


Objects and Object methods

SOME UNIVERSITY

Student Registration Form

Name:	<input type="text"/>
DOB:	<input type="text" value="dd/mm/yyyy"/> 
Sex:	<input type="radio"/> Male <input type="radio"/> Female
Email:	<input type="text"/>
Address:	<input type="text"/>
School:	<input type="text" value="Select.."/>
Department:	<input type="text" value="v"/>
Course:	<input type="text" value="v"/>
Mobile Number:	<input type="text"/>
<input type="button" value="Reset"/>	<input type="button" value="Submit"/>

some university motto n' stuff.

Array vs Objects(Key-Value Pairs)

Array

```
var subjects = ["maths", "scienc", "english", "Hindi"];  
var marks = [40, 50, 80, 20];
```

here, I have two arrays one is containing the subjects and the other contains the marks of that respective subject.

- Suppose If I want to find the marks in English, Then I need to search first in the subjects array for finding the subject index and then using that index I can directly access the marks in the marks array.
- To access the information, the process is complex.

Objects

- It is a data structure that stores the data in a key-value manner.
- It is similar to any other forms which we had filled in our daily life, one side which is known as a key, which is telling that what information you want to store and right side acts as a value representing the value of that information.

Storing Information in Arrays vs Objects

Code 1 : Declaring Arrays vs Objects

```
// Arrays
var user1 = ["Rahul", 25, "male", "Bangalore", "coding"];

// Objects
var user2 = {
  name : "Rahul",
  age  : 25,
  gender: "male",
  city  : "Bangalore",
  hobbies: "coding"
};
console.log(user2);
```

Note : Key should be unique.

Accessing information in Arrays vs Objects

Code 2 : Accessing the information gender in arrays vs objects

```
// Arrays
var user1 = ["Rahul", 25, "male", "Bangalore", "coding"];
console.log(user1[2]);

// Objects
var user2 = {
  name : "Rahul",
  age  : 25,
  gender: "male",
  city  : "Bangalore",
  hobbies: "coding",
  marks : [25, 100, 80, 90, 80]
};

// 1. Bracket Notation
console.log(user["gender"]);
console.log(user["marks"]);
console.log(user["marks"][2]);
console.log(user["marks"].length);

// 2. Dot Notation
console.log(user.gender);
console.log(user.marks);
console.log(user.marks[2]);
console.log(user.marks.length);
```

In Objects, we can access the information by two ways

1. **Bracket Notation :**

For Ex : object["key"]

2. **Dot Notation**

For Ex : object.key

Adding information in Objects

- There are two ways to add information to an object
 - Bracket Notation : **object['key'] = value**
 - Dot Notation: **object.key = value**

Code 3: Add the date of birth field in the given object.

```
// Objects
var user2 = {
  name : "Rahul",
  age  : 25,
  gender: "male",
  city  : "Bangalore",
  hobbies: "coding",
  marks : [25, 100, 80, 90, 80]
};

// Ist Way
user2['Date_of_Birth'] = "02-Oct-1984";

// IInd Way
user2.Date_of_Birth = "02-Oct-1984";

console.log(user2);
```

Updating Information in Objects

How do we update an element in an array?

```
var arr = [1, 2, 3, 4]
arr[1]=5
console.log(arr)
We've updated the array right.
```

How do we update an object now?

Let's modify the age in the details object.
details["age"] = 19
details.city = "Hyderabad"
console.log(details) will give updated object.

Consider the string, "Masai" and I want to print no of times each letter is present in the string, i.e.

