Classes

class Animal{

eat = () => console.log(‘I am eating’)

}

class Dog extends Animal{ // inheritate class

weight = 20;

age = 8;

bark = () => console.log(‘bark bark bark’)

}

puppy = new Dog()

puppy.bark();

puppy.eat()

Spread Operator

> pull value from an array, extract the properties of an object and can be used in a function

> [Syntax] : ...

-- Example 1--

oldArr = [1,3,5]

newArr = [...oldArr, 2,4]

console.log(newArr);

-- Output 1--

[1,3,5,2,4]

-- Example 2--

oldObj ={

name: ”Peter”,

height: 175

}

newObj = {

...oldObj, //want to add the value of oldObj into newObj

age : 18 // if this line is height:180, output is {name:”Peter”, height:180

}

console.log(newObj)

-- Output 2--

{

name: “Peter”

height : 180

}

-- Example 3--

const showValues = (...args) => {

console.log(...args)

for(i=0; i<args.length; i++){

console.log(args[i]);

}

}

showValue(“apple”, “orange”);

-- Output 3--

[“apple”, “orange”, “banana”]

apple

orange

Destructuring

> can be applied on both object and array.

-- Example 1--

var person = {

firstName: “Alan”,

lastName : “Smith”,

age: 50,

eyeColor: “blue”

};

const {firstName,age} = person;

console.log(firstName);

console.log(age);

-- Output 1--

Alan

50

-- Example 2--

[a,b] = [1,2]

console.log(a)

[a,b,...remain] = [1,2,3,4,5]

console.log(..remain);

[,,c] = [1,2,3]

console.log(c);

-- Output 2--

1

345

3

Primitive vs reference

Primitive Values

> data that are stored on the stack

> stored directly in the location that the variable access

> value is stored with assign by value

--Example 1--

var name = “Sam”;

var namecopy = name;

console.log(‘name’,name)

console.log(‘namecopy’, namecopy)

name= ”Mary”

console.log(‘name’,name)

console.log(‘namecopy’, namecopy)

-- Output 1--

name Sam

namecopy Sam

name Mary

namecopy Sam

References Values

> object that are stored in the heap

> stored in the variable location is a pointer to a location in memoru where the object is stored

> value is stored with assign by reference

--Example 1--

const user = {

name: “Mike”

}

const usercopy = user

console.log(‘user’, user);

console.log(‘usercopy’, usercopy);

user.name = “Mary”;

console.log(‘user’, user);

console.log(‘usercopy’, usercopy);

-- Output 1--

user{

name: “Mike”

}

usercopy{

name: “Mike”

}

user{

name: “Mary”

}

usercopy{

name: “Mary”

}

Import, Export

Export

> export default apple

> export const banana = “banana”

Import

> import apple from ”./directory”

> import {banana} from ”./directory”

Array

-- Example 1--

var numbers = [1,2,3]

const plus3 = numbers.map(num => num + 3)

console.log(plus3);

const plus4function = num => num+4

const plus4 = numbers.map(plus4function)

console.log(plus4)

-- Output 1--

[4,5,6]

[5,6,7]

For in \ For of

> Array is a group of objects that why it is “For of”

> cos loop inside a fruitobject thats why use “For in”

-- Example 1--

const fruitsArray = [“apple”, “orange”, “banana”]

for (let value of fruitArray){

console.log(value); // print out one by one

}

constfruitsObject = {

apple:13,

orange: 17,

banana:15

}

for (let key in fruitsObject){

console.log(fruitsObject[key]);

}

-- Output 1--

apple

orange

banana

13

17

15