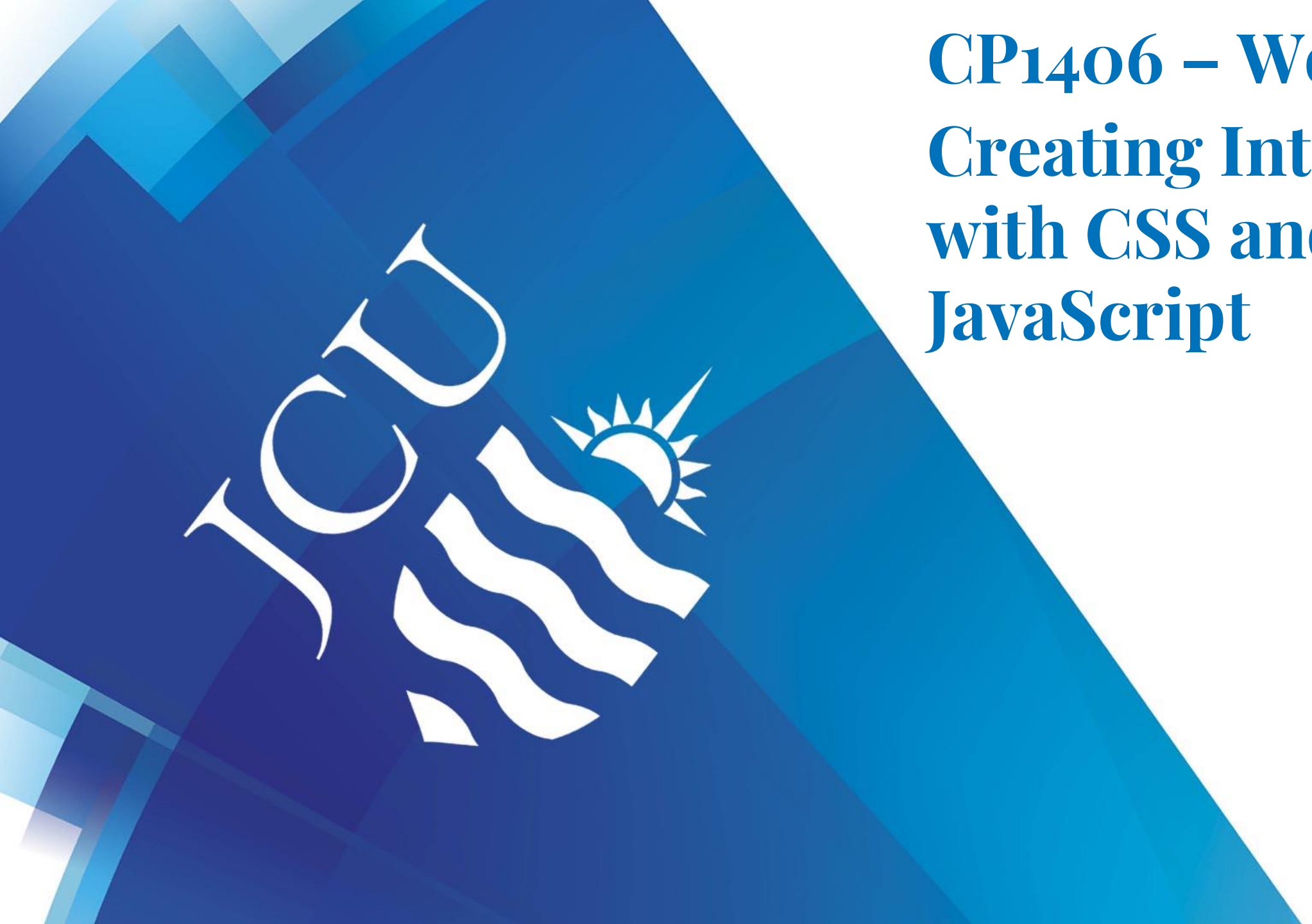


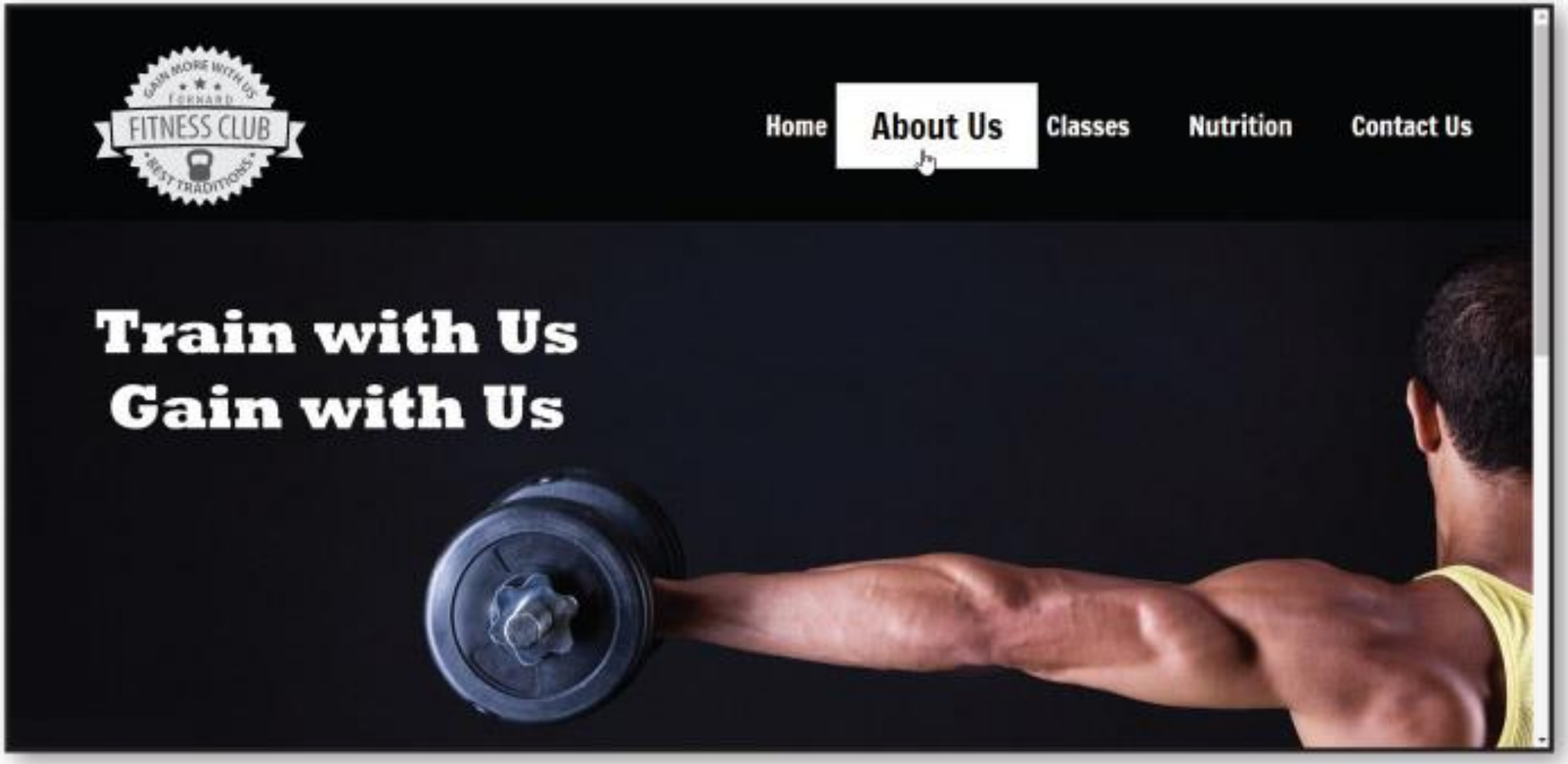
CP1406 – Week 8
Creating Interactivity with CSS and JavaScript

JAMES COOK UNIVERSITY
AUSTRALIA

Introduction

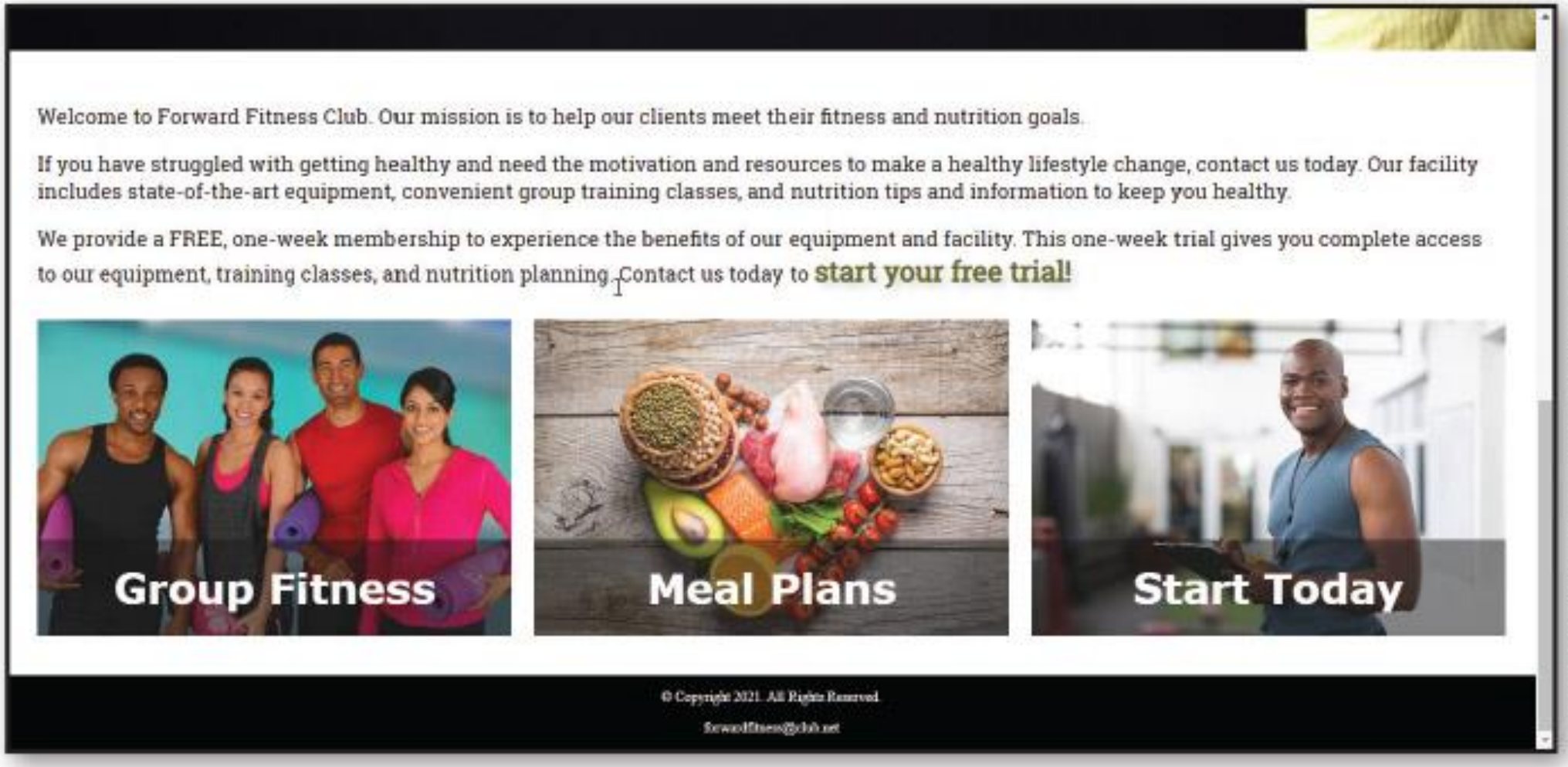
- Most modern websites include some form of interactivity
  - Commonly integrated within a website using CSS and JavaScript, a web programming language used to enhance a website and create interactivity
    - An effective way to create interesting and useful webpages is to include dynamic content to make the webpage interactive