I want an output like {m:1, a:2, s:1, i:1}

```
var str = 'masai'

var diary = {} ==> Initialising an object

for(let i = 0 ; i < str.length; i++) {
  var char = str[i]
  diary[char] = 1
}
console.log(diary)
Will this code yield the correct result?
```

The key 'a' will get replaced right. Then how do we achieve our result.

```
var str = 'masai'

var diary = {} ==> Initialising an object

for(let i = 0 ; i < str.length; i++) {
  var char = str[i]
  if(diary[char] == undefined) {
    diary[char] = 1
  }
  else {
    diary[char] = diary[char] + 1
  }
}
```

```
}  
}  
console.log(diary)
```

Delete Information in Objects

- to delete information use keyword **delete**

```
delete object['key'];
```

```
delete object.key;
```

```
// Objects  
var user2 = {  
    name : "Rahul",  
    age  : 25,  
    gender: "male",  
    city  : "Bangalore",  
    hobbies: "coding",  
    marks : [25, 100, 80, 90, 80]  
};  
  
// Ist way  
delete user2["gender"];  
  
// IInd way  
delete user2["gender"]  
  
console.log(user2);
```

Object inside Object

- We can also store objects inside objects. Suppose I want to add information i.e Address and Address will contain other subfields i.e State, Country, District, Pincode, etc.

- To access the information, we can use either bracket or dot notation.

```
// Objects
var user2 = {
  name : "Rahul",
  age  : 25,
  gender: "male",
  city  : "Bangalore",
  hobbies: "coding",
  marks : [25, 100, 80, 90, 80],

  address : {
    state : "Uttarakhand",
    country : "india",
    district : "Dehradun",
    pincode : "249201"
  }
};

//Bracket Notation
console.log(user["address"]);
console.log(user["address"]["country"]);
console.log(user["address"]["pincode"]);

// Dot Notation
console.log(user.address);
console.log(user.address.country);
console.log(user.address.pincode);
```

Loops in Objects

- We have a special loop to iterate in objects.
- This special loop is known as, **for-in** loop.

Consider the details object only.

```
var details = {  
    name : "Cherry",  
    age : 19,  
    gender: "Male",  
    city : "Bangalore",  
    hobbies: ["Singing", "Dancing"]  
};  
  
for(var key in details) {  
    console.log(key, ' ', details[key])  
}
```

Array of Objects

Suppose I want to store multiple students information

```
var classroom = ["Babu Rao", "Munna bhai", "popat lal"]  
console.log(classroom);
```

We have stored multiple students name in an array, but what if there are multiple information I want to store with respect to that student like age also.

Then the array solution will not work here.

- The solution for this is **Array of Objects, Using** Array of objects it is possible to store the information.
- Using array index, we can directly fetch the student information and also we can fetch the particular attribute value of an object

Problem 4

```
var classroom = [  
    { name : "Babu Rao", age: 40 },
```



```

    { name : "Munna bhai", age : 30 },
    { name : "popat lal", age : 35 }
];

console.log(classroom[0].age);
console.log(classroom[0]["age"]);
console.log(classroom[1].name);

```

Loops with Array of Objects

Problem 5

```

var classroom = [
    { name : "Babu Rao", age: 40 },
    { name : "Munna bhai", age : 30 },
    { name : "popat lal", age : 35 }
];

// console.log(classroom[0].name);
// console.log(classroom[1].name);
// console.log(classroom[2].name);

for(var i = 0; i<classroom.length; i++)
{
    console.log(classroom[i].name);
}

```

Problem 6

Given amazon data, We need to find the following :

1. Find All Products Name and rating
2. Find Average Rating
3. Find the product having minimum price

```

var amazon = [
    {name:"Speakers", price:5400, rating:4},
    {name:"headphones", price:3000, rating:3},
    {name:"playstation",price:20000, rating:5}
];

var x = {name:"mixer",price:2000, rating:4}
amazon.push(x);

// Part 1 : Find All Products Name and rating
for(var i=0; i<amazon.length; i++){
    console.log(amazon[i].name, amazon[i].rating);
}

// Part 2 : Find Average Rating
var sum=0;
for(var i=0; i<amazon.length; i++){
    sum = sum + amazon[i].rating;
}
console.log(sum/amazon.length);

// Part 3 : Find the product having minimum price
var pad_min=amazon[0].price;
var bag = "";
for(var i=0; i<amazon.length; i++)
{
    if(pad_min > amazon[i].price)
    {
        pad_min = amazon[i].price;
        bag = amazon[i].name;
    }
}

