14.2 Lesson Summary - Objects, ES6, and Tables

As a mature programming language JavaScript offers a number of useful ways to process arrays of data. JavaScript provides ways of organizing more complex data into objects. JavaScript also provides more concise ways of writing functions.

Concept: JavaScript often relies on defining a function to be run multiple times or after the completion of a certain process. While it may initially be difficult to organize your code in this manner there are significant benefits to performance and functionality. A ***foreach*** method takes advantage of this approach by running a specified function on each element of an array. For example:

*var lettersArray = ["a", "b", "c", "d"];*

*lettersArray.forEach(function(letter) {*

*console.log(letter);*

*});*

* Activity: 01-Ins\_forEach, 02-Stu\_Movie\_Score
* Suppl link: <https://www.w3schools.com/jsref/jsref_foreach.asp>

Concept: JavaScript **objects** are similar in usage and syntax to Python dictionaries. A JavaScript object's code translates almost directly into JSON (JavaScript Object Notation). Like Python dictionaries JavaScript objects contain **key-value** pairs and can contain other objects and arrays. A simple movie object might look like the following:

*var movie = {*

*name: "Star Wars",*

*year: 1977,*

*};*

*console.log(movie.name);*

* Activity: 03-Ins\_JavaScript\_Objects, 04-Stu\_Word\_Counter

Concept: JavaScript, like Python, has a ***split*** function that can be used to break a string up into an array. For example:

*var stringArray = myString.split(" ");*

* Activity: 04-Stu\_Word\_Counter

Concept: In JavaScript the ***map*** method uses the provided callback function to create a new, transformed array. For example:

*var lettersArray = ["a", "b", "c", "d"];*

*var superLettersArray = lettersArray.map(function(letter){*

*return "super" + letter;*

*})*

*console.log(superLettersArray);*

* Activity: 05-Evr\_Map
* Suppl link: <https://www.w3schools.com/jsref/jsref_map.asp>

Concept: In JavaScript **anonymous** functions are functions that you don't name. **Arrow function** syntax simplifies the creation of anonymous functions. For example, the map function used in the previous super letters example can be rewritten to:

var superLettersArrayFromArrow = lettersArray.map(letter => "super" + letter)

* Activity: 06-Evr\_Arrow\_Functions

Concept: You can iterate through the **keys** and **values** of objects by using the *Object.entries* method, for example:

*var movie = {*

*name: "Star Wars",*

*year: 1977,*

*};*

*Object.entries(movie).forEach(([key, value]) => console.log(*

*`Key: ${key} and Value ${value}`));*

* Activity: 07-Ins\_Object\_Iteration, 08-Stu\_Object\_Iteration

Concept: The ***filter*** function can be used in conjunction with a custom function to extract elements of an array into a new array. For example:

*var lettersArray = ["d", "d", "a", "b", "c", "d"];*

*var dLetters = lettersArray.filter(letter => letter === "d");*

* Activity: 09-Ins\_Filter, 10-Stu\_Filters

Concept: HTML Tables can be used to organize data into tables for webpages. For example:

*<table>*

*<tr>*

*<th>ID</th>*

*<th>Name</th>*

*</tr>*

*<tr>*

*<td>1</td>*

*<td>Jane Doe</td>*

*</tr>*

*<tr>*

*<td>2</td>*

*<td>John Doe</td>*

*</tr>*

*</table>*

* Activity: 11-Ins\_HTML\_table
* Suppl link: <https://www.w3schools.com/html/html_tables.asp>