Introduction – CSS Transform Property



Home Page with CSS Transform Property

Introduction - Animated Captions



Y Photo Studio/Shutterstock.com, Rob Marmion/ Shutterstock.com, Oleksandra Naumenko/ Shutterstock.com, michaeljung/Shutterstock.com

Home Page with Animated Captions

Introduction - Hamburger Menu Icon

nexusby/Shutterstock.com, Rob Marmion/Shutterstock.com

Home Page with Hamburger Menu Icon

Introduction - JavaScript

The following are common exercises that we encourage our clients to do as part of their daily exercise routine.

Burpee

Burpees are a great, full body exercise to increase your strength and endurance. Begin in a standing position, drop into a squat and extend your hands forward, kick your feet back and then forward again quickly, and then jump up from a squatted position. **View Example.**

Plank

Planks build your core strength. To perform a plank, get in a push up position and rest your forearms on the floor. Hold the position as long as you can.

Mountain Climber

Mountain climbers are a good cardio exercise. Place your hands on the floor in a push up position, then bring one knee up to your chest and then switch as quickly as you can (as though you are climbing a mountain).

Murphymedia/Shutterstock.com

JavaScript Displaying Video

Using CSS to Create Interactivity

- CSS began as a simple way to add colour and design a webpage
  - Today, it has evolved to become so much more
    - Can be used to move elements on a webpage, change the colour of an element, or change the appearance of an element

Using CSS to Create Interactivity

- CSS transform property allows you to rotate, scale, skew, or translate a block element

Table 10–1 Transform Property Values

Method	Description	Example
matrix( )	A 2D transformation; accepts six values	transform: matrix(1, 0.5, -0.5, 1, 10, 0)
rotate( )	A 2D rotation; rotates an element a specified number of degrees clockwise or counter-clockwise	transform: rotate(10deg)
rotateX( )	A 3D rotation; rotates an element a specified number of degrees on the elements X-axis	transform: rotateX(40deg)
rotateY( )	A 3D rotation; rotates an element a specified number of degrees on the elements Y-axis	transform: rotateY(30deg)
rotateZ( )	A 3D rotation; rotates an element a specified number of degrees on the elements Z-axis	transform: rotateX(20deg)
scale( )	A 2D scale transformation; resizes an element	transform: scale(1.5)
scaleX( )	A 2D scale transformation; resizes an element on its X-axis	transform: scaleX(1.5)
scaleY( )	A 2D scale transformation; resizes an element on its Y-axis	transform: scaleY(1.5)
skew( )	A 2D skew transformation; moves the top and bottom or left and right sides a specified number of degrees	transform: skew(10deg, 10deg)
skewX( )	A 2D skew transformation for the X-axis of an element	transform: skewX(20deg)
skewY( )	A 2D skew transformation for the Y-axis of an element	transform: skewY(30deg)
translate( )	A 2D translation; moves the block element from its original position on the webpage	transform: translate(30px, 40px)
translateX( )	A 2D translation; moves the block element from its original position on the webpage	transform: translateX(30px)
translateY( )	A 2D translation; moves the block element from its original position on the webpage	transform: translateY(40px)