```

```
console.log(pad_min, bag);
```

Problem 7

Given amazon data, We need to find the following :

1. Print those products name : whose rating is 4
2. Print All those products name, whose prices are between 2500 and 10000 including both

```
var amazon = [
    {name:"Speakers", price:5400, rating:4},
    {name:"headphones", price:3000, rating:3},
    {name:"playstation", price:20000, rating:5},
    {name:"mixer", price:2000, rating:4}
];

// Part 1 : Print those products name : whose rating is 4
for(var i=0; i<amazon.length; i++)
{
    if(amazon[i].rating == 4)
    {
        console.log(amazon[i].name);
    }
}

// Part 2 : Print All products name, whose prices are between 2500 and 10000
for(var i=0; i<amazon.length; i++)
{
    if(amazon[i].price >=2500 && amazon[i].price<=10000)
    {
```

```
        console.log(amazon[i].name, amazon[i].price);
    }
}
```

Problem 8 : Storing Student Information with Marks in different exams.

```
var data =
{
    name : "Rajesh",
    grade : "X",
    section: "A",
    marks : [
        {maths:30, science:40, english:50},
        {maths:60, science:70, english:80}
    ],
    hobbies : ["Dancing", "Singing"]
};

console.log(data.marks[0].maths);
```

To access the value of maths, we need to start from outside i.e data and then marks then subject

eg : To access maths marks

```
data.marks[0].maths
```

Problem 9: Map Charcters

```
mapChar(5);
```

```

function mapChar(N) {
  let obj = {}

  let alphabets = "abcdefghijklmnopqrstuvwxyz"

  for(let i = 0 ; i < N ; i++){
    let n = alphabets[i]

    obj[n] = i+1
  }

  // console.log(obj)

  for(let key in obj){
    console.log(key + "-" + obj[key] )
  }
}

```

Object methods,

Objects are usually created to represent entities of the real world, like users, orders and so on:

```

var user = {
  name: "John",
  age: 30
};

```

And, in the real world, a user can *act*: select something from the shopping cart, login, logout etc.

Actions are represented in JavaScript by functions in properties.

Method examples

For a start, let's teach the `user` to say hello:

```
var user = {  
  name: "John",  
  age: 30  
};  
  
user.sayHi = function() {  
  alert("Hello!");  
};  
  
user.sayHi(); // Hello!
```

Here we've just used a Function Expression to create a function and assign it to the property `user.sayHi` of the object.

Then we can call it as `user.sayHi()`. The user can now speak!

A function that is a property of an object is called its *method*.

So, here we've got a method `sayHi` of the object `user`.

Of course, we could use a pre-declared function as a method, like this:

```
var user = {  
  // ...  
};  
  
// first, declare  
function sayHi() {  
  alert("Hello!");  
}  
  
// then add as a method  
user.sayHi = sayHi;  
  
user.sayHi(); // Hello!
```

Problem 9 : Print Hello World using a Object Method print

```
var details = {  
  name : "Shubham",
```

```
    age : 31,

    print : function(){
        console.log("Hello world");
    }
};

console.log(details.name);
details.print();
```

Problem 10 :

```
var details = {
    name : "Shubham",
    age : 31,

    print : function(x){
        var y = x+2;
        return y;
    }
};

console.log(details.print(4));
```

Problem 11 :

```
var details = {
    name : "Shubham",
    age : 31,

    print : function(name){
        console.log("Hello", this.age, this.name);
    }
};
```

```

    }
};

var name = "Shyam";
details.print(name);

```

Problem 12 :

```

var details = {
  name : "Rajpal",
  age : 1000,
  hobbies : ["coding", "music"],

  print : function()
  {
    console.log("Hello", this.name);
    console.log("My age is", this.age);
    console.log("Hobbies are", this.hobbies.join(" "));
  }
}

details.print();

