April 30, 2019

Review the following article and be prepared to discuss on Thursday morning. Remember your critical thinking and bias.

https://reactkungfu.com/2015/09/common-react-dot-js-mistakes-unneeded-state/

Callback Exercise (Part 2) - April 29, 2019

Based on previous exercise, April 5, now write a function which logs to the console the total number of people, total age, and average age of people from BC and Alberta only, regardless of age.

```
let determineTotal = function() {write function here
};
//
determineTotal();
```

April 24, 2019

Review the following article and be prepared to discuss on Friday morning. Remember your critical thinking and bias.

https://codeburst.io/how-to-do-object-oriented-programming-the-right-way-1339c1a25286

Is there any syntax that you don't understand? What did you do about it?

April 23, 2019

Thursday we have a need that might interest some of your students.

We are looking for a few UX testers to test out and provide feedback for a new dashboard we are launching.

What is involved: we would only need 25 minutes of a person's time on Thursday, April 25th to play with the dashboard and offer their feedback to our director of product. This is unpaid.

Where: ATB Entrepreneur Centre, this Thursday, April 25th

If any of your students are available please have them select a convenient time by clicking on the calendly link below:

https://calendly.com/travisparkerm/bootkik-user-interviews?month=2019-04&date=2019-04-25

Leighton Healey | CEO
Mobile: +1.403.389.4680

bootkik

Website | Linkedin | Twitter | Instagram | Facebook
Bootkik is now live on the Apple and Google Play app stores!
Forbes: Bootkik is one of the 19 Innovative Tech Startups to Watch

Callback Exercise (Part 1) - April 5, 2019

You are working for a private company who looks after demographics of people living in BC and Alberta only. The data you received is from the 4 Western provinces.
Write two functions: 1) a function to process all of the people from the Western 4 provinces, selecting only BC and Alberta. This is your jurisdiction so this function will need to be in your company's general library to be used over and over again, 2) a callback function to log to the console the full names of people from BC and Alberta who are over 25 years of age.

Watch the following youtube video for assistance in working through this exercise:

https://www.youtube.com/watch?v=Nau-iEEgEoM

```
let people = [
      {fname:"Alex", lname:"Smith", province:"BC", age:33},
      {fname:"Angela", lname:"Jones", province:"AB", age:61},
      {fname:"Anne", lname:"Bird", province:"SK", age:35},
      {fname:"Brent", lname:"Riddle", province:"MN", age:79},
      {fname:"Byron", lname:"Cardenas", province:"BC", age:38},
      {fname:"Carrie", lname:"Ramirez", province:"AB", age:89},
      {fname:"Cheryl", lname:"Glenn", province:"SK", age:70},
      {fname:"Dima", lname:"Curry", province:"MN", age:67},
      {fname:"Dustin", lname:"Bullock", province:"BC", age:59},
      {fname:"Eva", lname:"Keiths", province:"AB", age:24},
      {fname:"Faith", lname:"Liu", province:"SK", age:46},
      {fname:"Fawad", lname:"Bowman", province:"MN", age:69},
      {fname:"Forest", lname:"Vaughn", province:"BC", age:52},
      {fname:"Giovanni", lname:"Browning", province:"AB", age:32},
      {fname:"Greg", lname:"Hogan", province:"SK", age:55},
      {fname:"Harpreet", lname:"Ramsey", province:"MN", age:18},
      {fname:"Ian", lname:"Fitzgerald", province:"BC", age:16},
      {fname:"James", lname:"Kramer", province:"AB", age:57},
      {fname:"Jarvis", lname:"Ortega", province:"SK", age:69},
      {fname:"Jawad", lname:"Huerta", province:"MN", age:35},
      {fname:"Jinbong", lname:"Robinson", province:"BC", age:26},
      {fname:"Jingnan", lname:"Hatfield", province:"AB", age:71},
      {fname:"Joe", lname:"Banks", province:"SK", age:37},
      {fname:"Kristina", lname:"Dalton", province:"MN", age:73},
      {fname:"Latora", lname:"Matthews", province:"BC", age:25},
      {fname:"Lauren", lname:"McClure", province:"AB", age:42},
      {fname:"Licedt", lname:"Rasmussen", province:"SK", age:30},
      {fname:"Linden", lname:"Pierce", province:"MN", age:68},
      {fname:"Luis", lname:"Price", province:"BC", age:23},
      {fname:"Marcela", lname:"Perez", province:"AB", age:20},
      {fname:"Marilou", lname:"Graham", province:"SK", age:32},
      {fname:"Matt", lname:"Novak", province:"MN", age:29},
      {fname:"Monica", lname:"Giles", province:"BC", age:34},
      {fname:"Niloufar", lname:"Carson", province:"AB", age:29},
      {fname:"Omar", lname:"Olson", province:"SK", age:69},
      {fname:"Roger", lname:"Woodard", province:"MN", age:84},
      {fname:"Roman", lname:"Swanson", province:"BC", age:21},
      {fname:"Seun", lname:"Kelly", province:"AB", age:60},
      {fname:"Shane", lname:"Frost", province:"SK", age:87},
      {fname:"Steven", lname:"Haynes", province:"MN", age:47},
      {fname:"Thomas", lname:"Hart", province:"BC", age:14},
      {fname:"Trent", lname:"Kerr", province:"AB", age:12},
      {fname:"Darrell", lname:"Koch", province:"SK", age:10},
      {fname:"Tylor", lname:"Torres", province:"MN", age:98}
];

let processPeople = function(data, callback) {write function here
      which will execute callback
}
//
// Here is invoking the processPeople function.  Write the callback
// function as an anonymous function
//
processPeople(people, function(item) {write callback function here
});
```

