

HTML5 and CSS3 Complete

Second Edition

Unit M

Programming Web Pages with JavaScript



Objectives

Unit M

- Explore the Document Object Model
- Access elements and properties using JavaScript
- Create statements
- Store and access data in variables

Objectives (continued)

- Create a function
- Add an event listener
- Change CSS with JavaScript
- Create an `if` statement

- JavaScript: the most widely used programming language for modern web browsers
 - Is not Java
 - Created in 1995 by Brendan Eich, who worked at Netscape, and now Mozilla
 - Originally name was Mocha, named after Marc Andreessen, founder of Netscape
 - - 1995 (LiveScript)

Explore the Document Object Model

- Document Object Model (DOM):
standardized way of referring to parts of a web page
 - Creates a hierarchical arrangement known as a DOM tree
 - Each part of HTML document represented by a node
 - W3C standard

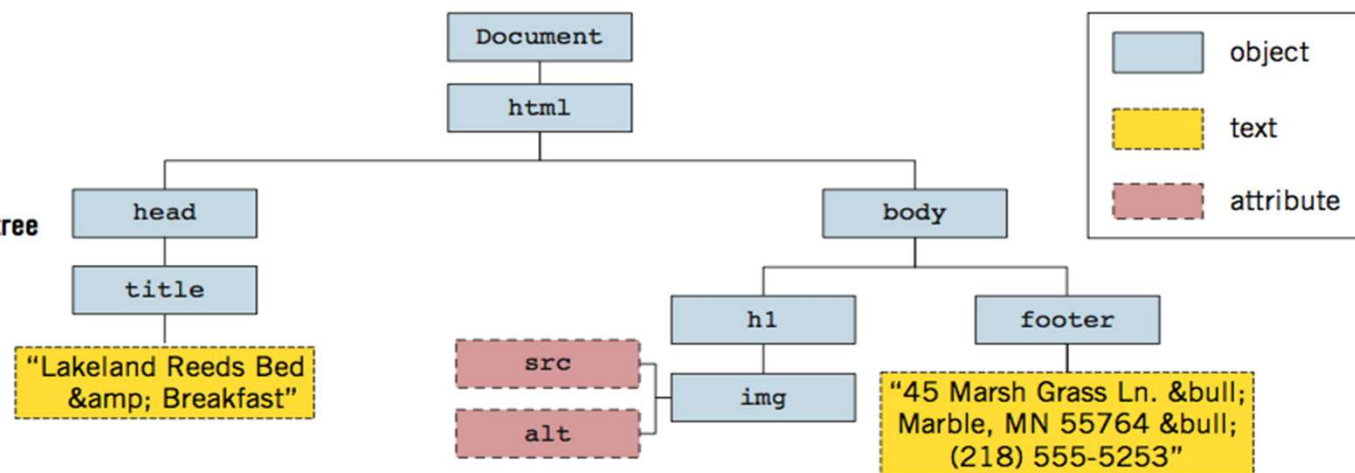
Explore the Document Object Model (continued)

