JAVASCRIPT HANDS-ON

1. JAVASCRIPT ARRAY ANSWERS

Objective

There is index.html file and index.js file in "app" folder. This "app" folder is available inside "javascript-array" folder. You need to complete code in these files. Use available IDE to edit these files OR you may also use vi editor commands.

Instruction

Create a page for a Car dealer.

UI requirement (for index.html): Create a label "Please Choose a Car to get its Details" followed by a dropdown box with id: "SelectCar". The dropdown values are: Innova, Dzire,i20 & i10

Test Data to be provided: carName - Innova, Dzire, i20 & i10
onRoadPrice - 900000 , 700000, 500000 & 400000 respectively
yearOfModel - 2016, 2017, 2013 & 2016 respectively

Script logic (for index.js): Create an array of Cars in below mentioned Javascript function with attributes & values as.
carName – Datatype:String
onRoadPrice- Datatype:number
yearOfModel- Datatype:number

On changing the option in the dropdown, a javascript function displayCarDetails has to be called to fetch the related details of the car from the array of cars. It has to be displayed inside the para tag with id "CarDetail". The format to display would be carName-onRoadPrice-yearOfModel, i.e if i10 is selected ,then it should display i10-500000-2013.

- Use `cd javascript-array` command to move into application folder.
- Use `npm install` to install the required packages.
- Use `npm start` to launch the application.

For web preview click the below link.

http://2886795399-8000-xplore01.environments.katacoda.com/

To test your application Use `npm test` command.

Make sure your passing both the test cases before clicking summary button.

ANSWERS

```
<html>
```

```
<head><script type="text/javascript" src="index.js"></script></head>

<body>

<label>Please Choose a Car to get its Details</label></br>

<select id="SelectCar">

<option value="Innova">Innova</option>

<option value="Dzire">Dzire</option>

<option value="i20">i20</option>

<option value="i10">i10</option></select>

<button onclick="displayCarDetails()">Get Data</button>

<p id="CarDetail"></p>

</body>

</html>
```

Index.js

```
let carDetails=[["Innova","900000","2016"],["Dzire","700000","2017"],["i20","500000","2013"],["i10","400000","2016"]];

function displayCarDetails()

{

        var x=document.getElementById('SelectCar').value;

if(x=="Innova")

document.getElementById('CarDetail').innerHTML=carDetails[0][0]+"-"+carDetails[0][1]+"-"+carDetails[0][2];

if(x=="Dzire")

document.getElementById('CarDetail').innerHTML=carDetails[1][0]+"-"+carDetails[1][1]+"-"+carDetails[1][2];

if(x=="i20")

document.getElementById('CarDetail').innerHTML=carDetails[2][0]+"-"+carDetails[2][1]+"-"+carDetails[2][2];

if(x=="i10")

document.getElementById('CarDetail').innerHTML=carDetails[3][0]+"-"+carDetails[3][1]+"-"+carDetails[3][2];

}
```

2. JAVASCRIPT DATATYPES

Objective

There is index.html file and index.js file in "app" folder. This "app" folder is available inside "javascript_datatypes" folder. You need to complete code in these files. Use available IDE to edit these files OR you may also use vi editor commands.

Instruction

Passenger Management

UI requirement (for index.html): The page should be created with a white background. Create a submit button with label "show".

Please note, all the Labels, attribute names and Ids mentioned are case sensitive.

Script logic (for index.js): Create a javascript function named "callMe", on click of the "show" button, the JavaScript Object named “Passenger” with the below fields should be displayed using document.write.

Field Name - name, age & reservedStatus
Data Type - String, Number & Boolean respectively
Value to be assigned - Arun, 28, true respectively.

The document.write should be written using the following.
With a div tag mapped to id “name” which should display “Name:XXXX”, where XXXX is the name of the passenger assigned to the object.
With another div tag mapped to id “age” which should display “Age:XXXX”, where XXXX is the age of the passenger assigned to the object.
With one more div tag mapped to id “reservedStatus” which should display the “Reservation Status:XXXX”, where XXXX is the reservedStatus of the passenger assigned to the object.

