Name: Aswin s

Email: aswinsrini1@gmail.com

GUVI TASK-2

11-02-2021

1.HTML and Script.js and run a for loop on the data and print all the country name in the console.

HTML:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>sample api data</title>
    <link rel="stylesheet" href="task.css">
</head>
<body>
    <script src="taskjs.js">

        </script>
</body>
</html>
```

SCRIPTJS

```javascript
var req =new XMLHttpRequest();
req.open("GET",'https://restcountries.eu/rest/v2/all',true);
req.send()
req.onload = function(){
    var data=JSON.parse(this.response);

    for(var i in data)
    {
        console.log(data[i].name);
        document.write("<br>");
        document.write(data[i].name);
        document.write(`  -  ${data[i].capital}`);
        document.write("<br>");
    }
}
```

2.Write a difference between call by value and call by reference

Call by value:

Ex

Var a =50;

Var b=a;

Here a is stored in address x00FF

And b is stored in different address say x0AAC

If we change the value of a=55 . again the value of b is not changed

Call by reference:

Ex

Var arr=[1,2,3]

Var acc=arr;

Now arr and acc points to the same memory location If any change made in any of those will change the value of both

1.Jason Task…

….1.cat

```
//add height weighy
cat.height=15;
cat.weight=5;

// fluffy name change
cat.name="fluffyy";

//cat friends name
for(var i in cat.catFriends)
{
    console.log(cat.catFriends[i].name);
}

//activites of fluffy cat friends

for(i in cat.catFriends)
{

    console.log("activities of ",cat.catFriends[i].name," ",cat.catFriends[i].activities);
}

//total weight of cat friends
let tw=0;
for(i in cat.catFriends)
{
    tw+=cat.catFriends[i].weight;
}
console.log("total weight of cat friends:",tw);

//acctivies of all cat
var tacc=cat.activities.length;
for(i in cat.catFriends)
{
    tacc+=cat.catFriends[i].activities.length;
}
console.log("\n total activitiesof all cat",tacc);

//add two more activites of bar and foo;
for(i in cat.catFriends)
{
    cat.catFriends[i].activities.push("scratching posts");
    cat.catFriends[i].activities.push("finding rat");

}
console.log(cat.catFriends[0].activities);
console.log(cat.catFriends[1].activities);
```

Output:

```
cat friends name are
bar
foo


activities of  bar   [ 'be grumpy', 'eat bread omblet' ]
activities of  foo   [ 'sleep', 'pre-sleep naps' ]


total weight of cat friends: 11

 total activitiesof all cat 5


the new activites are

acctiviesof cat name  bar   [ 'be grumpy', 'eat bread omblet',
  'scratching posts',
  'finding rat' ]
acctiviesof cat name  foo   [ 'sleep', 'pre-sleep naps', 'scratching posts', 'finding rat' ]
```

2.Mycar

```
var myCar = {
    make: "Bugatti",
    model: "Bugatti La Voiture Noire",
    year: 2019,
    accidents: [
    {
    date: "3/15/2019",
    damage_points: "5000",
    atFaultForAccident: true
    },
    {
    date: "7/4/2022",
    damage_points: "2200",
    atFaultForAccident: true
    },
    {
    date: "6/22/2021",
    damage_points: "7900",
    atFaultForAccident: true
    }
    ]
   }

   //loop over accident Array change atFaultForAccident to false
   for(var i in myCar.accidents)
   {
       myCar.accidents[i].atFaultForAccident=false;
       console.log(myCar.accidents[i]);
   }
   console.log("\n");

   //print date of my accidents
   var x=1
   for(i in myCar.accidents)
   {
       console.log(x,'-',myCar.accidents[i].date);
       x++;
   }
```

Output:

```
{ date: '3/15/2019',
  damage_points: '5000',
  atFaultForAccident: false }
{ date: '7/4/2022',
  damage_points: '2200',
  atFaultForAccident: false }
{ date: '6/22/2021',
  damage_points: '7900',
  atFaultForAccident: false }


1 '-' '3/15/2019'
2 '-' '7/4/2022'
3 '-' '6/22/2021'
```

3.Pass object as function and print array

```
function printAllvalues(obj)
{
    var arr=[]
    for(var i in obj)
    {
        arr.push(obj[i]);
    }
    return arr;
}

var obj = {name:"Rajinikanth", age: 33, hasPets : false};
  var ans=printAllvalues(obj);
  console.log(ans);
```

Output:

```
[ 'Rajinikanth', 33, false ]
```

Execution Time:

```
0.075s
```

Memory Used:

```
8312kb
```

4. Write a function called “printAllKeys” which returns an newArray of all the input object’s keys.