Using CSS to Create Interactivity

https://emporiumpies.com/pies

Using CSS to Create Interactivity

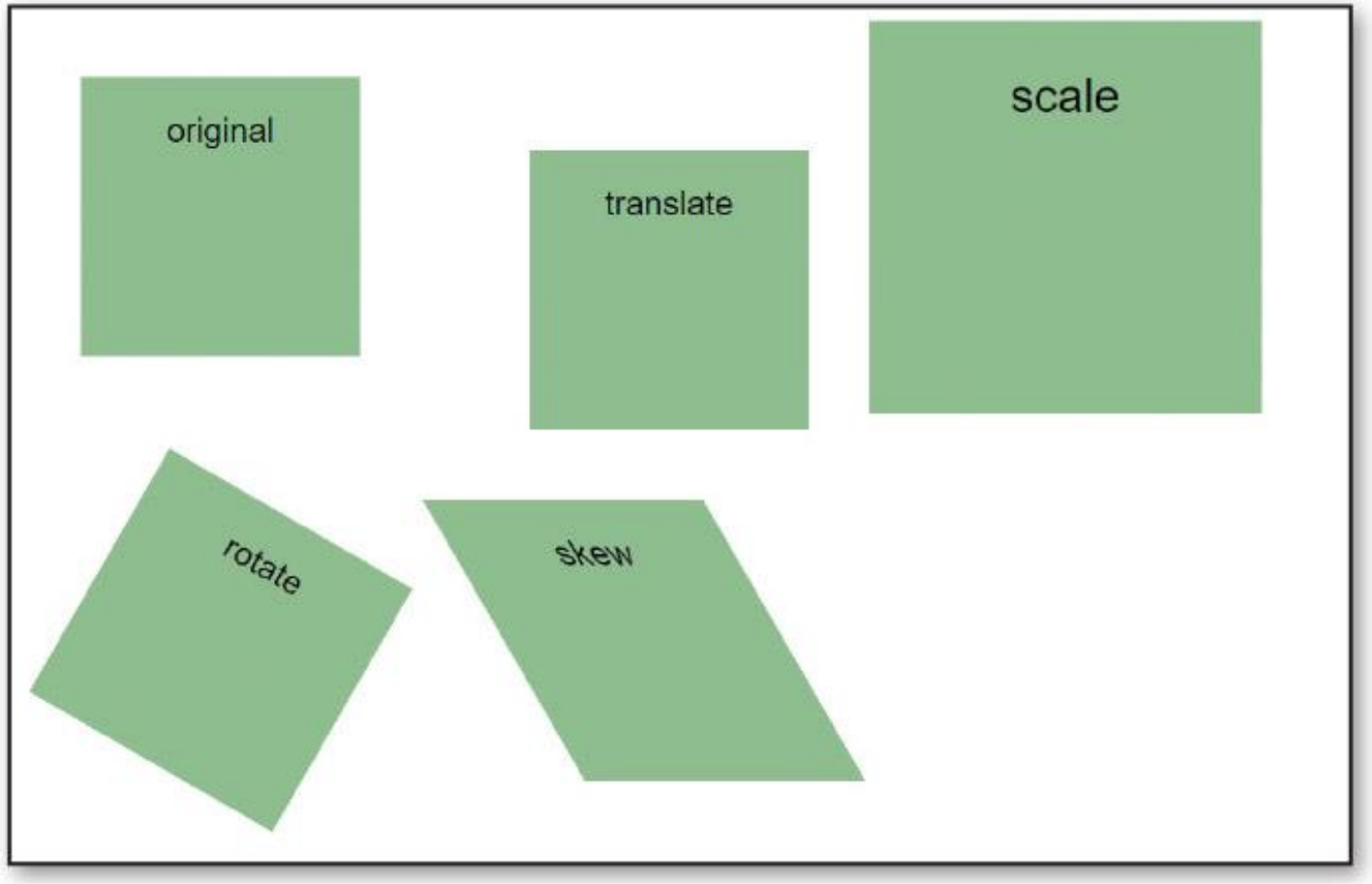
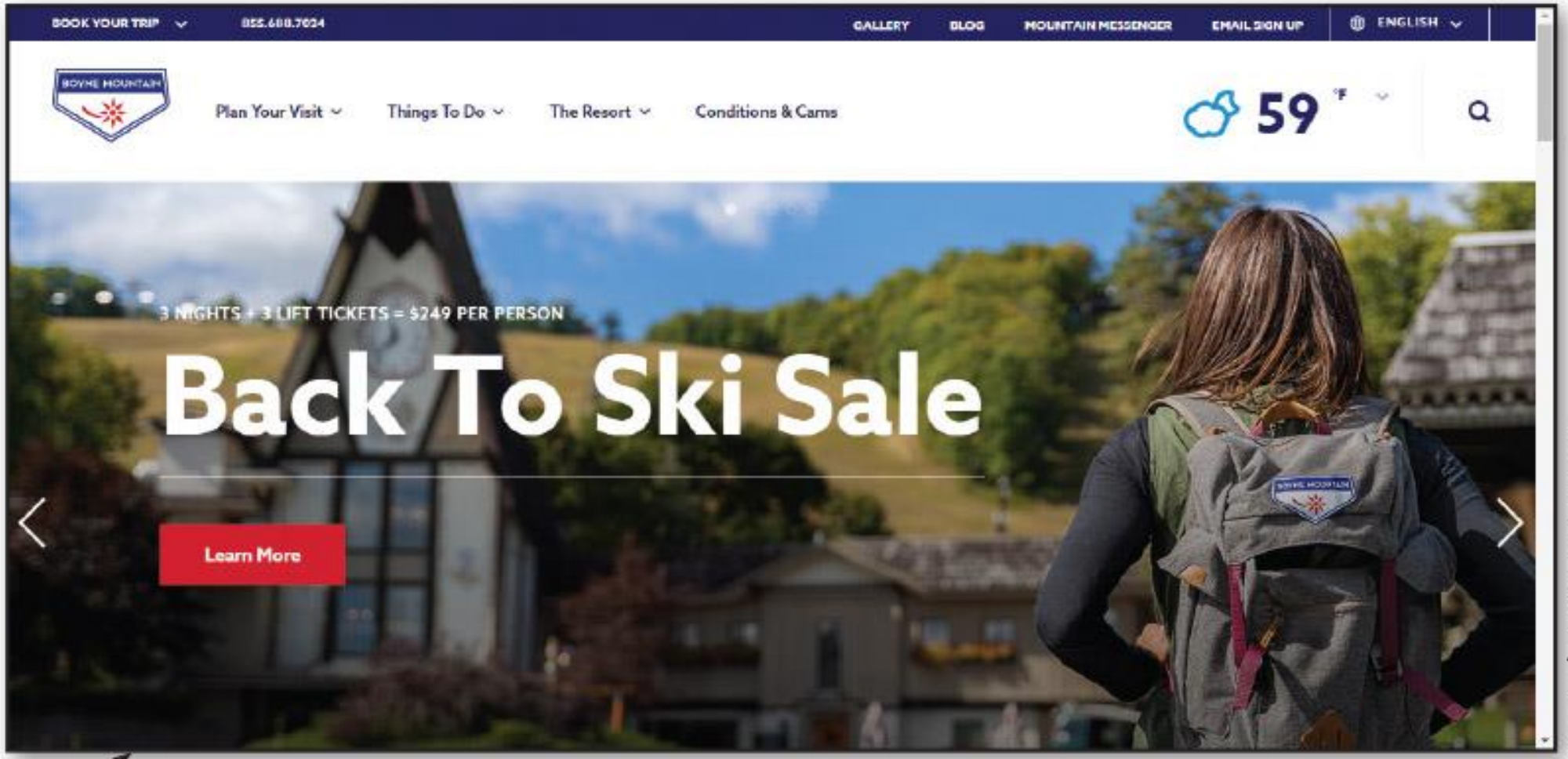


