JavaScript

1. Variables

In JavaScript, variables are containers for storing data values.

`var`, `let`, and `const`

- `var` is used to declare variables in JavaScript. It has function scope or global scope.

- `let` is also used to declare variables but it has block scope. It means the variable is only available within the block where it is defined.

- `const` is used to declare constants. The value of a constant cannot be changed once it is defined.

Key differences:

- `var` has function scope, while `let` and `const` have block scope.

- Variables declared with `var` can be re-declared and updated, while variables declared with `let` can be updated but not re-declared. Constants declared with `const` cannot be updated or re-declared.

```
var x = 10;
let y = 20;
const z = 30;

x = 15; // Valid
y = 25; // Valid
// z = 35; // Invalid, trying to reassign a constant

// Redeclaration examples
var x = 5; // Valid
// let y = 15; // Invalid, redeclaration not allowed
```

2. Control Structures

`if` Statement

The `if` statement is used to make decisions based on a condition.

```
let num = 10;
if (num > 0) {
  console.log("Positive number");
} else {
  console.log("Negative number");
```

```
}
```

`switch` Statement

The `switch` statement is used to perform different actions based on different conditions.

```
let day = "Monday";
switch (day) {
  case "Monday":
    console.log("It's Monday!");
    break;
  case "Tuesday":
    console.log("It's Tuesday!");
    break;
  default:
    console.log("It's neither Monday nor Tuesday.");
}
```

3. Loops

`for` Loop

The `for` loop is used to repeat a block of code a specific number of times.

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}
```

`while` Loop

The `while` loop is used to execute a block of code as long as the specified condition is true.

```
let i = 0;
while (i < 5) {
  console.log(i);
  i++;
}
```

`for...of` Loop

The `for...of` loop is used to iterate over iterable objects like arrays or strings.

```
let arr = [1, 2, 3];
for (let element of arr) {
  console.log(element);
}
```

`for...in` Loop

The `for...in` loop is used to iterate over the properties of an object.

```
let obj = { a: 1, b: 2, c: 3 };
for (let key in obj) {
  console.log(key + ": " + obj[key]);
}
```

`forEach` Method

The `forEach` method is used to execute a provided function once for each array element.

```
let arr = [1, 2, 3];
arr.forEach(function (element) {
  console.log(element);
});
```

4. Functions

Functions are reusable blocks of code that perform a specific task.

Regular Functions

```
function greet(name) {
  console.log("Hello, " + name + "!");
}

greet("Alice"); // Output: Hello, Alice!
```

Arrow Functions

Arrow functions are a concise way to write functions in JavaScript.

```
const greet = (name) => {
  console.log("Hello, " + name + "!");
};

greet("Bob"); // Output: Hello, Bob!
```

Key difference between regular functions and arrow functions:

- Arrow functions do not have their own `this` keyword. They inherit `this` from the enclosing lexical context.

5. Classes and JSON

Classes

Classes are a template for creating objects with methods and properties.

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

  greet() {
    console.log(
      "Hello, my name is " + this.name + " and I am " + this.age + " years
old."
    );
  }
}

let person1 = new Person("Alice", 30);
person1.greet(); // Output: Hello, my name is Alice and I am 30 years old.
```

JSON

JSON (JavaScript Object Notation) is a lightweight data interchange format.

```
let person = {
  name: "Bob",
  age: 25,
  city: "New York",
};

console.log(person.name); // Output: Bob
```

6. Prototyping

Prototyping in JavaScript allows objects to inherit properties and methods from other objects.

```
function Animal(name) {
  this.name = name;
}

Animal.prototype.speak = function () {
  console.log(this.name + " makes a sound.");
};
```

```
let dog = new Animal("Dog");
dog.speak(); // Output: Dog makes a sound.
```

7. Document Object and Window Object

Document Object

The Document Object represents the HTML document loaded in the browser window. It provides various properties and methods for manipulating the document's content.

```
// Example: Changing the title of the document
document.title = "New Title";
```

Window Object

The Window Object represents the browser window that contains the document. It provides properties and methods for manipulating the browser window.

```
// Example: Opening a new window
window.open("https://www.example.com");
```

