CHAPTER 3 – BUILDING ARRAYS AND CONTROLLING FLOW

Table of Contents

[Declaring and Initializing Arrays: 1](#_Toc13427621)

[Array literal 1](#_Toc13427622)

[Accessing Element Information: 2](#_Toc13427623)

[Array Object: 2](#_Toc13427624)

[Referencing Default Collections of Elements 2](#_Toc13427625)

[Repeating Code 2](#_Toc13427626)

[while statements 2](#_Toc13427627)

[do/while statements 3](#_Toc13427628)

[for statements 3](#_Toc13427629)

[Using continue Statements to Restart Execution 3](#_Toc13427630)

[continue statement 3](#_Toc13427631)

[Making Decisions 4](#_Toc13427632)

[if statement 4](#_Toc13427633)

[Else if statemets 4](#_Toc13427634)

[Switch statement 5](#_Toc13427635)

## Declaring and Initializing Arrays:

### Array literal

* + most common way to create an array
  + declares a variable and specifies array as content
  + JavaScript values assigned to array elements can be different data types

Syntax: var *name* = [*value1*, *value2*, *value3*, …];

Example:

Create an array named newsSections containing 4 strings as elements

var newsSections = ["world","local","opinion","sports"];

### Accessing Element Information:

* To access an element’s value; Include brackets and element index

Examples:

var sec1Head = document.getElementById("section1");

var sec2Head = document.getElementById("section2");

var sec3Head = document.getElementById("section3");

sec1Head.innerHTML = newsSections**[0];** // "world"

sec2Head.innerHTML = newsSections**[1]**; // "local"

sec3Head.innerHTML = newsSections**[2]**; // "opinion"

### Array Object:

* JavaScript represents arrays with the Array object

Contains a special constructor named Array()

Syntax: var newsSections = new Array(6);

Note: Create arrays using array literals (easier) rather than using Array() constructor.

## Referencing Default Collections of Elements

* getElementsByTagName() method

Can reference web page element by looking up all elements of a certain type in document and referencing one element in that collection

Resulting collection uses syntax similar to arrays

Example:

For instance, to reference the third li element in a document, you could use the getElementsByTagName() method to access the collection of all li elements in the document, and then add [2] to the end, indicating that you want to access the third instance of the li element in the document as follows:

document.getElementsByTagName("li")[2]

To access the content of this element, you could add the innerHTML property, as follows:

document.getElementsByTagName("li")[2].innerHTML

## Repeating Code

### while statements

Example:

assigning array element values to table cells:

function addColumnHeaders() {

var i = 0;

while (i < 7) {

document.getElementsByTagName("th")↵

[i].innerHTML = daysOfWeek[i];

i++;

}

}

### do/while statements

Example:

adding days of week with a do/while statement.

var i = 0;

do {

document.getElementsByTagName("th")[i].innerHTML =↵

daysOfWeek[i];

i++;

} while (i < 7);

### for statements

Repeats a statement or series of statements

As long as a given conditional expression evaluates to a truthy value

Can also include code that initializes a counter and changes its value with each iteration

Example:

addColumnHeaders() function with a for statement.

function addColumnHeaders() {

for (var i = 0; i < 7; i++) {

document.getElementsByTagName("th")[i].innerHTML↵

= daysOfWeek[i];

}

}

## Using continue Statements to Restart Execution

### continue statement

Halts a looping statement

Restarts the loop with a new iteration

Used to stop a loop for the current iteration

Have the loop to continue with a new iteration

Examples:

for loop with a continue statement

for (var count = 1; count <= 5; count++) {

if (count === 3) {

continue;

}

document.write("<p>" + count + "</p>");

}

## Making Decisions

Decision making

Process of determining the order in which statements execute in a program

### if statement

Most common type of decision-making statement

Example:

var today = "Tuesday"

if (today === "Monday") {

document.write("<p>Today is Monday</p>");

}

else {

document.write("<p>Today is not Monday</p>");

}

### Else if statemets

– is commonly used to enhance event listeners so they’re backward-compatible with older browsers. IE8 and earlier used the attachEvent method instead of the addEventListener method for creating event listeners. Here is an example that adds an event listener using the correct syntax.

