
60      return p1 * p2;
61    }
62    document.getElementById("demo").innerHTML = myFunction(4, 3);
63  </script>
```

Exercise

- Add 10 paragraphs to your code each with a unique ID
- Write out 10 different Functions to replace your each of the paragraphs in your HTML

HTML Button Tag

- The <button> tag defines a clickable button.

- Inside a <button> element you can put text.

```
11      <h1>The button Element</h1>
12
13      <button>Click Me!</button>
```

Exercise

- Create 5 buttons on your page

Styling buttons

- You can use CSS to style your Button
- Add a class to your button

```
14      <button class="button1">Click Me!</button>
```

```
1   .button1 {
2       border: none;
3       color: rgb(245, 32, 32);
4       text-align: center;
5       font-size: 16px;
6
7   }
```

Exercise

- Add a class to each button
- Style each button differently

HTML Events

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

- An HTML web page has finished loading
- An HTML button was clicked

Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

Adding JavaScript to Buttons

- We will use the onclick attribute to add JavaScript to our buttons
- Onclick is activated when a user clicks on an HTML element. In this case we are going to use buttons

```
22      <button onclick="document.getElementById('demo').innerHTML=Date()">The time is?</button>
23
24      <p id="demo"></p>
```

Exercise

- Add JavaScript to 3 of your buttons using the onclick attribute

Content of Element

- You can also change the content of the element you are working with by using

this.innerHTML

```
19    <button onclick="this.innerHTML=Date()">Click here for the Date</button>
```

Functions in events

- It is more common to see event attributes calling functions

```
34  <p>Click the button to display the date.</p>
35
36  <button onclick="displayDate()">Todays date and time</button>
37
38  <script>
39  function displayDate() {
40    document.getElementById("demo2").innerHTML = Date();
41  }
42  </script>
43
44  <p id="demo2"></p>
```

Exercise

- Create 10 new buttons
- Add onclick attribute that calls a function when the button is clicked
- Have all display the current date and time or some mathematical operation