```
function printAllvalues(obj)
{
    var arr=[]
    for(var i in obj)
    {
        arr.push(i);
    }
    return arr;
}

var obj = {name:"Rajinikanth", age: 33, hasPets : false};
  var ans=printAllvalues(obj);
  console.log(ans);
```

Output:

```
[ 'name', 'age', 'hasPets' ]
```

Execution Time:

0.077s

Memory Used:

8344kb

5. Parsing an JSON object and convert it to a list:

```
function printAllvalues(obj)
{
    var arr=[]
    for(var i in obj)
    {
        var arr1=[];
        arr1.push(i);
        arr1.push(obj[i]);
        arr.push(arr1);
    }
    return arr;
}

var object = {name: "ISRO", age: 35, role: "Scientist"};
var ans=printAllvalues(object);
console.log(ans);
```

Output:

```
[ [ 'name', 'ISRO' ], [ 'age', 35 ], [ 'role', 'Scientist' ] ]
```

Execution Time:

```
0.076s
```

Memory Used:

```
8328kb
```

Parsing a list and transform the first and last elements of it:

```
function transformFirstAndLast(arr) {
    var obj={};
    var key=arr[0];
    obj[arr[0]]=arr[arr.length-1];
       return obj;
   }
   var array = ["GUVI","IAM","A","GEEK"];
   var obj=transformFirstAndLast(array);
   console.log(obj);
```

Output:

```
{ GUVI: 'GEEK' }
```

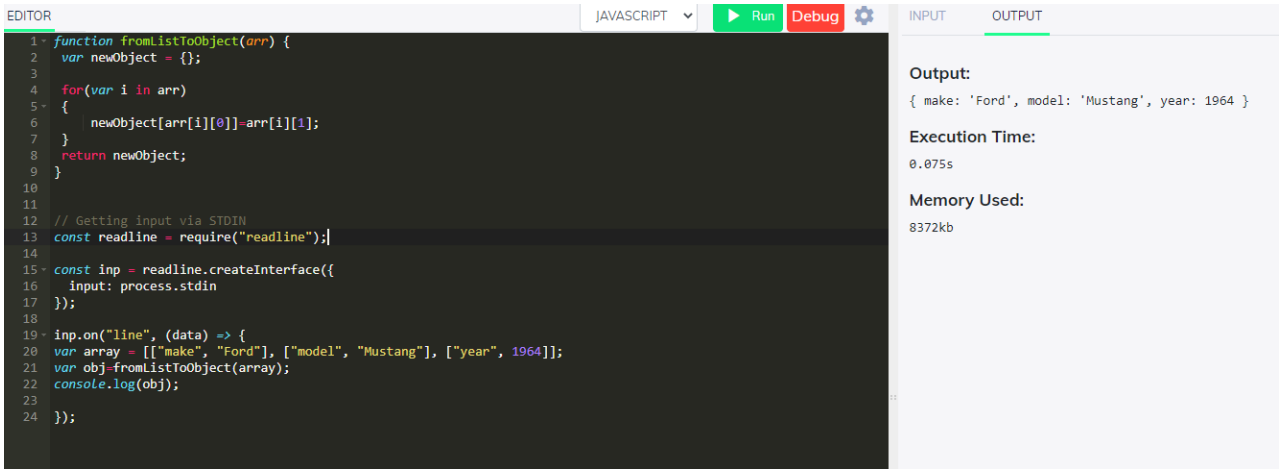
Execution Time:

0.075s

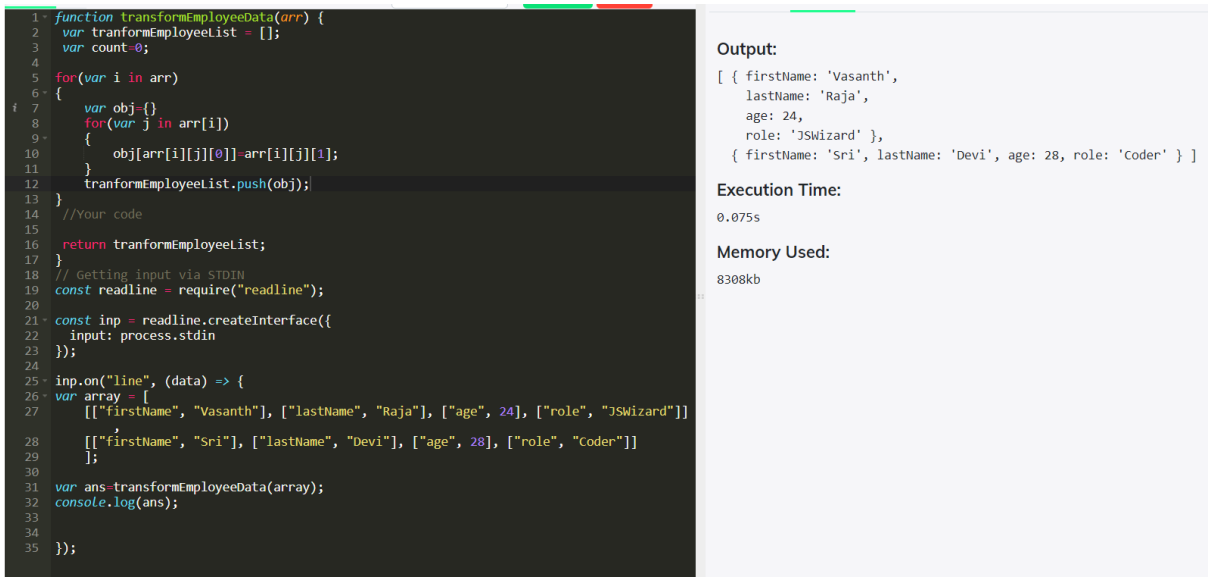
Memory Used:

8404kb

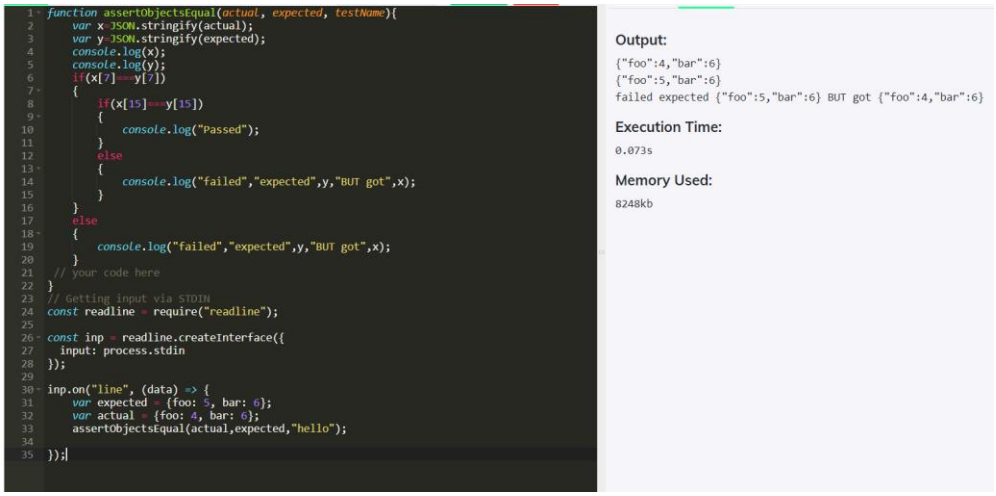