- Use `cd javascript_datatypes` command to move into application folder.
- Use `npm install` to install the required packages.
- Use `npm start` to launch the application.

For web preview click the below link.

https://2886795283-8000-xplore02.environments.katacoda.com/

To test your application Use `npm test` command.

Make sure your passing all the test cases before clicking summary button.

ANSWERS

```
<html>
<head>
  <title>Passenger Management</title>
  <script src="index2.js"></script>
</head>
<body>
  <button onclick="callMe()">Show</button>
</body>
</html>
```

Index.js

```
function callMe() {
  var name = "Arun";
  var age = 28;
  var reservedStatus = true;
  String(name);
  Number(age);
  Boolean(reservedStatus);

  document.write("<div id='name'>"+name+"</div>");
  document.write("<div id='age'>"+age+"</div>");
  document.write("<div id='reservedStatus'>"+reservedStatus+"</div>");
}
```

3. JAVASCRIPT DOM

Objectives

There is index.html file and index.js file in "app" folder. This "app" folder is available inside "javascript-dom" folder. You need to complete code in these files. Use available IDE to edit these files OR you may also use vi editor commands.

Instruction

'ICB' is an Online Recruitment survey system. Their major goal is to identify the age of the respondents and then branch them to a survey based on their age .Write a JS program to calculate the exact age (in years) and display it in the browser.

UI requirement (for index.html):

1. Create a h2 tag with the message “JavaScript Objects”
2. Create a label 'Enter Your DOB' followed by textbox (Date filed). The textbox should have attribute name and id mapped to value "dob".
3. Create a submit button inside the form, using the input type as Submit.

Script logic (for index.js): Write a function getAge() which uses getElementById to get the DOB from the above field and calculates the Age(in Yrs). Just consider the year, month and date can be ignored. If the selected year is the current or future year, display an error message “Wrong date!! in the paragraph tag with id “showresults”

Once the form is submitted , the paragraph tag with id “showresults” should display results in below format: “You are X year(s) old!!”.
Where X mentioned above, is the calculated age of the person (in Yrs).

- Use `cd javascript-dom` command to move into application folder.
- Use `npm install` to install the required packages.
- Use `npm start` to launch the application.

For web preview click the below link.

https://2886795312-8000-xplore04.environments.katacoda.com/

To test your application Use `npm test` command.

Make sure your passing both the test cases before clicking summary button.

ANSWERS

```
<html>

<head><script src="index.js" type="text/javascript"></script></head>

<body>

<h2>JavaScript Objects</h2>

<label>Enter your DOB</label>

<form><input type="text" id="dob" name="dob"></form>

<button type="submit" onclick="getAge()">submit</button>

<p id="showresults"></p>

</body>

</html>
```

Index.js

```
function getAge()

{

var x=document.getElementById('dob').value;

var d=new Date();

var dd=d.getFullYear();

var xx=x.substring(0,4);

if(xx<dd)

{

var y=dd-xx;

document.getElementById('showresults').innerText="You are "+y+" year(s) old!!";

}

else

{

document.getElementById('showresults').innerText="Wrong date!!";

}

}
```

4. JAVASCRIPT EVENTS

Objective

There is index.html file and index.js file in "app" folder. This "app" folder is available inside "javascript-events" folder. You need to complete code in these files. Use available IDE to edit these files OR you may also use vi editor commands.

Instruction

Create a sample form to register a User with the following UI controls

UI requirement (for index.html):

1. Label "UserId" followed by textbox. The textbox should have attributes name and id mapped to value "uid".
2. Label "Password" followed by textbox. The textbox should have attributes name and id mapped to value "password".
3. Label "ConfirmPassword" followed by textbox. The textbox should have attributes name and id mapped to value "confirm".