Figure above demonstrates, the **translate** method moves the element from its original position. The **scale** method increases the size of the original element. The **rotate** method rotates the element 30 degrees clockwise. The **skew** method skews the sides of the element.

Incorporating JavaScript

- JavaScript
  - Scripting language that provides various types of functionality to webpages, such as the ability to interact with the user
  - Web developers use it to control webpages
  - Client-side scripting language: browser interprets and renders the JavaScript

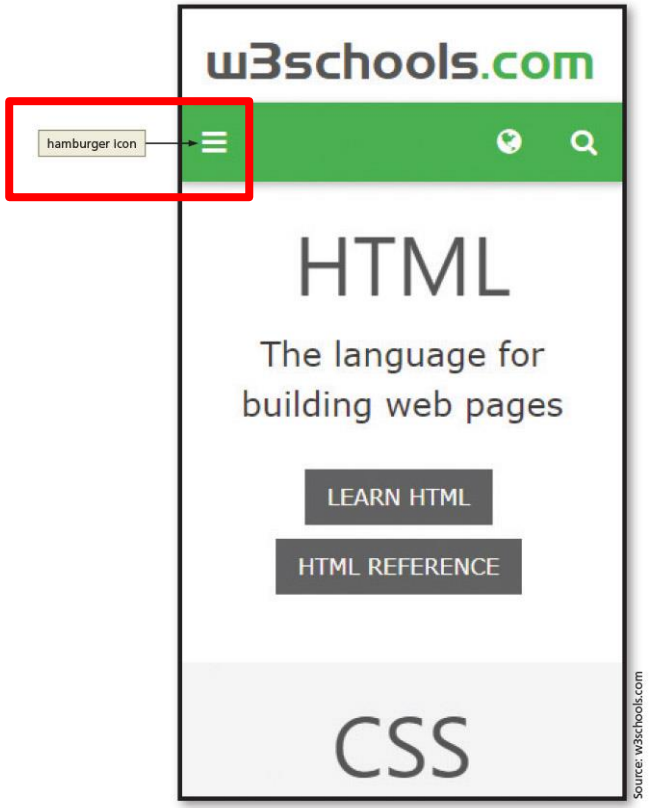
Incorporating JavaScript



JavaScript slide show

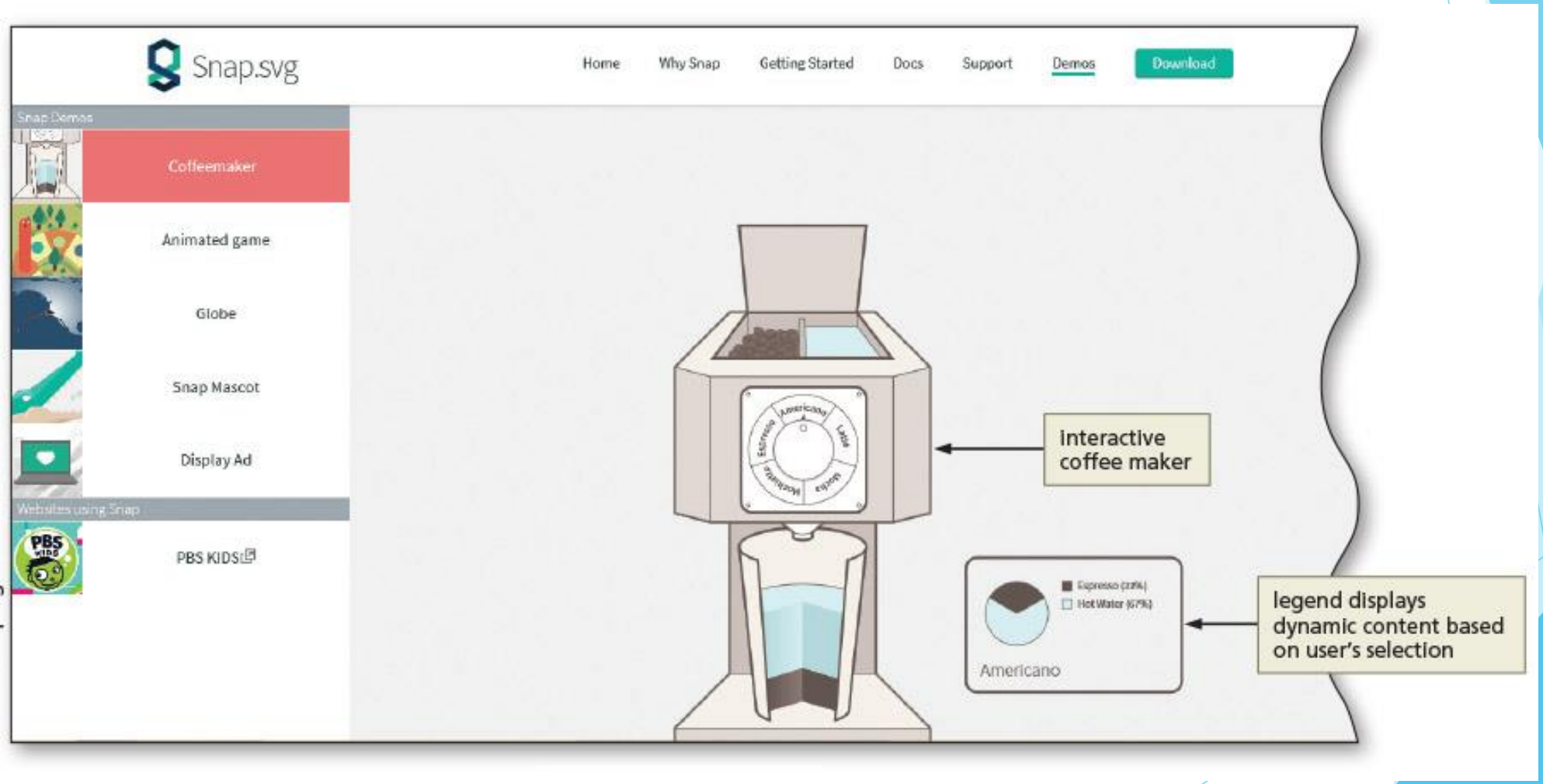
Incorporating JavaScript

- Many mobile websites integrate an icon commonly called the hamburger icon for use as a menu button
  - Consists of three, horizontal, parallel lines and uses JavaScript to display a menu and allow users to select an option



Incorporating JavaScript

http://snapsvg.io/demos/



CENGAGE

JavaScript Terminology

- To use JavaScript effectively, you should be familiar with its basic terminology
  - Statement: line of programming instructions to be executed by the client (the browser)
  - Object: programming code and data that can be treated as its own entity
  - Properties: attributes that describe an object's characteristics
  - Methods: actions that an object can perform

JavaScript Terminology

- To use JavaScript effectively, you should be familiar with its basic terminology (Cont.)
  - Argument: value given to a method
  - Function: set of JavaScript statements that perform a specific task
  - Variable: container that holds a value
  - Event handler: used by JavaScript to associate an action with a function

Writing JavaScript Code

- Syntax rules and guidelines should be followed when writing JavaScript code
  - JavaScript is case sensitive
  - One-line comment and multiline comments are written as follows:
    - // Single line comment syntax
    - /* Multiple line
      comment syntax */
  - Semicolons are used to end JavaScript statements