Parsing a list of lists and convert into a JSON object:



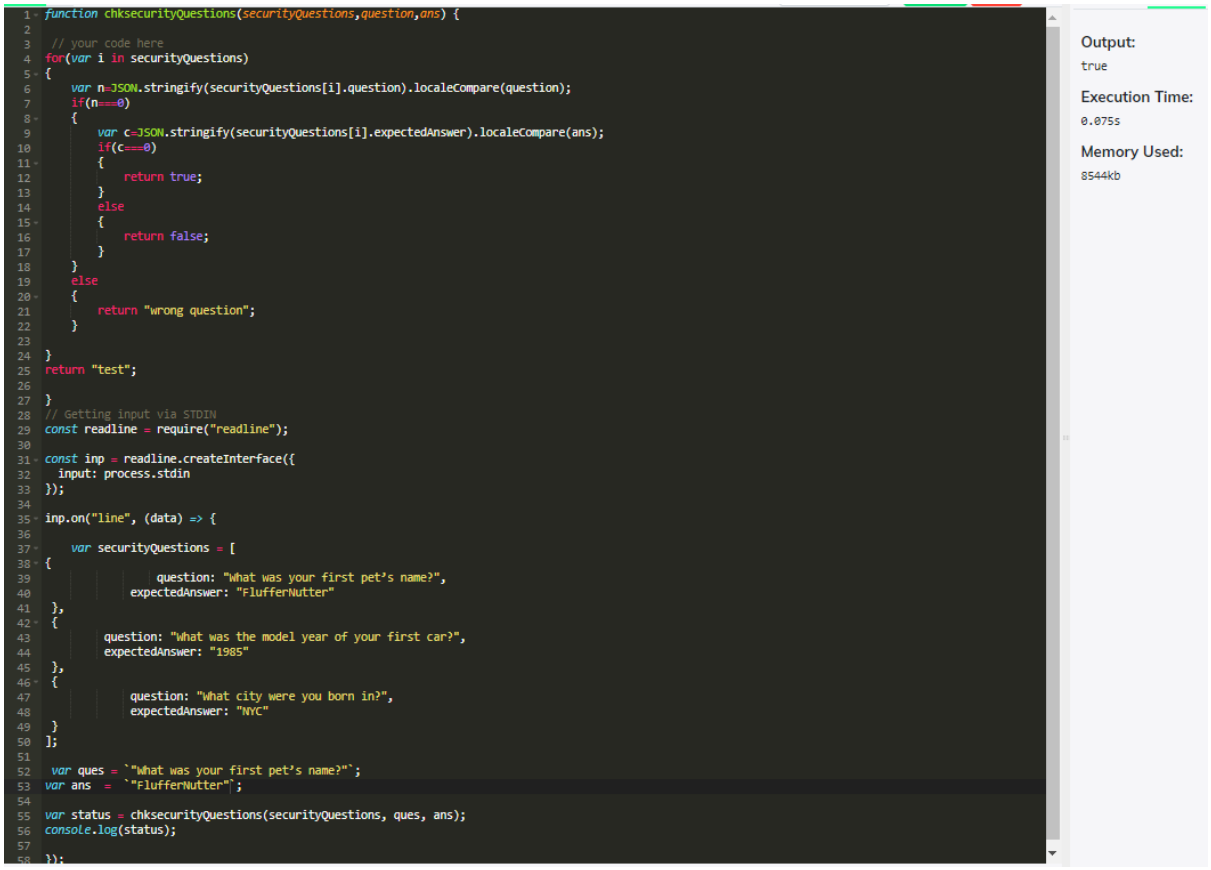
Parsing a list of lists and convert into a JSON object:



Parsing two JSON objects and Compare:



Parsing JSON objects and Compare:



```
function chksecurityQuestions(securityQuestions,question,ans) {

 // your code here
for(var i in securityQuestions)
{
    var n=JSON.stringify(securityQuestions[i].question).localeCompare(question);
    if(n===0)
    {
        var c=JSON.stringify(securityQuestions[i].expectedAnswer).localeCompare(ans);
        if(c===0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else
    {
        return "wrong question";
    }

}
return "test";

}
// Getting input via STDIN
const readline = require("readline");

const inp = readline.createInterface({
  input: process.stdin
});

inp.on("line", (data) => {

    var securityQuestions = [
{
            question: "What was your first pet's name?",
          expectedAnswer: "FlufferNutter"
 },
 {
        question: "What was the model year of your first car?",
        expectedAnswer: "1985"
 },
 {
            question: "What city were you born in?",
            expectedAnswer: "NYC"
 }
];

 var ques = `"What was your first pet's name?"`;
var ans  =  `"FlufferNutter"`;

var status = chksecurityQuestions(securityQuestions, ques, ans);
console.log(status);

});
```

Output:

true

Execution Time:

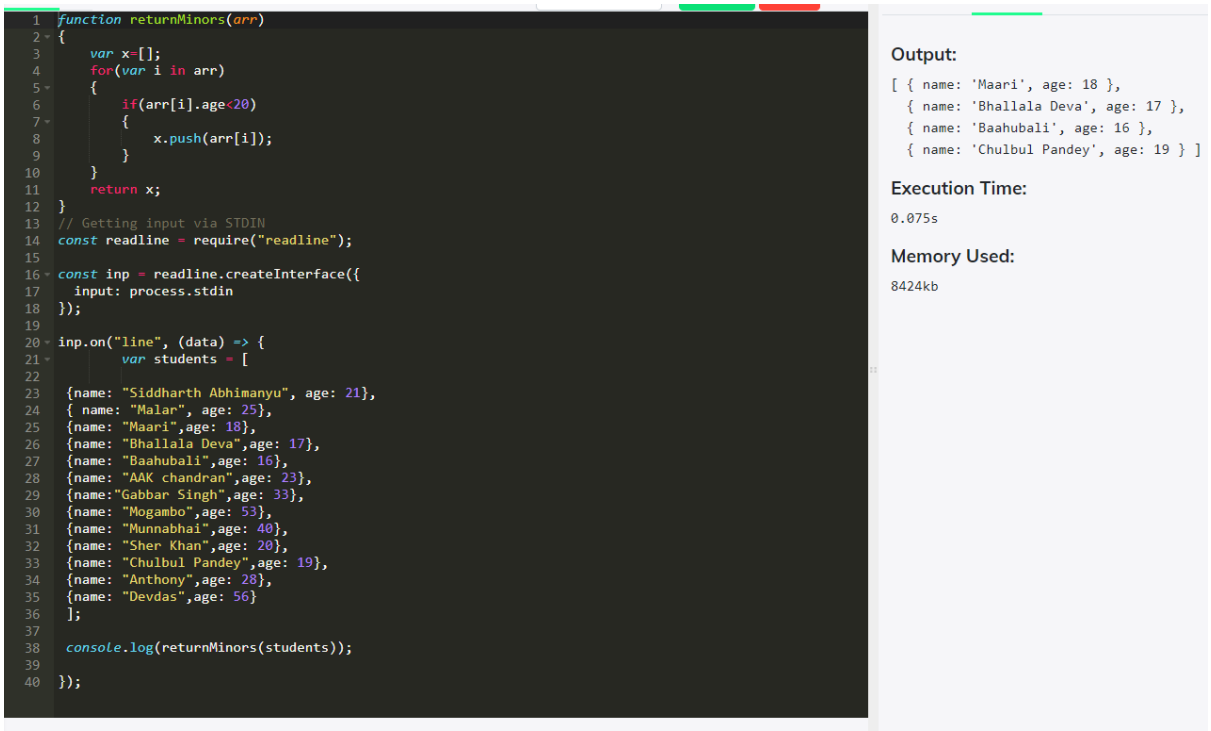
0.075s

Memory Used:

8544kb

Parsing JSON objects and Compare:

Write a function to return the list of characters below 20 age



```
function returnMinors(arr)
{
    var x=[];
    for(var i in arr)
    {
        if(arr[i].age<20)
        {
            x.push(arr[i]);
        }
    }
    return x;
}
// Getting input via STDIN
const readline = require("readline");

const inp = readline.createInterface({
  input: process.stdin
});

inp.on("line", (data) => {
        var students = [

 {name: "Siddharth Abhimanyu", age: 21},
 { name: "Malar", age: 25},
 {name: "Maari",age: 18},
 {name: "Bhallala Deva",age: 17},
 {name: "Baahubali",age: 16},
 {name: "AAK chandran",age: 23},
 {name:"Gabbar Singh",age: 33},
 {name: "Mogambo",age: 53},
 {name: "Munnabhai",age: 40},
 {name: "Sher Khan",age: 20},
 {name: "Chulbul Pandey",age: 19},
 {name: "Anthony",age: 28},
 {name: "Devdas",age: 56}
 ];

 console.log(returnMinors(students));

});
```

Output:

```
[ { name: 'Maari', age: 18 },
  { name: 'Bhallala Deva', age: 17 },
  { name: 'Baahubali', age: 16 },
  { name: 'Chulbul Pandey', age: 19 } ]
```

Execution Time:

0.075s

Memory Used:

8424kb