Branching

In JavaScript, branching refers to the ability to execute different code paths based on certain conditions. This is typically done using control flow statements such as if, else, and switch.

The if statement is used to check a condition, and if it evaluates to true, the code within the corresponding block will be executed. The else statement can be used in conjunction with the if statement to provide an alternative code path if the initial condition is not met.

Here is an example of an if statement:

```
if (age >= 18) {
  console.log("You are old enough to drive.");
}
else{
  console.log("Sorry, you can't drive")
}
```

The switch statement is similar to multiple if statements and is used to check for multiple conditions, where each case represents a specific condition. Once a matching case is found, the corresponding code block will be executed.

And here is an example of a switch statement:

```
let color = "red";
switch (color) {
  case "red":
    console.log("The color is red.");
    break;
  case "blue":
    console.log("The color is blue.");
    break;
  default:
    console.log("The color is neither red nor blue.");
}
```

Looping

JavaScript has several ways to loop through code. The most commonly used are:

1. for loop: This loop is used to iterate over a sequence of items, such as an array or a string. It has three parts: the initialization, the condition, and the increment/decrement.

```
for (let i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}
```

2. for...of loop: This loop is used to iterate over the values of an iterable object, such as an array, a string, or a Map. It does not work with plain objects.

```
for (const value of array) {
  console.log(value);
}
```

3. for...in loop: This loop is used to iterate over the properties of an object. It is not recommended to use arrays, because it will iterate over the properties of the array object, not the elements of the array.

```
for (const key in object1) {
  console.log(object1[key]);
}
```

4. while loop: This loop is used to repeat a block of code while a certain condition is true.

```
let i = 0;
while (i < ar.length) {
  console.log(ar[i]);
  i++;
}
```

5. do...while loop: This loop is similar to the while loop, but it will execute the code block at least once, and then check the condition.

```
let i = 0;
do {
  console.log(arr[i]);
  i++;
} while (i < arr.length);
```

6. Array.prototype.forEach(): This method is used to execute a provided function once for each array element.

```
arr.forEach(function(el) {
  console.log(el);
});
```