***Loops:***

To do work repetitively.

Syntax:

For(initialize;condition;expression){

}

Examples:

for(var i=0;i<5;i++){

    console.log(i);     //0 1 2 3 4

}

for(var i=4;i<=8;i++){

    console.log(i);      //4 5 6 7 8

}

for(var i=10;i>=1;i--){

   console.log(i);   // 10 9 8 7 6 5 4 3 2 1

}

Graphical user interface, text, application

Description automatically generated with medium confidence

//printing table using for loop through user input :

var num=parseInt(prompt("enter a number for table ?"));

for(var i=1;i<=10;i++){

    console.log(num+" X "+i+" = "+(num\*i));

}

***Break and continue(looping):***

Needs some breaks into some condition while loop is in process.same continue while skipping some iterations and continue code body execution when condition met.

Graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

// break and continue

for(var i=0;i<7;i++){

    if(i==3)

    break;                      // 0 1 2

    console.log(i);

}

for(var i=0;i<7;i++){

    if(i==3)

    continue;                      // 0 1 2 4 5 6

    console.log(i);

}

// prime numbers

var num=parseInt(prompt("enter a number to check "));

var isprime=true;

for(var i=2;i<num;i++){

    result=num%i;

    if(result==0){

        console.log("this number : " + num +" is not prime ! "+ " at checked "+i);

isprime=false;

        break;

    }

}

if(isprime){

    console.log("this number : " + num +" is prime ! "+ " at checked "+i);

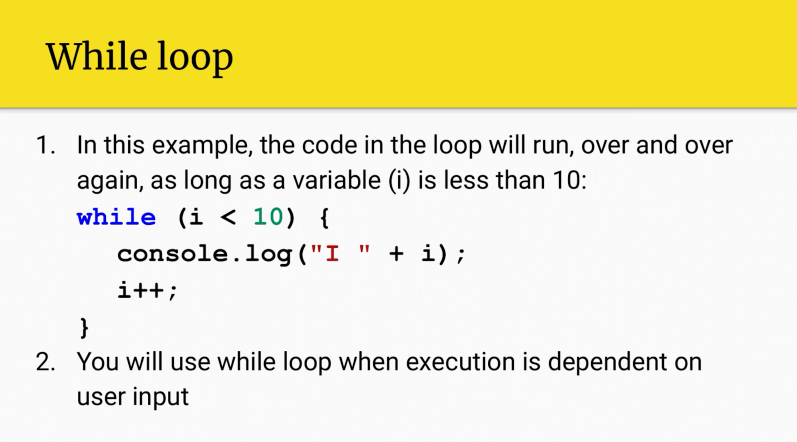
}

While

Do while

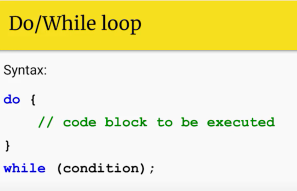
For each loop are more other types of loops we can use.

***While loop:***

This loop just needs condition , usually Boolean true or false. No initialization , expression etc. 

***Do While loop:***

Once a execution happen under do body.then while condition checks.



Graphical user interface, text, application, chat or text message

Description automatically generated

var i=0;

do{

    console.log(i);

    i++;

}while(i < 10);

Note needs to be initialized beforehand 0 is necessey otherwise nothing happened in looping.

var cond=true;

while(cond){

    console.log("hey there !");

    var asking=prompt("want to continue press Y or y to exit press other key...");

    if(asking.toLowerCase() !== "y"){

        cond=false;

    }

}

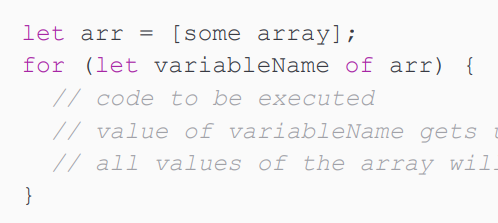
console.log("loop breaks game exit!");//:63 hey there !  loop breaks game exit!

***For of loop:***

It can be used to iterate over elements of an array. Using it we cant change index already associated with some element.

It lets you to iterate over arrays , strings , maps.

Syntax :



Example :

// for of loop

let arr=[1,2,4,6,7,8,9,2,3,44,44,6,3,1,6,7,9,56,33];

for(let array of arr ){

     console.log(array);

}

Example 02 :

const cars = ["BMW", "Volvo", "Mini"];

let text = "";

for (let x of cars) {

  text += x + "\t";

}

document.getElementById("demo").innerHTML=text;

***For in loop(looping over properties of objects):***

Looping over an object can be done in a few ways. We can use the for in loop to loop over the object directly, or we can convert the object to an array and loop over the array.

It loops through the properties of object.

Graphical user interface, text, application

Description automatically generated

Example :

// for in loop

let garih = {

    year : 2020,

    model : "A8",

    color : "black",

    engine : "V8"

};

for(let atr in garih){

    console.log(atr);

}

Do not use **for in** over an Array if the index **order** is important.

It is better to use a **for** loop, a **for of** loop, or **Array.forEach()** when the order is important

***For each loop:***

let arrried=["ahmed","ali","ansari"];

arrried.forEach(element => {

    console.log(element);

});

***nested loop:***

a loop inside another loop.

Syntax:

Text

Description automatically generated with medium confidence

Example :

for(var i=0;i<5;i++){

    for(var j=0;j<3;j++){

        console.log("I : "+i+" J : "+j);

    }

}

Output :

I : 0 J : 0

I : 0 J : 1

I : 0 J : 2

I : 1 J : 0

I : 1 J : 1

I : 1 J : 2

I : 2 J : 0

I : 2 J : 1

I : 2 J : 2

I : 3 J : 0

I : 3 J : 1

I : 3 J : 2

I : 4 J : 0

I : 4 J : 1

I : 4 J : 2