More Array Exercises (you’re freaking kidding me) - March 29, 2019

Write a reduce function to total the balances.

```
let tot = data.staff.reduce((??? insert code here ???));
console.log(tot);
assertEquals(tot, 3823);
```

More Array Exercises (Really) - March 28, 2019

Using one of the callback functions we researched in the last exercise (but may have not used previously) create a new array for balances >= 1000.
Copy and paste the following code.

```
let largeBalances = data.staff.??? (insert code here)  ???
// console.log(largeBalances);
assertEquals(largeBalances[0].fname, "Liam");
assertEquals(largeBalances[1].fname, "Emma");
```

More Array Exercises - March 21, 2019

Update your requests in $Request for Session.

Do not use JavaScript functions like total, sum.

1. write a function to receive the same array (staff) and return the total of balances
2. create a table of objects of people having name and age
3. write a function to receive the above array and return the total of the ages
4. write a function to receive the above array and return the average the ages

Understand the documentation. Now that we have a few arrays let’s practise using the following:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/reduce
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/every
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/some
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/find
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/findindex

loopStaff each / map - March 20, 2019

Do the same assignment again now starting to use callback functions.

```
console.log('-----emailWithEach')
const emailWithEach = loopWithEach(data.staff);
// console.log(emailWithEach);
assertEquals('Jane.Smith@evolveu.ca', emailWithEach[0]);
assertEquals('Olivia.Notly@evolveu.com', emailWithEach[3]);
assertEquals('Benjamin.Amis@evolveu.ca', emailWithEach[6]);

console.log('-----emailWithMap')
const emailWithMap = loopWithMap(data.staff);
// console.log(emailWithMap);
assertEquals('Jane.Smith@evolveu.ca', emailWithMap[0]);
assertEquals('Olivia.Notly@evolveu.com', emailWithMap[3]);
assertEquals('Benjamin.Amis@evolveu.ca', emailWithMap[6]);
```

March 19, 2019 - Noon Exercise

Things to study for this week, working with arrays.
Next: DOM, Objects

Do the same assignment as yesterday using the two forms of for:

```
console.log('-----emailForOf')
const emailForOf = loopWithForOf(data.staff);
// console.log(emailForOf);
assertEquals('Jane.Smith@evolveu.ca', emailForOf[0]);
assertEquals('Olivia.Notly@evolveu.com', emailForOf[3]);
assertEquals('Benjamin.Amis@evolveu.ca', emailForOf[6]);

console.log('-----emailForIn')
const emailForIn = loopWithForIn(data.staff);
// console.log(emailForIn);
assertEquals('Jane.Smith@evolveu.ca', emailForIn[0]);
assertEquals('Olivia.Notly@evolveu.com', emailForIn[3]);
assertEquals('Benjamin.Amis@evolveu.ca', emailForIn[6]);
```

loopStaff - March 18, 2019

This exercise will use two functions you have written in previous exercises: makeEmailObj and assertEquals.