Writing JavaScript Code

- Syntax rules and guidelines should be followed when writing JavaScript code (cont.)
  - JavaScript can be written within an HTML page or as a separate JavaScript file with the filename extension .js
  - When written within an HTML page, the code may be within the head element or the body element
  - When created as an external .js file, a script element is placed in the head element of the HTML file to specify the external .js file as the file source, as follows:
    - <script src="scripts/myfunction.js"></script>

How to include JavaScript in websites

1. Embedded in HTML directly, e.g.

```
<a id="button" href="#"
onclick="alert('Hello World')">Link</a>
```

2. Function calls in HTML + scripts in same file or linked file

```
<a id="button" href="#" onclick="fn()">Link
</a>
```

3. Unobtrusive - HTML doesn't know anything about JS except to link to it

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

Unobtrusive

- **HTML:**

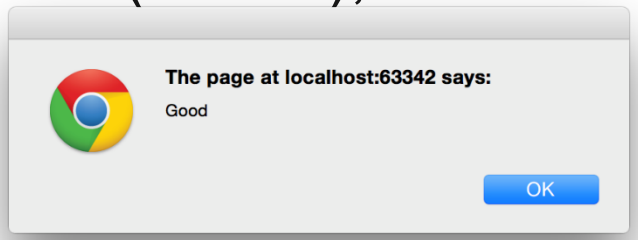
```
<script src="js/scripts.js"></script>
…
<a id="button" href="#">Link</a>
```

- **JavaScript file (scripts.js):**

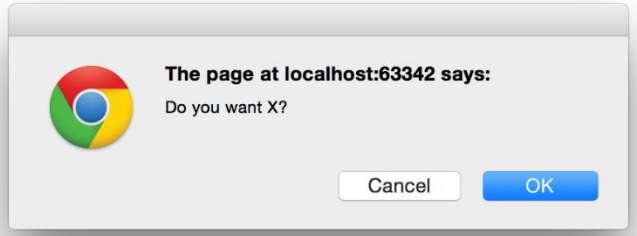
```
button = document.getElementById('button');
button.addEventListener("click", function() {
    console.log("Hello World!");
});
```

Useful commands & functions

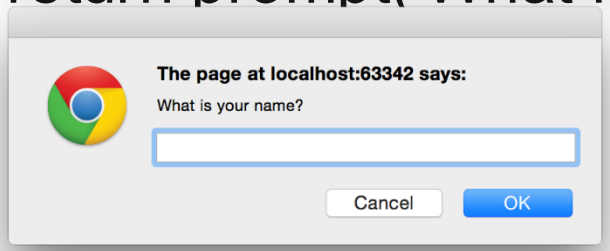
- alert('Good’);



- return confirm('Do you want X?');



- return prompt('What is your name?’);



- When you get input, use the return statement to send the value somewhere

Variables and Types

- Create variables using the var keyword:
  - var age = 18;
- Every variable has a type
- JavaScript has six different types of value. There are five primitive data types:
  - string
  - number
  - Boolean
  - undefined
  - null
- Everything else is an object

Properties and methods

- Objects have properties and methods
- (Primitive types also have these thanks to JavaScript implicitly creating wrapper objects so they appear to be objects)

```
>> name = "Abraham  "
<< "Abraham  "
>> typeof name
<< "string"
>> name.length
<< 9
>> name["length"]
<< 9
>> name.trim()
<< "Abraham"
```

>> Code

<< Output

JS is weakly (not strongly) typed

- Type coercion is the process of converting the type of a value in the background to try and make an operation work.

- "6" * 2 results in 12 (number)
- "6" + 2 results in "62" (string)

- Use type constructors to convert types more safely, e.g.
  - Number("23”)
  - String(6)

Functions

- In JavaScript, functions are considered to be just another value.
- This means that they do all the same tasks that other values and objects can do, such as be assigned to variables, changed and stored in arrays.
- You can even define a function inside another function.
- In technical terms, this means that functions are considered to be first-class objects in JavaScript.

Creating functions

- Function Literals

```
function goodbye() {
    alert("Goodbye World!");
}
```

- Function Expressions
  This assigns an anonymous function to a variable:

```
var goodbye = function(){
 alert("Goodbye World!");
};
```

In both cases, the function would be called like: `goodbye();`

Example

HTML:

```
<input type="text" name="email" id="email"
onBlur="changeColor(this)"
onFocus="changeStyle(this)">
```

event — function — parameter

JS:

```
function changeStyle(element) {
    element.style.color = "#FF0000";
}
```

