JavaScript Loops

A loop is a way to repeat actions in your code. Imagine you want to tell a computer to do something several times, like count numbers, repeat a song, or list favorite toys. Instead of writing the same instructions over and over, you can use a loop to make the computer repeat the actions for you.

Why Use Loops?

Loops make coding easier and your programs more efficient. Here’s why:

1. **Save Time**: Instead of writing the same code multiple times, you can write it once and let the loop handle the repetition.
2. **Reduce Errors**: Fewer chances of making mistakes since you’re not copying and pasting the same code.
3. **Make Changes Easily**: If you need to change the repeated action, you only change it in one place.

1. for Loop

The for loop is used to run a block of code a specific number of times.

```
/* FOR LOOP IN JAVASCRIPT*/

// how to write it?  for(initialize;condition;increment or decrement)

// for (let i = 0; i < 10; i=i+2) {
//   console.log(‘hello’);// hello will be printed 8 times
// }



/*using for loops in relation with array - we will start from the first item and move to the last*/

// console.log("here you are out of the loop");
// let names = ["kefe", "mohammed", "etenesh", "zebib", "mekbib", "fantahun"];
// for (let i = 0; i <= names.length; i++) {
//    console.log(names[i]);
// }
```

/*using for loops in relation with array - we will start from the last array and move to the first one backwards*/

```
// console.log("here you are out of the loop");

// let names = ["kefe", "mohammed", "etenesh", "zebib", "mekbib", "fantahun"];
// for (let i = names.length-1; i >= 0; i--) {
//   if (names[i] === "zebib") {
//     continue;
//   } else {
//     console.log(names[i]);
//   }
// }
```

```
// js code that print numbers from  0 to 4 using for loop
for (let i = 0; i < 5; i++) {
  console.log(i);  // Output: 0, 1, 2, 3, 4
}
```

NOTE: try to practice and do some exercises related to for loop; you have a question? ***You have a question? You know what to do ☺ reach the group & we will figure it out together as a team.***

2. while Loop

The while loop runs as long as a specified condition is true.

```
// how to write?  while(condition)
// let counter = 0;
// while (counter < 10) {
// //   console.log("now its: " + counter);
//   counter++;  // increment the counter for next iteration
// }
// console.log("here you are out of the while loop");
```

// using **break** and **continue** keywords to manage loop execution

```
// continue  go to the next iteration
// break     end the loop

/* WHILE LOOP IN JAVASCRIPT*/
// while(condition)
// let counter = 0;
// while (counter < 10) {
//   if (counter == 3) {
//     counter++;
//     continue; //jump 3 and move to next iteration
//   }
//  if(counter==5){
//break; //end the loop and be out of the loop
//}
//   console.log("now its: " + counter);
//   counter++;
// }
// console.log("here you are out of the while loop");
```

NOTE: try to practice and do some exercises related to while loop; you have a question?
You have a question? You know what to do ☺ reach the group & we will figure it out together as a team.

What is Object-Oriented Programming (OOP) and Why Use It?

Understanding OOP

Object-Oriented Programming, or OOP, is a way of writing computer programs using "objects." An object is like a real-life thing that has properties (characteristics) and behaviors (actions it can perform).

Real-Life Analogy

Imagine you have a toy car. This toy car has:

- **Properties**: color, size, and model.
- **Behaviors**: move forward, move backward, and honk.

In OOP, we create similar "objects" in our code that have properties and behaviors.

Why Use OOP?

1. **Organization**: OOP helps organize code by grouping related properties and behaviors into objects, making it easier to understand and manage.
2. **Reusability**: You can create objects that can be reused in different parts of your program, saving time and effort.
3. **Easier Maintenance**: If you need to change something, you can make updates in one place, and it will automatically apply to all instances of that object.

What is an Object in JavaScript?

An object in JavaScript is a collection of properties, where each property is an association between a key (or name) and a value. Objects can store data (properties) and methods (functions) that operate on that data.

Accessing properties and methods of an object in JavaScript can be done using dot notation or bracket notation. Better to use the dot notation

Dot Notation

Dot notation is the most common and straightforward way to access an object's properties and methods.

student.age // here we are using a dot notation (.) to access age property of the student

Creating Objects

Objects in JavaScript can be created using object literals or constructors.

Using Object Literal

```
/* USING OBJECT LITERALS  IN JS FOR OBJECT ORIENTED PROGRAMMING */
// let fantahun={
// fName:"kebede",
// lName:"Bishaw",
// age:26,
// fullName: function(){
//   return this.fName+' '+this.lName;
// }
// }
// let abebe={
//   fName:"Abebe",
//   lName:"Balcha",
//   age:56,
//   fullName: function(){
//     return this.fName+' '+this.lName;
//   }
//   }

// let fantahun={
//   fName:"kebede",
//   lName:"Bishaw",
//   age:26,
//   fullName: function(){
//     return this.fName+' '+this.lName;
//   }
//   }
// method : a function which is found inside an object
// console.log(fantahun.fullName());
// console.log(fantahun.fName+" " +fantahun.lName + " age: "+fantahun.age);
```

NOTE: try to practice and do some exercises related to objects in object literal case;
You have a question? You know what to do ☺ reach the group & we will figure it out together as a team.

Using Constructor Function

```javascript
// make the first leter of the function name capital
function EvangadiStudent(fname, lname, batch, age, height) {
  (this.firstName = fname),
    (this.lastName = lname),
    (this.batchNo = batch),
    (this.studentAge = age),
    (this.studentHeight = height),
    (this.printName = function () {
      return this.firstName + " " + this.lastName;
    });
}
//use the "new" keyword to create new object using a constructor
let fantahun = new EvangadiStudent("Fantahun", "Bishaw", "April 4", 26, 1.83);
let zebib = new EvangadiStudent("Zebib", "aaa", "April 4", 26, 1.8);
let kefe = new EvangadiStudent("Kefale", "bbb", "april 4", 26, 1.81);
console.log(fantahun.printName());
```

NOTE: try to practice and do some exercises related to objects using constructor case; ***You have a question? You know what to do ☺ reach the group & we will figure it out together as a team.***