8. Web APIs

JavaScript interacts with the browser through Web APIs. Some common Web APIs include:

- DOM Manipulation API

- Fetch API

- XMLHttpRequest (XHR) API

- Geolocation API

- Local Storage API

- Canvas API

A. DOM Manipulation API:

This API allows JavaScript to interact with the HTML document, enabling tasks such as modifying the content, structure, and styles of a webpage.

```
<!-- HTML -->
<div id="example">Hello, World!</div>

// JavaScript
const element = document.getElementById('example');
element.style.color = 'blue';
```

B. Fetch API:

The Fetch API allows making HTTP requests to servers and handling responses asynchronously.

```
fetch("https://api.example.com/data")
  .then((response) => response.json())
  .then((data) => console.log(data))
  .catch((error) => console.error("Error fetching data:", error));
```

C. XMLHttpRequest (XHR) API:

XMLHttpRequest enables communication between a web browser and a server. It's commonly used for AJAX requests.

```
const xhr = new XMLHttpRequest();
xhr.open("GET", "https://api.example.com/data", true);
xhr.onreadystatechange = function () {
  if (xhr.readyState === 4 && xhr.status === 200) {
    console.log(xhr.responseText);
  }
};
xhr.send();
```

D. Geolocation API:

This API allows retrieving the geographical position of a device.

```
if ("geolocation" in navigator) {
  navigator.geolocation.getCurrentPosition((position) => {
    console.log("Latitude:", position.coords.latitude);
    console.log("Longitude:", position.coords.longitude);
  });
} else {
  console.log("Geolocation is not supported.");
}
```

E. Local Storage API:

Local Storage enables storing key-value pairs in a web browser.

```
localStorage.setItem('username', 'John');
const username = localStorage.getItem('username');
console.log('Username:', username);
```

F. Canvas API:

The Canvas API allows drawing graphics, text, and images on a webpage dynamically.

```
<canvas id="myCanvas" width="200" height="100"></canvas>;

const canvas = document.getElementById('myCanvas');
const ctx = canvas.getContext('2d');
ctx.fillStyle = 'green';
ctx.fillRect(10, 10, 150, 80);
```

9. Exercises

1. Write a JavaScript program to find the largest of three numbers using `if` statement.

2. Create a function that takes a number as input and returns "Positive" if the number is greater than zero, "Negative" if it's less than zero, and "Zero" if it's equal to zero.

3. Implement a JavaScript program using a `switch` statement to print the day of the week based on a numeric input (1 for Monday, 2 for Tuesday, etc.).

4. Write a `for` loop to print all even numbers between 1 and 20.

5. Use a `while` loop to find the factorial of a given number.

6. Create an array of numbers and use a `for...of` loop to calculate their sum.

7. Define an object representing a person with properties like name, age, and city. Use a `for...in` loop to print all the properties of the object.

8. Write a function that takes an array of numbers as input and returns the sum of all the numbers using the `forEach` method.

9. Implement a function that calculates the square of a given number using a regular function declaration.

10. Rewrite the square function from exercise 9 using an arrow function.

11. Create a class representing a car with properties like make, model, and year. Write a method in the class to display all the details of the car.

12. Use JSON to represent information about a book (title, author, year) and log the title of the book to the console.

13. Define a constructor function for a person object with properties like name and age. Add a method to the prototype to greet the person.

14. Write a JavaScript program to change the background color of a webpage when a button is clicked using the Document Object.

15. Use the Window Object to open a new browser window when a link is clicked.

These exercises cover various concepts such as conditionals, loops, functions, objects, classes, and interaction with the Document Object Model (DOM) and browser window. They provide hands-on practice to reinforce the concepts learned in the tutorial.

10.Utils

https://javascript.info/

https://developer.mozilla.org/en-US/docs/Web/javascript

https://www.w3schools.com/js/DEFAULT.asp

11.Ussage in HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
  scale=1.0">
    <title>Document</title>

</head>
<body>
    <script>
        let a = 10;
        console.log(a);
    </script>

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