HTML attribute — CSS property

JavaScript properties

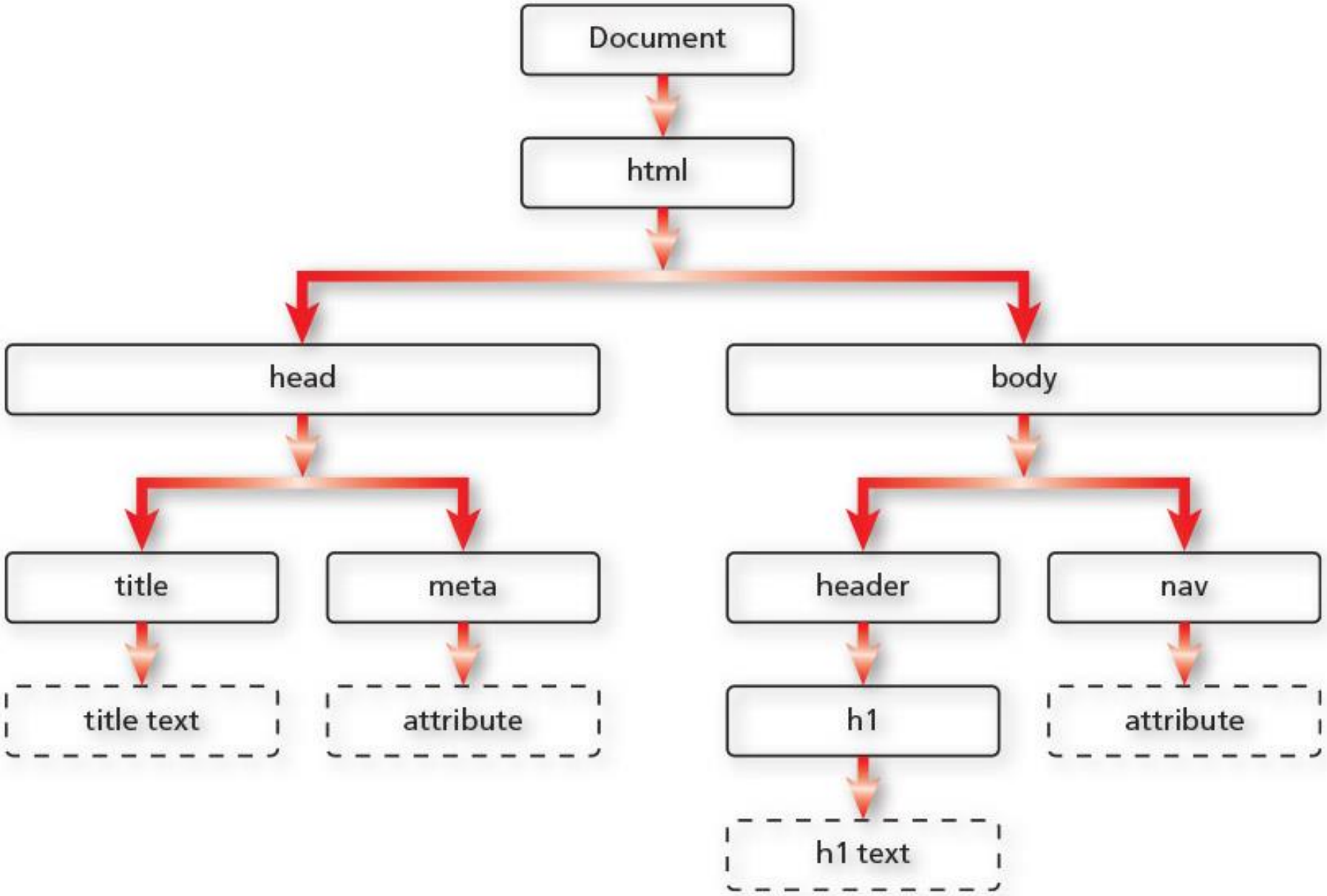
This is it

- this is a special keyword
- Inside the body of a function, this refers to the object the function is called on
- In HTML, this refers to the element where the code is e.g. here this = the input element with id=“email”:
- <input type="text" name="email" id="email" onBlur="changeColor(this)" onFocus="changeStyle(this)">

DOM Methods

- DOM stands for Document Object Model
  - Every element on an HTML page is an object
- HTML DOM consists of all the HTML elements, attributes, and text
  - Together, all of these items are objects on the page

DOM Methods


Document
html
head
body
title
meta
header
nav
title text
attribute
h1
attribute
h1 text

DOM Methods

Method	Description
getElementsByClassName( )	Returns all HTML elements with the specified class name
getElementsByName( )	Returns all HTML elements with the specified name value attribute
getElementsByTagName( )	Returns all HTML elements with the specified tag name
querySelector( )	Returns the first HTML element with the specified CSS selector
querySelectorAll( )	Returns a list of all HTML elements with the specified CSS selector

DOM Methods

DOM Methods

- Each item in the DOM tree is also known as a node
  - Any of these objects can be accessed using JavaScript DOM methods
  - Using a DOM method, you can use JavaScript to manipulate an HTML element

Using if/else Statements

- Common JavaScript statement used within a function that assesses a specified condition
  - Condition is true: a specific block of code is executed
  - Condition is false: a different block of code is executed

jQuery

- Commonly used library
  - JavaScript library that can significantly reduce the amount of code (and time) needed to complete a web project

Chapter Summary

- In this chapter, you learned how to:
  - Integrate interactivity using CSS and JavaScript
  - Use the CSS transform property
  - Create animations with CSS keyframes
  - Create a JavaScript file and write JavaScript functions
  - Use event handlers to call JavaScript functions