```

Problem 13 : Create two methods for the rectangle object

1. **perimeter**
2. **area**

```

var rectangle = {
  length : 30,
  breadth : 40,

```



```

    perimeter : function()
    {
        console.log(2*(this.length+this.breadth));
    },

    area : function()
    {
        console.log(this.length*this.breadth);
    },

};

```

```

rectangle.perimeter();
rectangle.area();
rectangle.length =10;
rectangle.breadth = 30;
rectangle.perimeter();
rectangle.area();

```

Problem 14 : Calculate the total price with the given quantity and price

```

var e_commerce = {
    products : ["earphone", "headphone", "earpods"],
    quantity : [4, 3, 2],
    price : [700, 800, 1000],

    total_price : function(){
        var sum=0;
        for(var i = 0; i<this.quantity.length; i++)
        {
            sum = sum + (this.quantity[i] * this.price[i]);
        }
    }
}

```

```

        return sum;
    }

};

console.log(e_commerce.total_price());

```

Problem 15 : Convert to Array of Objects

```

var products = ["macbook", "iphone", "ipad"];
var prices = [500000, 40000, 50000];

var data = [];
for(var i=0; i<products.length; i++)
{

    var obj = {};
    // obj["name"] = products[i];
    // obj["price"] = prices[i];
    obj.name = products[i];
    obj.price = prices[i];

    data.push(obj);
}

console.log(data);

```

IW Assignment

Problem 1 : Create an object with the following functionality

- **Ability to add student details and 3 subject marks**

```
//Create an object with the following functionality
//Ability to add student details and 3 subject marks

var details = {

  data : [],

  addStudent : function(nam, math, sci, eng)
  {
    var obj = {};
    obj.name = nam;
    obj.maths = math;
    obj.science = sci;
    obj.english = eng;

    this.data.push(obj);
  }
};

details.addStudent("Shubham", 10, 20, 30);
details.addStudent("Rahul", 15, 25, 35);

console.log(details.data);
```

Problem 2 :

- **Create a object method to find low score student**

```
var details = {

  data : [],
```

```

addStudent : function(nam, math, sci, eng)
{
    var obj = {};
    obj.name = nam;
    obj.maths = math;
    obj.science = sci;
    obj.english = eng;

    this.data.push(obj);
},

lowScore : function(){

    var lowStudent = null;
    var lowTotal = null;

    for(var i=0; i<this.data.length; i++)
    {
        var currStudent = this.data[i];
        var total = currStudent.maths + currStudent.science;

        if(lowTotal == null || total<lowTotal)
        {
            lowTotal = total;
            lowStudent = currStudent;
        }
    }

    return lowStudent;

}

};

```

```
details.addStudent("Shubham", 10, 20, 30);
details.addStudent("Rahul", 15, 25, 35);

// console.log(details.data);
console.log(details.lowScore());
```

Object methods,

Objects are usually created to represent entities of the real world, like users, orders and so on:

```
var user = {
  name: "John",
  age: 30
};
```

And, in the real world, a user can *act*: select something from the shopping cart, login, logout etc.

Actions are represented in JavaScript by functions in properties.

Method examples

For a start, let's teach the `user` to say hello:

```
var user = {
  name: "John",
  age: 30
};

user.sayHi = function() {
  alert("Hello!");
};

user.sayHi(); // Hello!
```

Here we've just used a Function Expression to create a function and assign it to the property `user.sayHi` of the object.

Then we can call it as `user.sayHi()`. The user can now speak!

A function that is a property of an object is called its *method*.

So, here we've got a method `sayHi` of the object `user`.

