TinkerHub
GEC IDUKKI

JS BOOTCAMP #DAY 2

Arrays

An array is a special variable, which can hold more than one value.

```
var a = ['geci', 'bootcamp', 'TinkerHub']
```

Printing elements in an array

```
var a = ['geci', 'bootcamp', 'TinkerHub'];

for( i = 0; i<3; i++)
  {
      console.log( a[i]);             // prints  gec, bootcamp, TinkerHub
  }
```

Sort( )

The `sort()` method sorts an array

```
var num = [ 12, 23, 34, 25, 46 ];
num.sort();
console.log(num);                    // prints [ 12, 23, 25, 34, 46 ]
```

Reverse( )

To print the array in descending order `reverse()` method is used

```
var num = [ 12, 23, 34, 25, 46 ];
num.reverse();
console.log(num);                         // prints [ 46, 34, 25, 23, 12 ]
```

Array containing strings can also be sorted alphabetically

```
 var a = ['geci', 'bootcamp', 'TinkerHub'];
 a.sort();
 console.log(a);              // prints [ ‘bootcamp’, ‘gec’, ‘TinkerHub’ ]
```

Accessing elements of a String

```
var name = ‘TinkerHub’;
console.log( name [0] )            // prints ‘T’
```

Finding Length of a String

```
var name = ‘TinkerHub’;
console.log( name.length )          // prints 9
```

Functions in JS

A JavaScript function is a block of code designed to perform a particular task.

A JavaScript function is executed when "something" invokes it (calls it).

```
function printName()
{
    console.log('Hey');
}

printName();
printName();
printName();


OUTPUT
Hey
Hey
Hey
```

Function with parameters

Function **parameters** are the **names** listed in the function definition

function functionName(parameter1 , parameter2)
{
  // code block
}

```
function printName( name )
{
   console.log( name );
 }

printName("TinkerHub");           //calling the function by passing arguments
```

Function with more than one parameter

```
function printName( name, num )
{
   console.log( name + num );
}
printName("TinkerHub", 2021);           // prints TinkerHub2021
```

Some Math functions

`Math.round(x)` : returns the nearest integer

```
a = Math.round(4.6);
console.log(a)                      // prints 5
```

`Math.random()` : returns a random number between 0 (inclusive), and 1 (exclusive):

```
a = Math.random();
console.log(a)                    // returns a random number like 0.60845371
```

```
var a = Math.random() * 10;

// Returns a random integer from 0 to 9
```

JS Dates

```
var a = Date()
console.log (a)                    // prints the current date
```

for of loop

The JavaScript `for of` statement loops through the values of an iterable object.

```
var colours = ['blue', "green", "yellow" ];
for ( name of colours )
```

```
{
   console.log(name)
}

OUTPUT
blue
green
yellow
```

Objects in JS

Objects are variables too. But objects can contain many values. The values are written as **name:value** pairs (name and value separated by a colon).

```
var student = { name:"Rahul", age:21, college:"GECI" }

console.log(student.name)
console.log(student.age)
console.log(student.college)

OUTPUT
Rahul
21
GECI
```

When array come as an object property

```
var student = {
    name : "Rahul" ,
    age :  21,
    arr : [2 ,4 ,5 ,7]
    }

console.log(student.name)              //prints Rahul
console.log(student.age)               //prints 21
console.log(student.arr[0])            //prints 2
```

Array of objects

```
var student = [
              {name : "Rahul" , age : 21, dept : "cs"},
              {name : "Sunil" , roll: 32}
              ]

console.log(student)


OUTPUT
[ { name: 'Rahul', age: 21, dept: 'cs' }, { name: 'Sunil', roll: 32 } ]
```

or

```
var student = {
              1:{name : "Rahul" , age :  21, dept : "cs"},
              2:{name : "Sunil" , roll: 32}
              }


console.log(student[1].name)           //prints Rahul
console.log(student[2].roll)           //prints 32
```

for in loop

The JavaScript `for in` statement loops through the properties of an Object.

```
var test = { fname : "gec" , lname :  "idukki" }

for (x in test )
{
    console.log( test[x] )
}

OUTPUT
gec
idukki
```

💡 *Here to print 'gec' (from code) we want to access it by* `test.fname`

return statement in JS

```
function test(num)
{
   var sum = num + num ;
   return sum;
}

var x = test(10)
console.log(x)                                  // prints 20
```

The `return sum` statement returns the value of the `sum` to the calling function and it gets passed to variable `x`

Happy Tinkering!!