Objects

Objects

Objects → Represent Properties, Methods[Functionalities]

Properties → Attribute: Color, weight etc.

Method → Actions, functions

Object is a way to store data in memory in `Key Value` Pair form.

They can hold properties (variables) and methods (functions).

Creation

1. **Object Literal**

```
let ChaiRecipe = {
  name: "Masala Chai",
  ingredients: {
    teaLeaves: "Assam Tea",
    milk: "Full Cream Milk",
    sugar: "Brown sugar",
    spices: ["Daalchini", "Ginger"]
  },
  instruction: "Boil water, add tea leaves, milk, sugar and spices"
};

console.log(ChaiRecipe.ingredients.spices); // [ 'Daalchini', 'Ginger' ]
console.log(ChaiRecipe.ingredients.spices[0]); // Daalchini
console.log(ChaiRecipe.instruction); // Boil water, add tea leaves, milk, sugar and
console.log(ChaiRecipe.ingredients.milk); // Full Cream Milk
```

2. Using `new Object()`

```
const car = new Object();
car.brand = "Toyota";
car.model = "Camry";
console.log(car.brand); // Output: Toyota
```

3. Using a Constructor Function

```
function Person(name, age) {
  this.name = name;
  this.age = age;
}
const user = new Person("Raj", 30);
console.log(user.name); // Output: Raj
```

```
// Constructor function for a Person
function Person(name) {
  this.name = name;
}

// Adding a method to the prototype
Person.prototype.greet = function () {
  console.log(`Hello, my name is ${this.name}`);
};

// Creating instances
const john = new Person("John");
const jane = new Person("Jane");

// Calling methods
john.greet(); // Output: Hello, my name is John
jane.greet(); // Output: Hello, my name is Jane
```

4. Using ES6 Classes

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

  // Method inside the class
  greet() {
    console.log(`Hello, my name is ${this.name}`);
  }
}

// Creating instances
const john = new Person("John");
const jane = new Person("Jane");

// Calling methods
john.greet(); // Output: Hello, my name is John
jane.greet(); // Output: Hello, my name is Jane
```

Accessing

1. Dot Notation

```
const person = {
  name: "Amit",
  age: 25,
  greet: function () {
    console.log(`Hello, my name is ${this.name}`);
  }
};

console.log(person.name); // Amit
console.log(person.age); // 25
person.greet(); // Hello, my name is Amit
```

2. Bracket Notation

```
const person = {
  name: "Amit",
  age: 25,
  greet: function () {
    console.log(`Hello, my name is ${this.name}`);
  }
};

console.log(person["name"]); // Amit

person['age'] = 30;
console.log(person["age"]); // 30
```

Adding Properties in a Object

```
const person = { name: "John" };

// Adding a new property
person.age = 30;
person["city"] = "New York";

console.log(person); // Output: { name: "John", age: 30, city: "New York" }
```

Deleting Properties in a Object

You can delete properties using the `delete` keyword.

```
const person = { name: "John" };

// Adding a new property
```

```
person.age = 30;
person["city"] = "New York";

delete person.age;

console.log(person); // Output: { name: "John", city: "New York" }
```

Iterating Over Objects

1. Using `for...in` Loop

```
const population = {
  male: 4,
  female: 93,
  others: 10
};

// Iterate through the object
for (const key in population) {
  console.log(`${key}: ${population[key]}`);
}

// male: 4
// female: 93
// others: 10
```

JSON formatter:

https://jsonformatter.org/