Of course, we could use a pre-declared function as a method, like this:

```
var user = {  
  // ...  
};  
  
// first, declare  
function sayHi() {  
  alert("Hello!");  
}  
  
// then add as a method  
user.sayHi = sayHi;  
  
user.sayHi(); // Hello!
```

Problem 9 : Print Hello World using a Object Method print

```
var details = {  
  name : "Shubham",  
  age : 31,  
  
  print : function(){  
    console.log("Hello world");  
  }  
};  
  
console.log(details.name);  
details.print();
```

Problem 10 :

```
var details = {  
  name : "Shubham",
```

```
    age : 31,

    print : function(x){
        var y = x+2;
        return y;
    }
};

console.log(details.print(4));
```

Problem 11 :

```
var details = {
    name : "Shubham",
    age : 31,

    print : function(name){
        console.log("Hello", this.age, this.name);
    }
};

var name = "Shyam";
details.print(name);
```

Problem 12 :

```
var details = {
    name : "Rajpal",
    age : 1000,
    hobbies : ["coding", "music"],

    print : function()
```

```

        {
            console.log("Hello", this.name);
            console.log("My age is", this.age);
            console.log("Hobbies are", this.hobbies.join(" "));
        }
    }

    details.print();

```

Problem 13 : Create two methods for the rectangle object

1. **perimeter**
2. **area**

```

var rectangle = {
    length : 30,
    breadth : 40,

    perimeter : function()
    {
        console.log(2*(this.length+this.breadth));
    },

    area : function()
    {
        console.log(this.length*this.breadth);
    },

};

rectangle.perimeter();
rectangle.area();
rectangle.length =10;

```



```
rectangle.breadth = 30;  
rectangle.perimeter();  
rectangle.area();
```

Problem 14 : Calculate the total price with the given quantity and price

```
var e_commerce = {  
  products : ["earphone", "headphone", "earpods"],  
  quantity : [4, 3, 2],  
  price : [700, 800, 1000],  
  
  total_price : function(){  
    var sum=0;  
    for(var i = 0; i<this.quantity.length; i++)  
    {  
      sum = sum + (this.quantity[i] * this.price[i]);  
    }  
    return sum;  
  }  
};  
  
console.log(e_commerce.total_price());
```

Problem 15 : Convert to Array of Objects

```
var products = ["macbook", "iphone", "ipad"];  
var prices = [500000, 40000, 50000];  
  
var data = [];  
for(var i=0; i<products.length; i++)
```

```

{

    var obj = {};
    // obj["name"] = products[i];
    // obj["price"] = prices[i];
    obj.name = products[i];
    obj.price = prices[i];

    data.push(obj);
}

console.log(data);

```

IW Assignment

Problem 1 : Create an object with the following functionality

- Ability to add student details and 3 subject marks

```

//Create an object with the following functionality
//Ability to add student details and 3 subject marks

var details = {

    data : [],

    addStudent : function(nam, math, sci, eng)
    {
        var obj = {};
        obj.name = nam;
        obj.maths = math;
        obj.science = sci;
    }
}

```

```

        obj.english = eng;

        this.data.push(obj);
    }
};

details.addStudent("Shubham", 10, 20, 30);
details.addStudent("Rahul", 15, 25, 35);

console.log(details.data);

```

Problem 2 :

- Create a object method to find low score student

```

var details = {

    data : [],

    addStudent : function(nam, math, sci, eng)
    {
        var obj = {};
        obj.name = nam;
        obj.maths = math;
        obj.science = sci;
        obj.english = eng;

        this.data.push(obj);
    },

    lowScore : function(){

        var lowStudent = null;
        var lowTotal = null;
    }
};

```

```

        for(var i=0; i<this.data.length; i++)
        {
            var currStudent = this.data[i];
            var total = currStudent.maths + currStudent.science;

            if(lowTotal == null || total<lowTotal)
            {
                lowTotal = total;
                lowStudent = currStudent;
            }

        }

        return lowStudent;

    }

};

details.addStudent("Shubham", 10, 20, 30);
details.addStudent("Rahul", 15, 25, 35);

// console.log(details.data);
console.log(details.lowScore());

```