- A DOM Tree

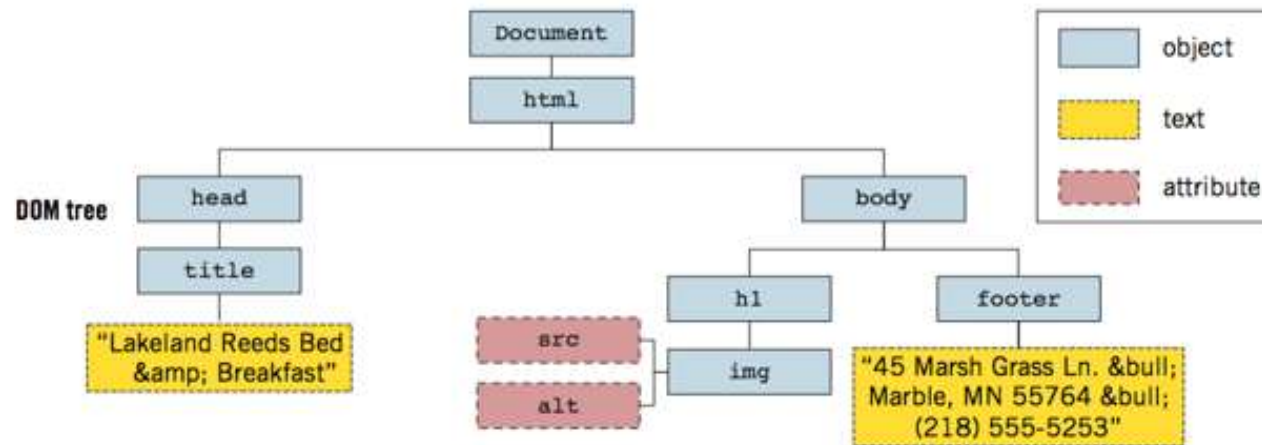
HTML code

```
<html>
  <head>
    <title>
      Lakeland Reeds Bed & Breakfast
    </title>
  </head>
  <body>
    <h1>
      
    </h1>
    <footer>
      45 Marsh Grass Ln. &bull; Marble, MN 55764 &bull;
      (218) 555-5253
    </footer>
  </body>
</html>
```

DOM tree



Explore the Document Object Model (continued)



JavaScript can:

1. change all the HTML elements in the page
2. change all the HTML attributes in the page
3. change all the CSS styles in the page
4. can remove existing HTML elements and attributes
5. can add new HTML elements and attributes
6. can react to all existing HTML events in the page
7. can create new HTML events in the page

Explore the Document Object Model (continued)

- Object: HTML element in DOM
 - Specific object must be identified in order to manipulate it using JavaScript
- Property: piece of a standard set of information associated with DOM node
 - Attributes are considered their own nodes and are associated with their own properties

Explore the Document Object Model (continued)

- Method: action that can be performed for a node
 - Method names are followed by parentheses between which you specify information specific to the method
 - `querySelector()` method lets you access any HTML element by specifying a CSS selector
 - Example:
`querySelector("#nameinput")` selects the element with the `id` value `nameinput`

Access Elements and Properties Using JavaScript

- `querySelector()` method lets you reference objects and properties
 - `querySelector()` is a child of the Document object
- To use a method, specify its parent object, a period, and method name:
`document.querySelector()`

