**Naming Conventions**

## 1. Naming Convention for Variables

var dogName = 'Scooby-Doo';

## 2. Naming Convention for Booleans

var isBark = false;

var areIdeal = true;

var hasOwner = true;

## 3. Naming**Convention for Functions**

function getName(dogName, ownerName) {

return '${dogName} ${ownerName}';

}

## 4. Naming Convention for Constants

var LEG = 4; var TAIL = 1; //if one word

var MOVABLE = LEG + TAIL;

var DAYS\_UNTIL\_TOMORROW = 1; // if more than one word

## 5. Naming Convention for Classes

class DogCartoon {

constructor(dogName, ownerName) {

this.dogName = dogName;

this.ownerName = ownerName;

}

}

var cartoon = new DogCartoon('Scooby-Doo', 'Shaggy');

## 6. Naming Convention for Components

function DogCartoon(roles) {

return (

< div >

< span > Dog Name: { roles.dogName } < /span>

< span > Owner Name: { roles.ownerName } < /span>

< /div>

);

}

## 7. Naming Convention for Methods

class DogCartoon {

constructor(dogName, ownerName) {

this.dogName = dogName;

this.ownerName = ownerName;

}

getName() {

return '${this.dogName} ${this.ownerName}';

}

}

var cartoon= new DogCartoon('Scooby-Doo', 'Shaggy');

console.log(cartoon.getName());

// "Scooby-Doo Shaggy"

## 8. Naming Convention for Denoting Private Functions

class DogCartoon {

constructor(dogName, ownerName) {

this.dogName = dogName;

this.ownerName = ownerName;

this.name = \_toonName(dogName, ownerName);

}

\_toonName(dogName, ownerName) {

return `${dogName} ${ownerName}`;

}

}

var cartoon = new DodCartoon('Scooby-Doo', 'Shaggy');

// good

var name = cartoon.name;

console.log(name);

// "Scooby-Doo Shaggy"

## 9. Naming Convention for Global Variables

For global JavaScript variables, there are no specific naming standards.

It is recommended to use **camel case** for **mutable** global variables and **uppercase**for**immutable**global variables.

**Or \_baseUrl**

## 10. Naming Convention for File Names

So, it is recommended to use **lowercase** file names in all servers despite their case-sensitive support.