Example:

Used to create backward-compatible event listeners:

var submitButton = document.getElementById("button");

if (submitButton.addEventListener) {

submitButton.addEventListener("click", submitForm,↵

false);

}

else if (submitButton.attachEvent) {

submitButton.attachEvent("onclick", submitForm);

}

### Switch statement

Controls program flow by executing a specific set of statements

Dependent on an expression value

Compares expression value to value contained within a case label

default label

Executes when the value returned by the switch statement expression does not match a case label

*break statement*

Ends execution of a switch statement

Should be final statement after each case label

Example:

function city\_location(americanCity) {

switch (americanCity) {

case "Boston":

return "Massachusetts";

break;

case "Chicago":

return "Illinois";

break;

case "Los Angeles":

return "California";

break;

case "Miami":

return "Florida";

break;

case "New York":

return "New York";

break;

default:

return "United States";

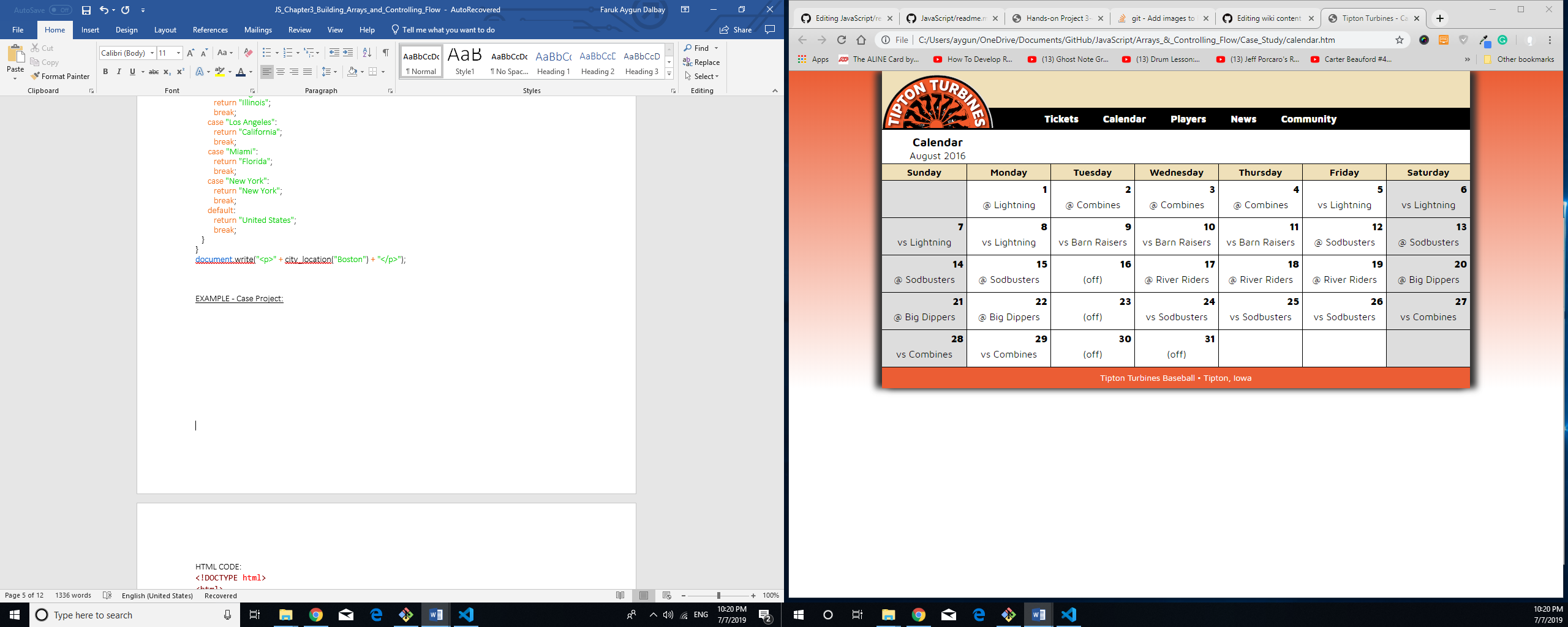
break;

}

}

document.write("<p>" + city\_location("Boston") + "</p>");

**EXAMPLE - Case Project:**



HTML CODE:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1.0">

<title>Tipton Turbines - Calendar</title>

<link rel="stylesheet" media="screen and (max-device-width: 999px)" href="tthand.css" />

<link rel="stylesheet" media="screen and (min-device-width: 1000px)" href="turbines.css" />

<!--[if lt IE 9]>

<link rel="stylesheet" href="turbines.css" />

<![endif]-->

<link href="http://fonts.googleapis.com/css?family=Maven+Pro:400,700,900" rel="stylesheet" type="text/css">

<script src="modernizr.custom.05819.js"></script>

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

</head>

<body>

<div id="container">

<header>

<h1>

<img src="images/ttlogo.png" alt="Tipton Turbines" width="182" height="93" title="" />