W3C DOM Standard: 3 different parts

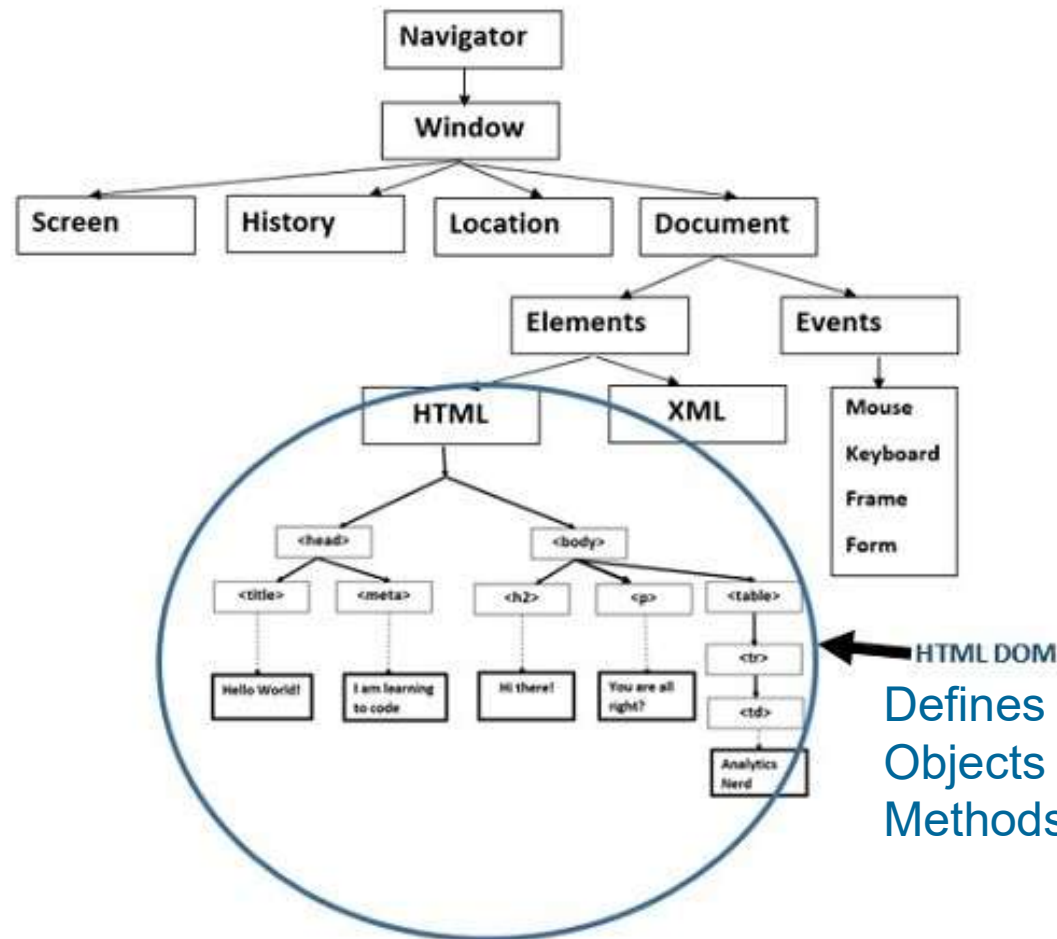
1. Core DOM - standard model for all document types
2. XML DOM - standard model for XML documents
3. HTML DOM - standard model for HTML documents

Core / HTML DOM

Unit M

Define the logical structure of any structure element.

Core DOM



Defines HTML elements as Objects – properties, Methods, events

Example:

`<a id="shark" href=http://www.sharks.com target="_blank", title="attack" class="shark rocks"...`

Access Elements and Properties Using JavaScript (continued)

Unit M

- Specify CSS selector within parentheses of method to reference an object

➤ To select the `aside` element:

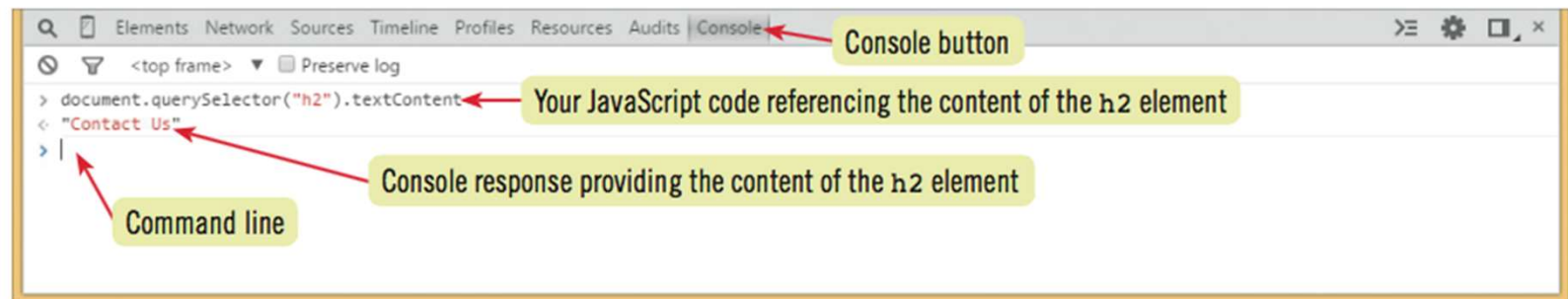
```
document.querySelector("aside")
```

- To access a property, add dot and property name after method:

```
document.querySelector("aside").textContent
```


Access Elements and Properties Using JavaScript (continued)

- Console: part of developer tools in modern browsers; can be used to enter test code and view error messages related to JavaScript
- The browser console in Chrome:





JavaScript Continue

Unit M

PART 2

Create Statements

Unit M

- Statement: a JavaScript instruction that performs an action:
- Assignment operator (=): Part of a statement that lets you assign a new property value
 - Code on left of = accesses property
 - Code on right of = specifies new value
 - often enclosed in quotes
 - Syntax: **Operator**: `x = y`
 - i.e., `x = y` means assign the value of `y` to `x`.