Write a function that will take an array and return an array of emails. The new function you are writing today will call makeEmailObj that you written in a previous exercise.

Copy and paste the code provided. Only put code in-between the two comment lines.

```
const data = {
       staff: [
              {fname:"Jane", lname:"Smith", balance:10},
              {fname:"Liam", lname:"Henry", balance:1000},
              {fname:"Emma", lname:"Jones", balance:1330},
              {fname:"Olivia", lname:"Notly", balance:310},
              {fname:"Noah", lname:"Ho", balance:503},
              {fname:"William", lname:"Lee", balance:520},
              {fname:"Benjamin", lname:"Amis", balance:150},
       ],
       company: "EvolveU",
       city: "Calgary",
       prov: "Alberta"
};

// Write the function after this comment ---

// and before this comment ---

console.log('-----loopStaff')
const staffEmail = loopStaff(data.staff);
// console.log(staffEmail);
assertEquals('Jane.Smith@evolveu.ca', staffEmail[0]);
assertEquals('Olivia.Notly@evolveu.com', staffEmail[3]);
assertEquals('Benjamin.Amis@evolveu.ca', staffEmail[6]);
```

makeEmailObj - March 14, 2019

Update your requests in [$Request for Session](#).

Things to study for the next few days. Working with arrays:

- Basics
    - for
    - while
    - do while
- Next Level
    - for in
    - for of
    - forEach
- https://medium.com/poka-techblog/simplify-your-javascript-use-map-reduce-and-filter-bd02c593cc2d
    - map
    - reduce
    - filter

Exercise

Write a function that will receive an object / map. Create an evolveu email from the object.

Copy and paste the code provided. Only put code in-between the two comment lines.

```
/*
	Write the function to format an email based on an object / map
*/

// Write the function after this comment ---

// and before this comment ---

const objLarry = {fname:'larry', lname:'shumlich'};
assertEquals('larry.shumlich@evolveu.ca', makeEmailObj(objLarry));
assertEquals('bill.smith@evolveu.ca', makeEmailObj({fname:'bill',lname:'smith'}));
assertEquals('amy.jones@evolveu.ca', makeEmailObj({fname:'amy',lname:'jones'}));
```

makeEmailArr - March 13, 2019

This exercise will build on our last exercise. We want to use the ‘assertEquals’ function that we wrote last time. We want to share code we wrote between exercises. What are the options for sharing? Pick one sharing option for this exercise and we will talk about it as a group.

Make one update to the ‘assertEquals’ function. If the two parameters are equal log ‘ok ->’ followed by only the first parameter.

Write a function that will receive an array. The first entree in the array is the first name, the second entree is the last name. The array only has one person in it. Create an evolveu email from the array.

Copy and paste the code provided. Only put code in-between the two comment lines.

```
/*
      Write the function to format an email based on an array
*/

// Write the function after this comment ---

// and before this comment ---

const arrayLarry = ['larry', 'shumlich'];
assertEquals('larry.shumlich@evolveu.ca', makeEmailArr(arrayLarry));
assertEquals('bill.smith@evolveu.com', makeEmailArr(['bill','smith']));
assertEquals('amy.jones@evolveu.ca', makeEmailArr(['amy','jones']));
```

AssertEquals - March 11, 2013

Write a function that will receive two parameters. The function will compare the parameters and do the following:

- if the two parameters are equal, log nothing and return true
- if the two parameters are not equal, log to the console the following and return false:

“ *** the two values are not the same” and also log the two parameters as follows:
    “p1-->” parameter 1
    “p2-->” parameter 2

Copy and paste the code provided. Only put code in-between the two comment lines.

```
/*
      Write the function that will create this output:

*** the two values are not the same:
    p1--> a
    p2--> b
*** the two values are not the same:
    p1--> 1
    p2--> 2
*** the two values are not the same:
    p1--> 2
    p2--> 2
*/

// Write the function after this comment ---

// and before this comment ---

assertEquals("a","b");
assertEquals("a","a");
assertEquals(1,2);
assertEquals(2,2);
assertEquals("2",2);
assertEquals("This value","This value");
```