4. By the side of each textbox there must be a div tag with id “msg”{fieldName}, where {fieldName} is the value provided to the name attribute for each of the textbox elements (follow java naming convention – camel casing).
5. Create a button with id “create”.

Script logic (for index.js): Create a Java script function called validate to perform the below validation

1. All the inputs are mandatory.
2. The values entered in the password and confirm password should be the same.
3. Show error message (Invalid Entry), in the div tag next to the fields when proper values are not entered.
4. If expected values are entered (OK!) should be displayed, in the div tag next to the fields.

Messages(Invalid Entry / OK!) should be shown both on onkeyup as well as on button(create) click events

- Use `cd javascript-events` command to move into application folder.
- Use `npm install` to install the required packages.
- Use `npm start` to launch the application.

For web preview click the below link.

https://2886795290-8000-xplore01.environments.katacoda.com/

To test your application Use `npm test` command.

Make sure your passing both the test cases before clicking summary button.

ANSWERS

```
<html>

<head></head>

<body>

<form>

<label>UserId <Input type="text" name="uid" id="uid" onkeyup="validate();"/></label>

<label>Password <Input type="password" name="password" id="password" onkeyup="validate();"/></label>

<label>Confirm Password <Input type="password" name="confirm" id="confirm" onkeyup="validate();"/></label>

<button id="create" type="button" onclick="validate()">Create</button>

</form>

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

```
</body>

</html>
```

Index.js

```
function validate()
{
        var x=document.getElementById("uid").value;
        var y=document.getElementById("password").value;
        var z=document.getElementById("confirm").value;
        if(x=="")
        {
          document.getElementById("msgUid").innerHTML="Invalid Entry";
        }
        else
          {
          document.getElementById("msgUid").innerHTML="OK!";
        }
        if(y=="")
        {
          document.getElementById("msgPassword").innerHTML="Invalid Entry";
        }
        else if(y!=z)
        {
          document.getElementById("msgPassword").innerHTML="Invalid Entry";
        }
        else
          {
          document.getElementById("msgPassword").innerHTML="OK!";
        }
        if(z=="")
        {
```

```
      document.getElementById("msgConfirm").innerHTML="Invalid Entry";
     }
     else if(y!=z)
     {
      document.getElementById("msgConfirm").innerHTML="Invalid Entry";
     }
     else
      {
      document.getElementById("msgConfirm").innerHTML="OK!";
     }
 }
```

5. JAVASCRIPT LOOP

Objective

There is index.html file and index.js file in "app" folder. This "app" folder is available inside "javascript-loop" folder. You need to complete code in these files. Use available IDE to edit these files OR you may also use vi editor commands.

Instruction

Write a program to print the sum of Fibonacci series less than 100 (i.e only till 100, fibonacci series exceeding 100 is not required)

UI requirement (for index.html): Create a button named “Generate Series” with id “generate”. On clicking the button, it should show the result I.e the sum of the Fibonacci series less than 100 in the para tag with id “result”.

Script logic (for index.js): create a javascript function called SumOfSeries and place the sum in the para tag with id “result”.

- Use `cd javascript-loop` command to move into application folder.
- Use `npm install` to install the required packages.
- Use `npm start` to launch the application.

For web preview click the below link.

https://2886795291-8000-xplore01.environments.katacoda.com/

To test your application Use `npm test` command.

Make sure your pass the test case before clicking summary button.

ANSWERS

```
<html>

<head></head>

<body>

<p id="result"></p>

<button onclick="SumOfSeries()" id="generate">Generate Series</button>

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

</body>

</html>
```

Index.js

```
function SumOfSeries(){

                var i=0;

                var j=1;

                var sum=0;

                while(j<100)

                {

                  var k=i;

                  i=j;

                  j=k+j;

                  sum=i+j-1;

                }

                document.getElementById("result").innerText=sum;

 }
```