Create Statements (continued)

- Every JavaScript statement ends with a semicolon (;)
 - It's a good practice, but it's not necessary.

- When do you need a semicolon?

`var x = 0; x++` //semi-colon is obligatory

`var x = 0` //semi-colon is optional

`x = ++` //semi-colon is optional

Create Statements (continued)

- Statements created in external JavaScript file
 - Text file with .js extension
 - Referenced within HTML document using `script` element

Create Statements (continued)

- Statements added to file:

```
1  /*
2     Lakeland Reeds Bed & Breakfast style sheet
3     Filename: script.js
4
5     Author:
6     Date:
7     HTML5 and CSS3 Illustrated Unit M, Lessons
8  */
9
10 document.querySelector(".feedback-from").textContent = "Faduma Egal";
11 document.querySelector(".feedback-content").textContent = "Do you allow pets?";
```

A multi-line JavaScript comment starts with `/*` and ends with `*/`, just like a CSS comment

Code to select an element property

Assignment operator

Value to assign to selected property

Every JavaScript statement ends with a semicolon

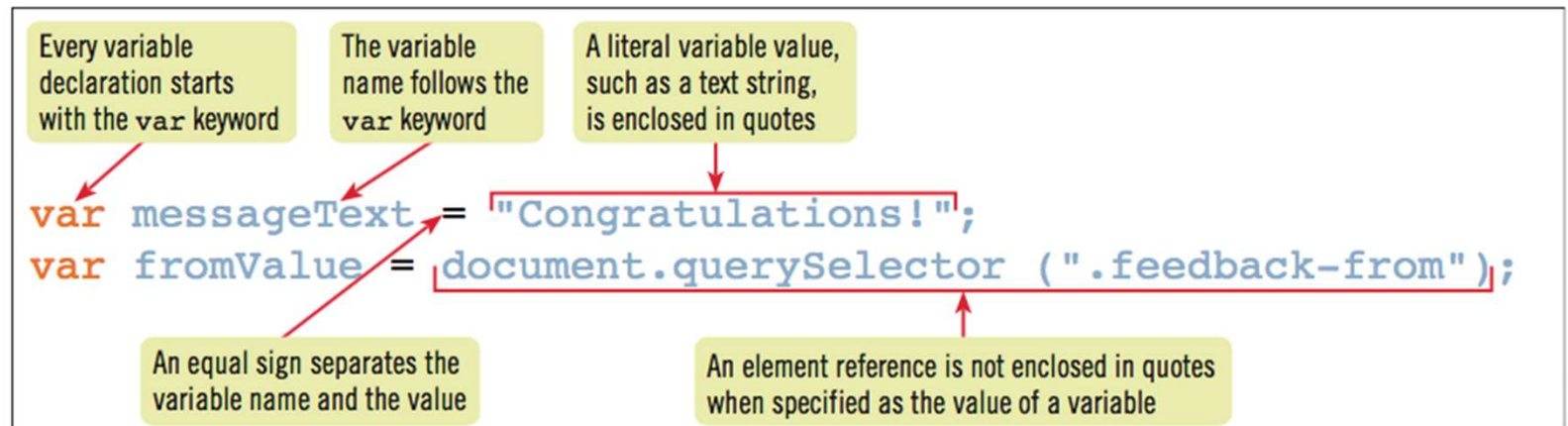
- `script` element in HTML document:

```
93     </footer>
94     <script src="script.js"></script>
95     </body>
96 </html>
```

