```
<script>

    // operators
    10 + 3
    10 - 3
    10 * 3
    10 / 3
    10 ** 3
    10 % 3

    10 + 3 * 2
    (10 + 3) * 2


    // variables
    let x = 10 + 3;
    let y = 20;
    x + y;
    x - y;
    x % y;


    // comparision operators
    10 > 3
    10 < 3

    10 === 10
    10 === 3
    10 !== 3

    // logical operators
    console.log(null || 10);
    console.log(20 || 10);
    console.log(20 && 10)
    console.log(null && 10)
    console.log(20 && null)

    // typeof operator
    typeof x;
    typeof y;

    //
```

```
43
44      // types
45      // string, numbers, booleans, undefined, null, bigint, symbol
46      // Literals are constants, and increment/decrement would try to change its argument respectively. But constant
values cannot be changed.
47      let z = "ahmad";
48
49
50
51
52
53      // comments:
54      // single line: explain a peice of code
55
56      /*
57          multi-line commments
58          - comment out code
59      */
60
61
62
63
64
65
66
67
68  </script>
```

```html
<script>
    // alert is a built-in function.
    // argument "Hello world" is passed.
    // statement ends with semicolon
    alert("Hello world");
</script>
```

```
<script>
    // Syntax error: grammatical error
    // semicolon is automatically inserted at the end.
    alert("Hello world);
</script>
```

```
<script>
    let name = prompt("What is your name?");
    alert("hello, " + name);
</script>
```

```
<script>
    let x = Number(prompt("What's x?"));
    let y = Number(prompt("what's y?"));

    if(x < y){
        alert("x is less than y");
    }
</script>
```

```
<script>
    let x = Number(prompt("What's x?"));
    let y = Number(prompt("what's y?"));

    // all potential outcomes are now accounted for.
    if(x < y){
        alert("x is less than y");
    }
    else if(x > y){
        alert("y is less than x");
    }
    else {
        alert("x is equal to y");
    }
</script>
```

```
<script>
    let c = prompt("do you agree?");

    if(c === 'Y' || c === 'y'){
        alert("Agreed");
    }
    else if(c === 'N' || c === 'n') {
        alert("Not Agreed");
    }


</script>
```

```html
<script>
    alert("salam");
    alert("salam");
    alert("salam");
</script>
```

```html
<script>
    // how to tell computer to repeat something.
    let i = 0;
    while(i < 3){
        alert("salam");
        i++;
    }
</script>
```

```html
<script>
    // how to tell computer to repeat something.
    for(let i = 0; i < 3; i++){
        alert("salam");
    }
</script>
```

```html
<script>

    // infinite loop
    while(true){
        alert("salam");
    }

</script>
```

```
<script>
    // how to tell computer to repeat something.
    let i = 3;
    while(i > 0){
        alert("salam");
        i--;
    }
</script>
```

```
<script>
    // array to store list of items
    const scores = [70, 60, 90];

    scores[0];
    scores[1] = 50;

    scores.length;
    scores.sort(); // sorts array in place (mutates it)

    // iterating over array
    for(let score of scores){
        console.log("score is " + score);
    }

    scores.push(85);


</script>
```

```
<script>

</script>
```

```
<script>
     // functions, return values, arguments
    // built-in functions
    // custom functions
    alert("hi");
    prompt("whta is your name");

    // create your own function

    function sum(x, y){
        return x + y;
    }

    sum(3,4);

    // function as value
    const subtract = function(x,y){
        return x - y;
    }

    // function combination
    subtract(sum(3,4), sum(1,2));


    // function as callback
    const greet = function(name, callback){
        console.log("Hi " + name);
        callback();
    }

    const askHow = function(){
        console.log("how are you?");
    }

    const welcome = function(){
        console.log("welcome!");
    }



</script>
```

```html
<script>
    // why we need object?
    /*
        Objects make it easier to understand how a program works by
        bringing together data and its behavior (or method) in a
        single bundle called an “object.”
    */

    // without object
    // both are related
    let name = "ahmad";
    let age = 20;

    // use object instead
    const person = {
        name: "ahmad",
        age: 20,
        introduceSelf: function(){
            console.log(`hi, I'm ${this.name}`);
        }
    }

    // constructor
    function Person(name, age){
        this.name = name;
        this.age = age;

        this.introduceSelf = function(){
            console.log(`hi, I'm ${this.name}`);
        }
    }

    // create objects
    const user1 = new Person("Ahmad", 10);
    const user2 = new Person("Ali", 25);


    user1.name;
    user1["age"];

    user1["age"] = 11;
    user1.fatherName = "Mahmood";
```

```
43
44
45
46
47      // for in loop: iterate over object
48      for(let key of user1) {
49          console.log(`${key} is ${user1[key]}`);
50      }
51
52
53
54
55
56
57      /*
58          built-in objects
59          - Array (push, pop, foreach, for-of)
60          - Date
61          - Math
62
63
64
65      */
66
67
68  </script>
```