</h1>

</header>

<nav>

<ul>

<li><a href="#">Tickets</a></li>

<li><a href="#">Calendar</a></li>

<li><a href="#">Players</a></li>

<li><a href="#">News</a></li>

<li><a href="#">Community</a></li>

</ul>

</nav>

<article>

<h2>Calendar</h2>

<table>

<caption>August 2016</caption>

<thead>

<tr>

<th></th>

<th></th>

<th></th>

<th></th>

<th></th>

<th></th>

<th></th>

</tr>

</thead>

<tbody>

<tr>

<td class="differentMonth"></td>

<td id="08-1">

<p></p>

<p></p>

</td>

<td id="08-2">

<p></p>

<p></p>

</td>

<td id="08-3">

<p></p>

<p></p>

</td>

<td id="08-4">

<p></p>

<p></p>

</td>

<td id="08-5">

<p></p>

<p></p>

</td>

<td id="08-6">

<p></p>

<p></p>

</td>

</tr>

<tr>

<td id="08-7">

<p></p>

<p></p>

</td>

<td id="08-8">

<p></p>

<p></p>

</td>

<td id="08-9">

<p></p>

<p></p>

</td>

<td id="08-10">

<p></p>

<p></p>

</td>

<td id="08-11">

<p></p>

<p></p>

</td>

<td id="08-12">

<p></p>

<p></p>

</td>

<td id="08-13">

<p></p>

<p></p>

</td>

</tr>

<tr>

<td id="08-14">

<p></p>

<p></p>

</td>

<td id="08-15">

<p></p>

<p></p>

</td>

<td id="08-16">

<p></p>

<p></p>

</td>

<td id="08-17">

<p></p>

<p></p>

</td>

<td id="08-18">

<p></p>

<p></p>

</td>

<td id="08-19">

<p></p>

<p></p>

</td>

<td id="08-20">

<p></p>

<p></p>

</td>

</tr>

<tr>

<td id="08-21">

<p></p>

<p></p>

</td>

<td id="08-22">

<p></p>

<p></p>

</td>

<td id="08-23">

<p></p>

<p></p>

</td>

<td id="08-24">

<p></p>

<p></p>

</td>

<td id="08-25">

<p></p>

<p></p>

</td>

<td id="08-26">

<p></p>

<p></p>

</td>

<td id="08-27">

<p></p>

<p></p>

</td>

</tr>

<tr>

<td id="08-28">

<p></p>

<p></p>

</td>

<td id="08-29">

<p></p>

<p></p>

</td>

<td id="08-30">

<p></p>

<p></p>

</td>

<td id="08-31">

<p></p>

<p></p>

</td>

<td class="differentMonth"></td>

<td class="differentMonth"></td>

<td class="differentMonth"></td>

</tr>

</tbody>

</table>

</article>

<footer>

<p>Tipton Turbines Baseball &bull; Tipton, Iowa</p>

</footer>

</div>

</body>

</html>

JavaScript CODE:

// global variables

var daysOfWeek = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",

"Friday", "Saturday"];

var opponents = ["Lightning", "Combines", "Combines", "Combines", "Lightning",

"Lightning", "Lightning", "Lightning", "Barn Raisers", "Barn Raisers", "Barn Raisers", "Sodbusters", "Sodbusters", "Sodbusters","Sodbusters", "(off)", "River Riders", "River Riders", "River Riders", "Big Dippers", "Big Dippers", "Big Dippers", "(off)", "Sodbusters", "Sodbusters",

"Sodbusters", "Combines", "Combines", "Combines", "(off)", "(off)"];

var gameLocation = ["away", "away", "away", "away", "home", "home", "home",

"home", "home", "home", "home", "away", "away", "away", "away", "", "away", "away", "away", "away", "away","away", "", "home", "home", "home", "home", "home","home", "", ""];

// function to place daysOfWeek values in header row cells

function addColumnHeaders() {

var i = 0;

while (i < 7) {

document.getElementsByTagName("th")[i].innerHTML = daysOfWeek[i];

i++;

}

}

// function to place day of month value in first p element within each table data cell that has an id

function addCalendarDates() {

var i = 1;

var paragraphs = "";

do {

var tableCell = document.getElementById("08-" + i);

paragraphs = tableCell.getElementsByTagName("p");

paragraphs[0].innerHTML = i;

i++;

} while (i <= 31);

}

// function to place opponents and gameLocation values in second p element within each table data cell that has an id

function addGameInfo() {

var paragraphs = "";

for (var i = 0; i < 31; i++) {

var date = i+1;

var tableCell = document.getElementById("08-" + date);

paragraphs = tableCell.getElementsByTagName("p");

if (gameLocation[i] === "away") {

paragraphs[1].innerHTML = "@ ";

}

if (gameLocation[i] === "home") {

paragraphs[1].innerHTML = "vs ";

}

/\*switch (gameLocation[i]) {

case "away":

paragraphs[1].innerHTML = "@ ";

break;

case "home":

paragraphs[1].innerHTML = "vs ";

break;

}\*/

paragraphs[1].innerHTML += opponents[i];

}

}

// function to populate calendar

function setUpPage() {

addColumnHeaders();

addCalendarDates();

addGameInfo();

}

// runs setUpPage() function when page loads

if (window.addEventListener) {

window.addEventListener("load", setUpPage, false);

} else if (window.attachEvent) {

window.attachEvent("onload", setUpPage);

}