The `src` attribute specifies the name of the JavaScript file; `script` elements are often placed at the end of the body section to allow the page content to download before the script

Store and Access Data in Variables

- Variables: Stored values you can access with a name you specify
 - Can store many types of information
 - Create with `var` keyword followed by name, equal sign, and value:



Store and Access Data in Variables (continued)

- Shorter statements easier to work with
 - Common to store object references as variables, then reference in other statements using variable names:

```
1  /*
2    Lakeland Reeds Bed & Breakfast style sheet
3    Filename: script.js
4
5
6
7    Illustrated Unit M, Lessons
8
9
10 /* create variables */
11 var fromValue = document.querySelector(".feedback-from");
12 var feedbackValue = document.querySelector(".feedback-content");
13
14 fromValue.textContent = "Faduma Egal";
15 feedbackValue.textContent = "Do you allow pets?";
```

You use the variable name to reference the variable's value in other statements

Illustrated Unit M, Lessons

The variable value is a reference to an object in the HTML document

The variable names take the place of the object references

DEMO: document.method();

Unit M

SYNTAX: document.querySelector(selectors);

- Returns the first Element within the document that matches the Specified
 - Selector
 - group of selectors
 - or null if no matches are found.

Contact Us

All fields are required

Name

Mar Castro

Email

mcastro@itu.edu

Feedback or questions

Test

Using F12 Console

```
> document.querySelector("article");
```

```
< ▲ <article id="contentstart">  
    <h2>Contact Us</h2>  
    > <form class="contact-form" action="results.html">...</form>  
    > <div class="feedback-preview show">...</div>  
</article>
```

> The string argument pass to querySelector must follow the CSS syntax.

DEMO: document.method();

Unit M

SYNTAX: document.getElementById()

- Returns reference by ID

You can manipulate the contents of your HTML tag.

Contact Us

All fields are required

Name

Mar Castro

Email

mcastro@itu.edu

Feedback or questions

Test

Using F12 Console

```
All  Errors  Warnings  Info  Logs  [X] Preserve Log
[Info] HTML1300: Navigation occurred.
> document.getElementById("contactme").innerHTML = "thanks";

> var e = document.getElementById("contactme");

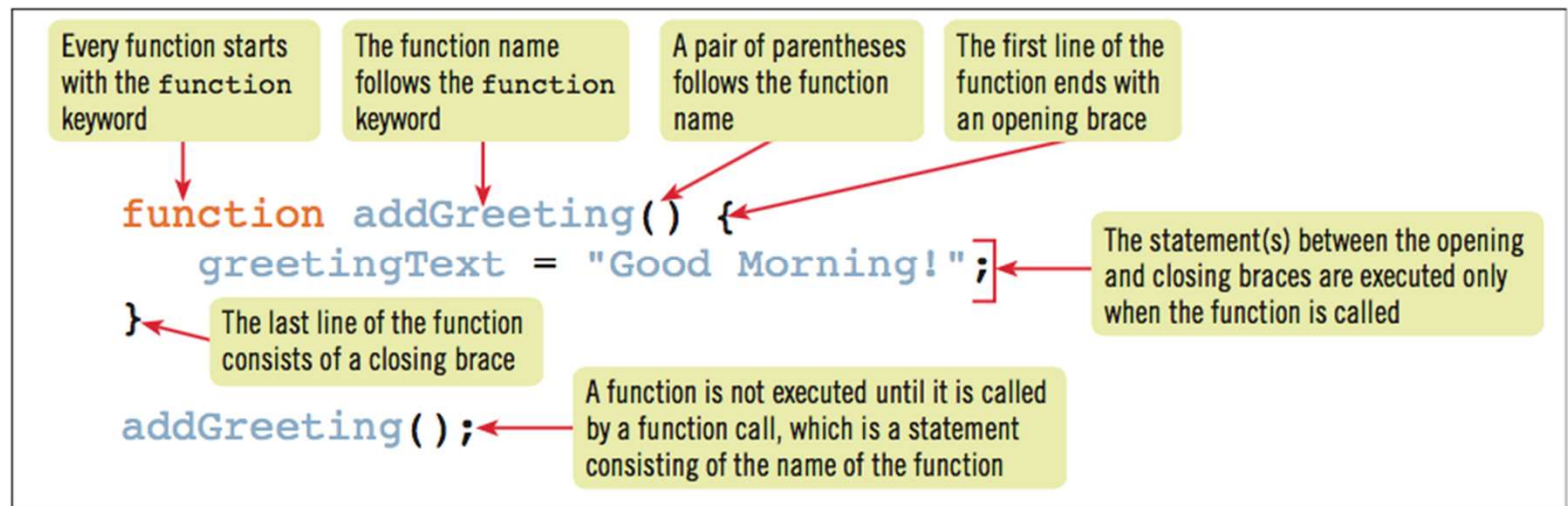
< undefined
> e.style.color="red";
< "red"
>
```


