3 LESSON

SCSA JavaScript Courses

1)ARRAY

2) LOOP

3) OBJECTS

Array

Create variable?

we want to Create many data?

Create Array?

```
const name1="David";
const name2="George";
const name3="Anna";
```

```
const names=["David", "George", "Anna"];
console.log(names)
```

Array Two Types

```
const names=["David", "George", "Anna"];
console.log(names)
```

```
[ ‘David', 'George', 'Anna' ]
```

```
const names=["David", "George", "Anna"];
console.log(names[0])
```

```
David
```

```
const names= new Array(“David", "George", "Anna");
console.log(names)
```

```
[ ‘David', 'George', 'Anna' ]
```

Array Tasks

```
const calcAge= (Birthday) => 2020-Birthday;
const add_arr= new Array(1, 5 , 15);
console.log(calcAge(add_arr))
```

Output:

1) 2019 2) NaN 3) 2020 4) Underfined 5) 1999

Arrays can be created using the:

1) Array() Counstructor
2) [] literals
3) Both Of the above

Objects

Objects in programming are just like the objects in real life. They have properties and methods. Take the example of a television. Its properties include color, resolution, price etc. and methods include muting the sound, increasing the volume, changing the brightness etc. Properties are the description of an object and methods are functions relating to the object.

Objects

The most convenient way to create objects in JavaScript is using the object literals {}

```
var tv = {
   color: "black",
   res: 1080,
   price: 500,
   getPrice: function() { return this.price - 50; }
}
```

Going with out tv example, here we have created an object named tv with four properties color, res, price and getPrice. The property names are followed by a colon and then their respective values. Multiple properties are seperated using commas.

Getting Values:

Console.log(tv.color) *// "black"*

For calling methods just put **parentheses** at the end of the property. **Remember methods are also properties**!

Console.log(tv["getPrice"]()); *// 450 // or*
Console.log(tv.getPrice()); *// 450*

Add Property

```
// creating an object
var tv = {
   color: "black",
   res: 1080,
   price: 500,
   getPrice: function() { return this.price - 50; }
}

 // properties added
tv.size = 32;
tv["type"] = "LED";

// output values
console.log(tv.color); // "black"
console.log(tv.getPrice()); // 450
console.log(tv["size"]); // 32
```

Loop

```
console.log('Hi I am  1 year old ')
console.log('Hi I am  2 year old ')
console.log('Hi I am  3 year old ')
console.log('Hi I am  4 year old ')
console.log('Hi I am  5 year old ')
console.log('Hi I am  6 year old ')
console.log('Hi I am  7 year old ')
console.log('Hi I am  8 year old ')
console.log('Hi I am  9 year old ')
console.log('Hi I am  10 year old ')
```

```
for(let i=1; i<=10; i++){
    console.log(`Hi I am ${i} years old`)
}
```

Syntax Of the for Loop

var declarations is the statement declaring and/or initialising variables that, in loops, are usually called iterators, counters.

condition is the expression to be tested and checked for to execute the loop once more.

incrementing expression is the expression incrementing the counter(s).

```
for (/*var declarations*/ ; /*condition*/ ;
/*incrementing expression*/) {
    // code to be repeated
    // this enclosed block of code is also known as a
LOOP BLOCK
}
```

```
for(let i=1; i<=10; i++){
    console.log(`Hi I am ${i} years old`)
}
```

While Loop

```
for(let i=1; i<=10; i++){
    console.log(`Hi I am ${i} years old`)
}
```

```
let i=1;
while(i<10){
    console.log(`Hi I am ${i} years old`);
    i++;
}
```

Looping Array

In Array is items: 5,15,30, “mobile”, [“cars”, 180];

Loop this array and print;

Loop and print child array;

HomeWork

// შექმენით მასივი "arr";

// შექმენით ცარიელი ცვლადი;

// შექმენით ფუნქცია,მასივისთვის, რომელიც გამოანგარიშებს, თუ arr მნიშვნელობები მეტია 10-ზე. ისინი გადაამრავლოს 10-ზე . რომელი მნიშვნელობებიც ნაკლებია 10-ზე გაყოფ 0.5-ზე

// arr მნიშვნელობები: 5, 15, 20, 60, 88, 6,