13. Javascript Objects, Classes & JSON

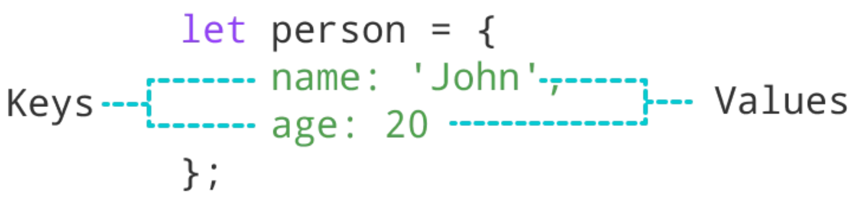
Table of Content

1. Javascript Objects

JavaScript object is a non-primitive data type that allows you to store multiple collections of data. If you are familiar with other programming languages, JavaScript objects are a bit different. You do not need to create classes in order to create objects.

```
// object
const student = {
    firstName: 'ram',
    class: 10
};
```

In JavaScript, "key: value" pairs are called properties. For example,



```
let person = {
    name: 'John',
    age: 20
};
```

You can also create values with nested objects, arrays, functions, etc.

▼ Accessing Object Properties

1. **Using dot Notation**

```
const person = {
    name: 'John',
    age: 20,
};

// accessing property
console.log(person.name); // John
```

2. **Using bracket Notation**

```
const person = {
    name: 'John',
    age: 20,
};

// accessing property
console.log(person["name"]); // John
```

▼ for…in loop with Object

```
const student = {
    name: 'Monica',
    grade: 7,
    age: 12
}

// using for...in
for ( let key in student ) {

    // display the properties
    console.log(`${key} => ${student[key]}`);
}
```

▼ Object Destructuring

The destructuring assignment introduced in ES6 makes it easy to assign array values and object properties to distinct variables. For example,

```
// assigning object attributes to variables
const person = {
    name: 'Sara',
    age: 25,
    gender: 'female'
}

// destructuring assignment
let { name, age, gender } = person;

console.log(name); // Sara
console.log(age); // 25
console.log(gender); // female
```

▼ Javascript Object Methods

You can create functions as values inside an object.

```
// object containing method
const person = {
    name: 'John',
    greet: function() { console.log('hello'); }
};
```

Javascript `this` keyword

To access a property of an object from within a method of the same object, you need to use the `this` keyword. Let's consider an example.

```
const person = {
    name: 'John',
    age: 30,

    // accessing name property by using this.name
    greet: function() { console.log('The name is' + ' ' + this.name); }
};

person.greet();
```

2. Javascript Classes

Classes are one of the features introduced in the ES6 version of JavaScript. A class is a blueprint for the object. You can create an object from the class.

```
// creating a class
class Person {
  constructor(name) {
    this.name = name;
  }
}

// creating an object
const person1 = new Person('John');
const person2 = new Person('Jack');
```

▼ Javascript Class Methods

```
class Person {
    constructor(name) {
      this.name = name;
    }

    // defining method
    greet() {
        console.log(`Hello ${this.name}`);
    }
}

let person1 = new Person('John');

// accessing property
console.log(person1.name); // John

// accessing method
person1.greet(); // Hello John
```

Note: JavaScript class is a special type of function. And the `typeof` operator returns `function` for a class.

For example,

```
class Person {}
console.log(typeof Person); // function
```

3. JSON

JSON stands for Javascript Object Notation. JSON is a text-based data format that is used to store and transfer data. For example,

```
// JSON syntax
{
    "name": "John",
    "age": 22,
    "gender": "male",

}
```

In JSON, the data are in **key/value** pairs separated by a comma `,`.

JSON was derived from JavaScript. So, the JSON syntax resembles JavaScript object literal syntax.

JSON is the most commonly used format for transmitting data (data interchange) from a server to a client and vice-versa. JSON data are very easy to parse and use. It is fast to access and manipulate JSON data as they only contain texts.

JSON is language independent. You can create and use JSON in other programming languages too.

JSON	JavaScript Object
The key in key/value pair should be in double quotes.	The key in key/value pair can be without double quotes.
JSON cannot contain functions.	JavaScript objects can contain functions.
JSON can be created and used by other programming languages.	JavaScript objects can only be used in JavaScript.

▼ Converting JSON to Javascript Object

You can convert JSON data to a JavaScript object using the built-in `JSON.parse()` function. For example,

```
// json object
const jsonData = '{ "name": "John", "age": 22 }';

// converting to JavaScript object
const obj = JSON.parse(jsonData);
```

```
// accessing the data
console.log(obj.name); // John
```

▼ Converting JavaScript Object to JSON

You can also convert JavaScript objects to JSON format using the JavaScript built-in `JSON.stringify()` function. For example,

```
// JavaScript object
const jsonData = { "name": "John", "age": 22 };

// converting to JSON
const obj = JSON.stringify(jsonData);

// accessing the data
console.log(obj); // "{"name":"John","age":22}"
```

Assignments

1. Create a Class called Vehicle with the properties of a Vehicle like `wheels, isDeisel` and functions like `drive(), break()` etc.
   a. Make multiple objects of these class with different properties.
   b. Iterate through the object using the for loops
   c. Convert the objects to JSON Strings.