Create a Function

- Function: Group of one or more statements with an assigned name
 - Statements in function referenced as a single unit
 - Create with function keyword, followed by name of function and ()
 - Statements enclosed in a single pair of braces { }

Create a Function (continued)

- Function call: reference to a function name elsewhere in code to indicate when function should be executed
- Creating and calling a function:





Function Demo

Unit M

```
function completePreview() {  
  
    /**access the text content and replace with nameField*/  
    fromValue.textContent = nameField.value;  
  
    /**access the text content and replace with feedbackField*/  
    feedbackValue.textContent = feedbackField.value;  
  
    feedbackPreview.className = "feedback-preview show";  
  
    /**Returns true if an input element contains valid data*/  
    if (form.checkValidity() === true) {  
        /**get class property submitbutton and show*/  
        submitButton.className = "submitbutton show";  
    }  
  
    /**Returns true if an input element contains valid data*/  
    if (form.checkValidity() === false) {  
        /**get class property*/  
        /**SYNTAX returns class name property: HTMLElementObject.className */  
        /**SYNTAX set the class name property: HTMLElementObject.className = class  
        class specifies the class name of an element  
        */  
  
        /**set class property submitbutton -- set to hide "none" */  
        submitButton.className = "submitbutton";  
    }  
}
```

Contact Us

All fields are required

Name

Email

Feedback or questions

Add an Event Listener

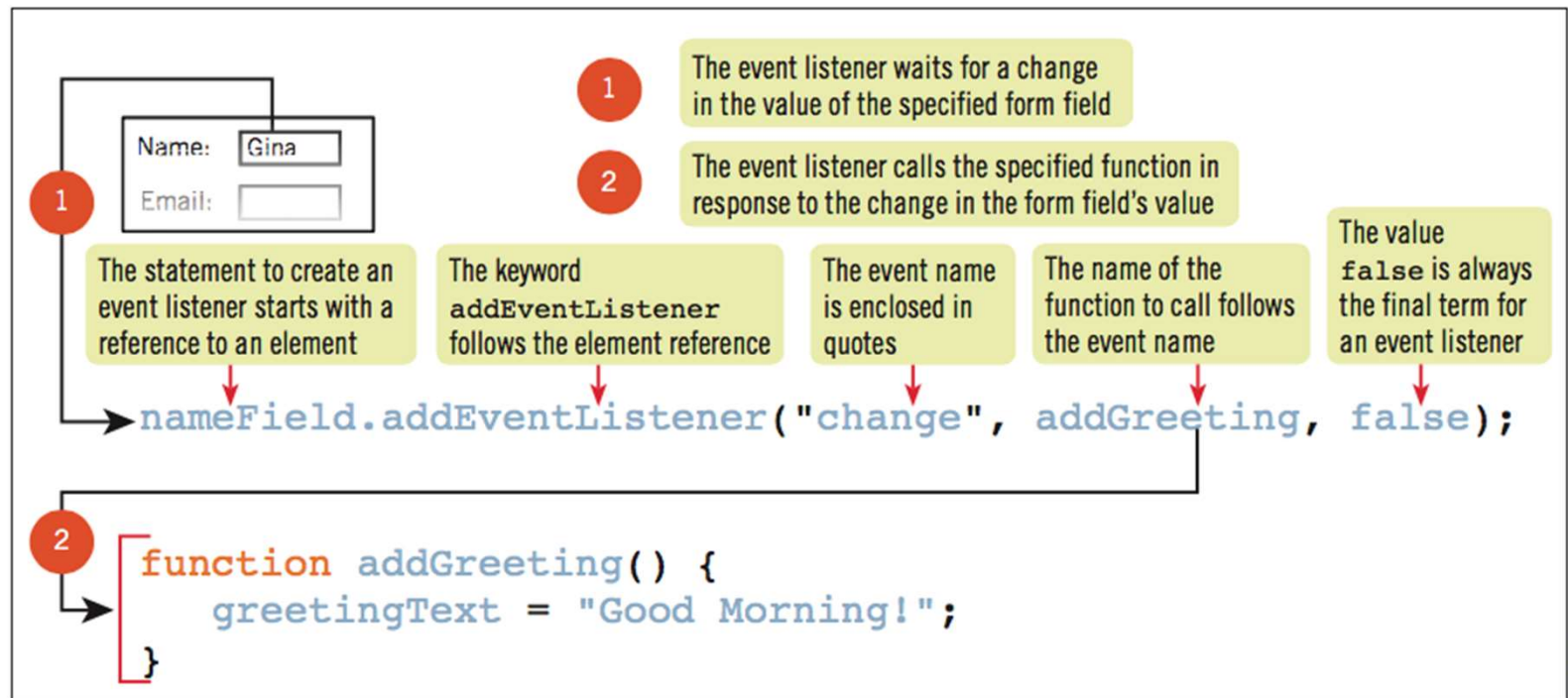
Unit M

- Events: actions commonly performed on a web page
 - Can write JavaScript that responds to events
- Commonly used events:

event	description	event	description
change	the value of an element changes	mouseover	a user moves the mouse pointer over an element or touches an element
click	a user clicks or touches an element	submit	a user submits a form
load	a document or image loads	touchend	a user removes a finger or stylus from the screen
mouseout	a user moves the mouse pointer off an element or stops touching an element	touchstart	a user touches a finger or stylus to the screen

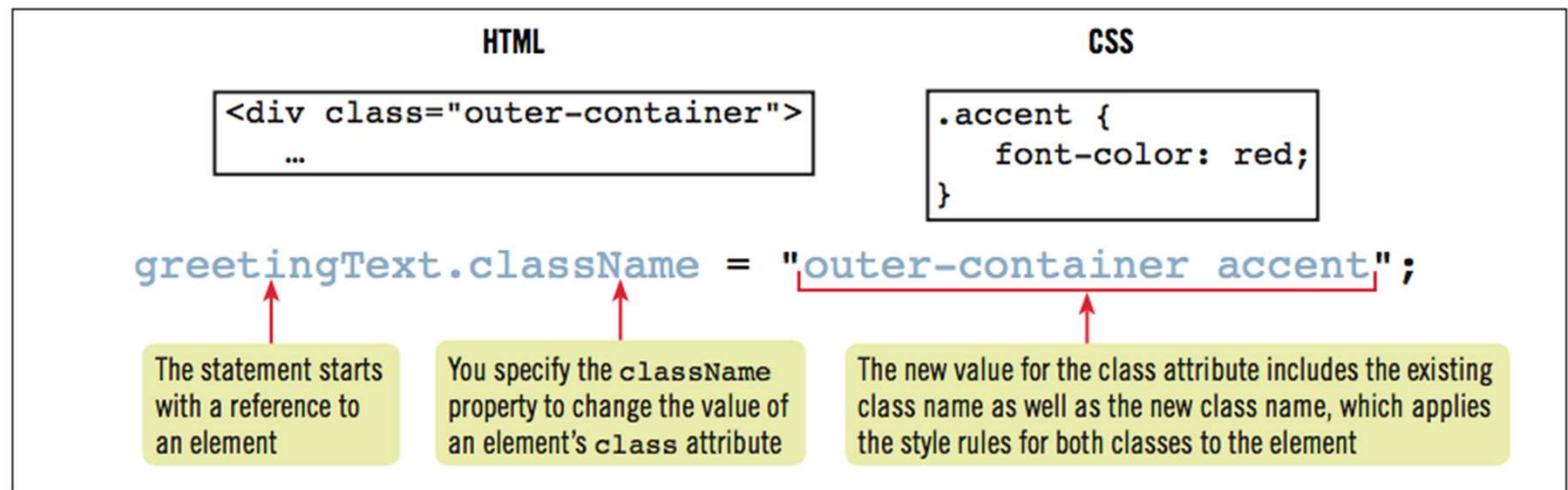
Add an Event Listener (continued)

