script.js

```javascript
/*

Question 1: What are conditional statements? Explain conditional statements with syntax and examples.

Ans: Conditional statements are programming constructs that allow you to make decisions in your code based on certain conditions. These statements execute different blocks of code depending on whether a given condition is true or false.

Types of Conditional Statements

1. if Statement: The if statement is the simplest form of a conditional statement. It executes a block of code only if the specified condition is true.

2. if else Statement: The if else statement provides an alternative block of code to execute if the condition is false.

3. else if Ladder: The else if ladder statement allows you to specify multiple alternative blocks of code to execute if the first condition is false.

Syntax as well as examples are shown below:

*/

// if statement

let dividend = 10;

let divisor = 5;

if (divisor != 0) {

    let result = dividend / divisor;

    console.log("The result is: " + result);
}

// if else statement

```

```javascript
let age = 16;

if (age >= 18) {

    console.log("You are eligible to vote.");
}

else {

    console.log("You are not eligible to vote.");
}

// else if ladder

let score = 85;

if (score >= 90) {

    console.log("Grade A");
}

else if (score >= 80) {

    console.log("Grade B");
}

else if (score >= 70) {

    console.log("Grade C");
}

else {

    console.log("Grade D");
}
```

script.js

```javascript
// Question 2: Write a program that grades students based on their marks

let marks = 85;

if (marks > 90) {

    console.log("A Grade");
}

else if (marks >= 70 && marks <= 90) {

    console.log("B Grade");
}

else if (marks >= 50 && marks <= 70) {

    console.log("C Grade");
}

else {

    console.log("F Grade");
}
```

script.js

```
/*

Question 3: What are loops, and why do we need them? Explain different types of loops with their syntax and examples.

Ans: Loops are programming constructs that allow you to execute a block of code repeatedly until a certain condition is met. They are useful for tasks that require repetition, such as iterating over a collection of data.

Different types of loops include:

1. For Loop

2. While Loop

3. Do-While Loop

4. For-in Loop

5. For-of Loop

Sytax and Examples:

*/

// For Loop

console.log("For Loop:");

for (let i = 0; i < 5; i++) {

    console.log(i);
}

// While Loop

console.log("While Loop: ");

```

```
let i = 5;

while (i < 10) {

    console.log(i);

    i++;
}

// Do-While Loop

console.log("Do-While Loop: ");

let j = 15;

do {

    console.log(j);

    j++;

} while (j < 20);

// For-in Loop

console.log("For-in Loop: ");

const person = {name: "John", age: 30, city: "New York"};

for (let key in person) {

    console.log(key + ": " + person[key]);
}

// For-of Loop

console.log("For-of Loop: ");

```

```
let numbers = [2, 4, 6, 8, 10];

for (let num of numbers) {

    console.log(num);
}
```

script.js

```
/*

Question 4: Generate numbers between any 2 given numbers.

Example:

const num1 = 10;
const num2 = 25;

Output: 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

*/

const num1 = 10;

const num2 = 25;

for(let i = num1; i <= num2; i++){

    console.log(i);
}
```

script.js

```javascript
/*

Question 5:  Use the while loop to print numbers from 1 to 25 in ascending and descending order.

*/

console.log("Ascending Order:");

let i = 1;

while (i <= 25) {

    console.log(i);

    i++;
}

console.log("Descending Order:");

let j = 25;

while (j >= 1) {

    console.log(j);

    j--;
}
```