1. The **for** loop:

   The **for** loop is a standard loop used in most programming languages. It has three parts: initialization, condition, and incrementation.

   The syntax of a **for** loop in JavaScript is as follows:

   ```
   for (initialization; condition; incrementation) {
     // code to be executed
   }
   ```

   For example, to print the numbers 0 to 4 using a **for** loop, we can use the following code:

   ```
   for (let i = 0; i < 5; i++) {
     console.log(i);
   }
   ```

   The output of this code will be:

   0

   1

   2

   3

   4

2. The **for...in** loop:

   The **for...in** loop is used to loop through the properties of an object. It iterates over the enumerable properties of an object in an arbitrary order, where the property names are assigned to the loop variable.

   The syntax of a **for...in** loop in JavaScript is as follows:

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

   For example, to loop through the properties of an object called **person**, we can use the following code:

   ```
   const person = {name: 'John', age: 30, gender: 'male'};
   for (let property in person) {
     console.log(property ,':', person[property]);
   ```

}

The output of this code will be:

name: John

age: 30

gender: male

3. The **for...of** loop:

   The **for...of** loop was introduced in ECMAScript 6 (ES6) and provides a cleaner syntax for iterating over an iterable object. It does not iterate over object properties like the **for...in** loop, but instead directly accesses the values of the iterable object.

   The syntax of a **for...of** loop in JavaScript is as follows:

   for (variable of iterable) {

   // code to be executed

   }

   For example, to print the elements of an array using a **for...of** loop, we can use the following code:

   const arr = [1, 2, 3, 4, 5];

   for (let num of arr) {

   console.log(num);

   }

   This will output:

   1

   2

   3

   4

   5

   Note that the **for...of** loop only works with iterable objects, so you cannot use it with plain objects or other non-iterable data types.