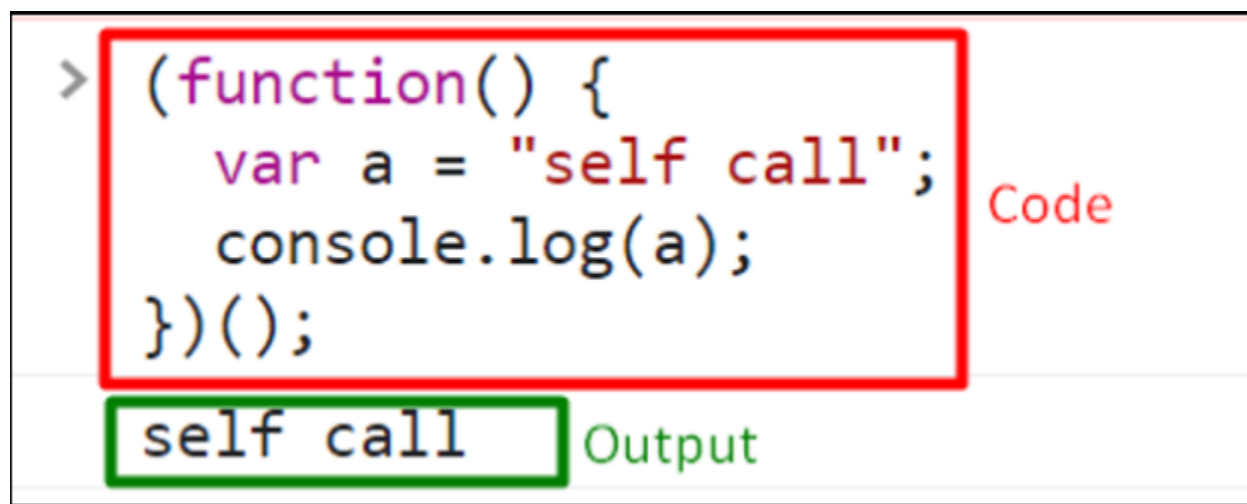# Interview questions 2

1.invoke function

The code inside a function is executed when the function is invoked

It is common to use the term "call a function" instead of "invoke a function".

```javascript
(function() {
    var a = "self call";
    console.log(a);
})();
```
Code

`self call` Output

2. Difference between break and continue

The `break` statement "jumps out" of a loop.

The `continue` statement "jumps over" one iteration in the loop.

3.Function Types

There are 3 ways of writing a function in JavaScript:

- Function Declaration

- Function Expression
- Arrow Function

```
// Function declaration

function add(a, b) {

        console.log(a + b);

}
```

```
// Calling a function

add(2, 3);
```

```
// Function Expression

const add = function(a, b) {

        console.log(a+b);

}
```

```
// Calling function

add(2, 3);
```

```
// Single line of code

let add = (a, b) => a + b;
```

```
console.log(add(3, 2));
```

4. String methods with example

1.charAt

const str = "Hello World";

// using charAt() method

console.log(str.charAt(1));

console.log(str.charAt(6));

2.charCodeAt

```
1    const str = "Hello World";
2
3    // using charCodeAt() method
4    console.log(str.charCodeAt(0));
5    console.log(str.charCodeAt(1));
```

▶ Try It   Run Here ›

72
101
>

3,concat

```
1    const firstName = "Hello";
2    const lastname = "World";
3
4    // using concat method
5    let result1 = firstName.concat(" ", lastname);
6    console.log(result1);
7    // multiple arguments
8    let result2 = "Learning".concat(" ", "to", " ", "code.");
9    console.log(result2);
```

▶ Try It   Run Here ❯

```
Hello World
Learning to code.
>
```

4.endsWith

```
1    const question = "What is DOM?";
2
3    // using endsWith method
4    // checks whether the string ends with "?"
5    console.log(question.endsWith("?"));
```

▶ Try It   Run Here ❯

```
true
>
```

5.includes

```
1   const sentence = "Carbon emission is increasing Day by day.";
2
3   // check if the string contains words
4   console.log(sentence.includes("day")); // true
5   console.log(sentence.includes("Day")); // true
6   console.log(sentence.includes("DAY")); // false
```

▶ Try It   Run Here ›

true
true
false

6.indexOf

```
1   const sentence = "Carbon emission is increasing day by day";
2
3   // get 1st index of 'day'
4   console.log(sentence.indexOf("day"));
5   console.log(sentence.indexOf("Carbon"));
6   // case-sensitive, returns -1
7   console.log(sentence.indexOf("carbon"));
```
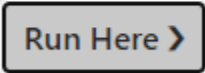
▶ Try It   Run Here ›

30
0
-1
>

7.replace

```
1   const str = "Carbon emission is increasing day by day.";
2
3   // using string to match
4   console.log(str.replace("day", "year"));
5   // using regular expression to match
6   console.log(str.replace(/day/g, "year"));
```

▶ Try It   Run Here ⟩

```
Carbon emission is increasing year by day.
Carbon emission is increasing year by year.
>
```

5.Difference between replace and replace all example

Replace :

The first parameter can be a string or a regular expression. If it is a string value, only the first instance of the value will be replaced.

```
const str = "JavaScript Courses";

const newStr = str.replace('JavaScript', 'Java');

console.log(newStr); // Java Courses
```

replaceAll();

This method replaces all appearances of the search string with the replacement text and returns a new string.

6.Ternary operator

A ternary operator evaluates a condition and executes a block of code based on the condition.

```
condition ? expression1 : expression2
```

// program to check pass or fail

let marks = prompt('Enter your marks :');

// check the condition
let result = (marks >= 40) ? 'pass' : 'fail';

console.log(`You ${result} the exam.`);

```
})();


// program to check pass or fail

let marks = prompt('Enter your marks :');

// check the condition
let result = (marks >= 40) ? 'pass' : 'fail';

console.log(`You ${result} the exam.`);
// program to check pass or fail

let marks = prompt('Enter your marks :');

// check the condition
let result = (marks >= 40) ? 'pass' : 'fail';

console.log(`You ${result} the exam.`);
```