Const: cannot be changed.

Camel cape doLikeThis

Hoisting: bring variable to top of the scope only, not outside

Hoisting work for var and function

Truthy and falsy is nature, can not be compared to other variable

**Break**: jump out of the loop;

**Return**: stop execution of a function and return last value of the function;

**Method**= function in the object

var today=new Date().getDay();

switch (today) {

case 6:

console.log("Today is Saturday");

break;

case 0:

console.log("Today is Sunday");

break;

default:

console.log("Looking forward to the Weekend");

}

Can have 2 cases for same action

switch (a) {

case 2:

case 10:

// some cool stuff

break;

case 42:

// other stuff

break;

default:

// fallback

}

* **Array** [lskdjf,s;dkljfs;aldk,sdfsadj;l] or = “a1, a2, a3”;

[] aray: collection of value [1, 2, 3, 4]

Index start from 0 1 2 3

Arraynam[index]; call element

Arraynam.length; length of array

Normally hold one type of data in 1 array

Array.shift(); remove the beginning and return old array

Array.pop(); remove the end and return old array

Array.push(); add items at the end and return new array

Array.unshift(); add items to the beginning and return new array

Array.slice(start,end); select element

*array*.splice(index, howmany items to remove, item1, ....., itemX) replace and add elements to an array

*array1*.concat(array2, array3, ..., arrayX) join arrays

*Determines whether the array contains a value*

var pets = ["cat", "dog", "bat"];

console.log(pets.includes("cat"));

*// expected output: true*

console.log(pets.includes("at"));

*// expected output: false*

*string*.split(separator, limit) string into array

let str = "How are you doing today?";  
const myArr = str.split(" ");

* **Object** {a: “slth”, b:543, c:true};

Inlcuding key values

Cannot have 2 key with 1 name

ObjectName.a; call value of a

ObjectName[“a”]; call value of a

Var test={

A= “text”

B=43

};

Var B=”A”

Test[B]; =>”text”

Test[“B”]; =>43

Console.log(objectNam.functionName);

Console.log(objectNam.functionName());

The Object.keys() method returns an array of strings containing all of the object's keys, sorted by order of appearance:

The Object.values() method returns an array of strings containing all of the object's field values, sorted by order of appearance

Can only get length of array, not object

{} object: pairs of value {a=42314, b=094832094}

Hasownproperty(); check if have property in object or not;

Objectnam.newKey= 'newValue'; add new property to object

delete person.age;  // or delete person["age"]; delete propterty

**STRING**

return "cat"[1]; *// returns "a"*

var sentence = "The quick brown fox jumps over the lazy dog.";

var word = "fox";

console.log(sentence.includes(word)); *// expected output: true*

var str1 = "Saturday night plans";

console.log(str1.startsWith("Sat")); *// expected output: true*

var str2 = "Is this a question";

console.log(str2.endsWith("?")); *// expected output: false*

* slice(start [, end]): Extracts a section of a string and returns a new string.

console.log(str.toLowerCase());

*// expected output: "the quick brown fox jumps over the lazy dog."*

console.log(str.toUpperCase());

*// expected output: "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG."*

const greeting = " Hello world! ";

console.log(greeting);

*// expected output: " Hello world! ";*

console.log(greeting.trim());

*// expected output: "Hello world!";*

**Nested for Loop**