- Event listener: a statement that specifies an object, an event, and function to call in response to event



Change CSS with JavaScript Unit M

- Can use JavaScript to change element's CSS in response to event
 - Create style rule using class selector, then use JavaScript to add/remove class values from element based on events:



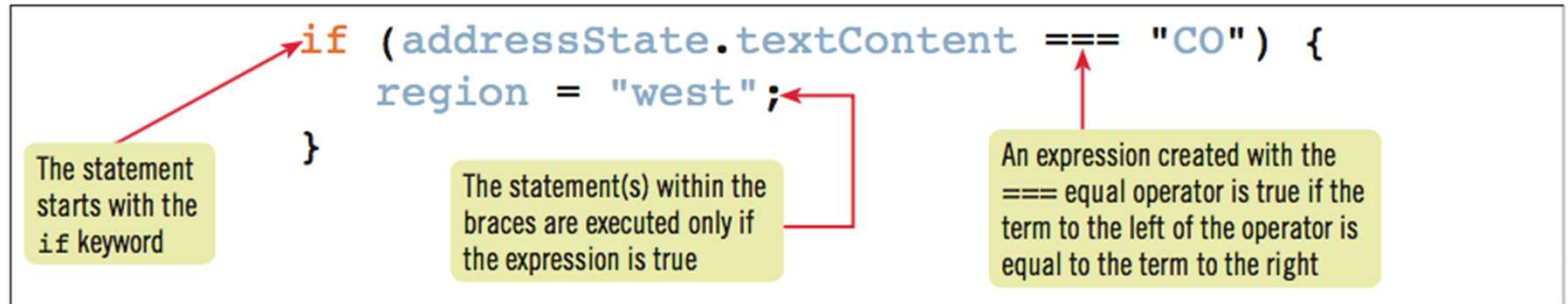
Create an `if` Statement

Unit M

- operators: symbols to compare or change values of multiple objects or properties
 - assignment operator (=)
 - comparison operators: determine whether 2 values same or different
- if statement: compares 2 values; if result is true, statements are executed

Create an `if` Statement (continued)

- Syntax for creating an `if` statement:



- Operators:

operator name	operator	description	example
assignment	=	assigns the value on the right to the object on the left	<code>var name = "Faduma"</code>
equal	===	true if the value on the left is equal to the value on the right	<code>if (a === b)</code>
not equal	!==	true if the value on the left is not equal to the value on the right	<code>if (a !== b)</code>

Summary

- DOM is a standardized way of referring to parts of a web page
- The `querySelector()` method lets you reference objects and properties
- Script code is created by combining DOM objects, properties, and methods
- A statement is a JavaScript instruction that performs an action

Summary (continued)

- The assignment operator lets you assign a new property value
- Variables are stored values you can access with a name you specify
- A function is a group of one or more statements with an assigned name
- A function must be called for its statements to be executed

Summary (continued)

- An event listener is a statement that specifies an object, an event, and function to call in response to event
- You can use JavaScript to change the CSS for an element in response to an event
- An `if` statement compares 2 values and executes statements only if the comparison result is true