NAME - SHREYA RAKSHIT

COLLEGE - IIT (ISM) DHANBAD

DOMAIN - FULL STACK WEB DEVELOPMENT

TASK NO. - 2

PROJECT NAME - INTRODUCTION TO WEB STANDARDS AND JAVASCRIPT CONCEPTS AND FUNDAMENTALS

TASK NAME - CONTROL STRUCTURES IN JAVASCRIPT

IMPORTANCE OF CONTROL STRUCTURES IN JAVASCRIPT

CONTROL STRUCTURES:

1. if, else and else if:

   if is used to specify a block of code to be executed if a specific condition is met.

   Syntax:

   ```
   if (condition) {
     //  block of code to be executed if the condition is true
   }
   ```

   else is used to specify a block of code to be executed if the condition is false.

   Syntax:

   ```
   if (condition) {
     //  block of code to be executed if the condition is true
   } else {
     //  block of code to be executed if the condition is false
   }
   ```

   else if is used to specify a new condition if the first condition is false.

   Syntax:

   ```
   if (condition1) {
     //  block of code to be executed if condition1 is true
   } else if (condition2) {
     //  block of code to be executed if the condition1 is false and condition2 is true
   } else {
   ```

```
  //  block of code to be executed if the condition1 is false and condition2 is false
}
```

2. switch statement:
   switch statement is used to select a block of code to be executed depending on the value of a given expression.
   Syntax:
   ```
   switch(expression) {
     case x:
       // code block
       break;
     case y:
       // code block
       break;
     default:
       // code block
   }
   ```
   The value of the expression is evaluated once and compared with all the test cases. If there is a match, that case will be executed and if there are multiple matches, the first one will be executed. In case no case matches, the default code block will be executed.
   break keyword is used to break the execution of the switch block. If it is omitted, the next case block will be executed even if there is no match.
   Sometimes, multiple switch cases may share the same code block.
   Switch cases use strict comparison (===) that means it will be true only if the operands are of the same data type and same value.

LOOPS:

Loops are very useful to execute repetitive tasks.

1. For Loop:
   It creates loops through a block of code a number of times.
   Syntax:
   ```
   for (expression 1; expression 2; expression 3) {
     // code block to be executed
   }
   ```
   Expression 1 is executed once before the execution of the code block. It is mainly used to initialize variables (one or many) used in the loop. If it is not required, it can be omitted.
   The expression 2 gives a condition to be checked before executing the loop every time. If it is true, the loop will be executed again, otherwise not. It is also optional.
   Expression 3 increments the value of the initial variable after running through the loop each time. This is also optional.
2. For In Loop:
   The JavaScript for in statement loops through the properties of an Object or the properties of an array.
   Syntax:

```
for (key in object) {
  // code block to be executed
}

for (variable in array) {
  code
}
```

It should not be used if the index order is important. a for loop, a for of loop, or Array.forEach() should be used in this case.
The forEach() method calls a function (a callback function) once for each array element. The function takes 3 arguments: The item value, The item index, The array itself.

3. For Of Loop:
   The JavaScript for of statement loops through the values of an iterable object. It is used ro loop over iterable data structures such as Arrays, Strings, Maps, NodeLists, and more.
   Syntax:
   ```
   for (variable of iterable) {
     // code block to be executed
   }
   ```
   variable - For every iteration the value of the next property is assigned to the variable. Variables can be declared with const, let, or var.
   iterable - An object that has iterable properties.
   It is supported in Chrome 38, Edge 12, Firefox 51, Safari 7, Opera 25 and not supported in Internet Explorer.
4. While Loop:
   The while loop loops through a block of code as long as a specified condition is true.
   Syntax:
   ```
   while (condition) {
     // code block to be executed
   }
   ```
   If the variable used in the condition is not updated after each iteration, It will turn into an infinite loop.
5. Do While Loop:
   It is a variant of the while loop. This loop executes the code block once, before checking if the condition is true, then it repeats the loop as long as the condition is true.
   Syntax:
   ```
   do {
     // code block to be executed
   }
   while (condition);
   ```
   If the variable used in the condition is not updated after each iteration, It will turn into an infinite loop.

REFERENCES:

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