Using For Each Loops

#### **Learning Outcome Addressed**

1. Use callback functions to filter, sort and map arrays and objects in JavaScript

Follow the instructions in Next Tech to complete the activity.

**Working with ES6 functions**

**Using forEach Loops**

When working with an array, you will often find yourself iterating through its items to perform different operations on them. This is where forEach becomes a useful tool.

array.forEach allows you to loop through **each** item in the array, in the ascending order, without mutating the array.

The first argument of forEach(callback, [optional]) is the callback function that is called for each and every item in an array. Each element of an array is passed into a function and the function performs its actions on each element of the array. Let's see how this works:

let seasons = ['summer','fall','winter','spring']  
   
function printSeason(season){  
    console.log(season);  
}  
   
seasons.forEach((season) => printSeason(season));  
// expected result:  
// "summer"  
// "fall"  
// "winter"  
// "spring"

In this example, for each element in the seasons array, the callback function printSeason is called. Thus, each season is printed on the console.

**Now, it's your turn to use forEach**

Your task in this activity is to implement a function called createParagraph that takes an array of strings, and returns a single string containing all the items in the array separated by a space.

Task

Use forEach to create a single paragraph from a list of words.

**0**

Filetree

* ~/workspace
  + foreach.js
* **foreach.js**

1

// This is a list of words

let words = ['Loops', 'are', 'a', 'great', 'way', 'to', 'find', 'elements', 'in', 'an', 'array'];

// TODO: implement this function to return a string containing all words in a paragraph.

function createParagraph(words) {

  let paragraph = ' ';

}

// Prints paragraph to console

console.log(createParagraph(words));

// don't change this line

if (typeof module !== 'undefined') {

  module.exports = createParagraph;

}