Cú pháp es6

Cú pháp hàm:

console.log("Hello C07");

// declare function (hàm thường dùng)

function tinhTong(a,b) {

 return a+b;

}

console.log(tinhTong(10,20));

// express function (hàm thường nhưng gán số)

const sum = function(a,b){

}

//arrow function

//(giá trị đối số 1,2)=>{thao tác trong hàm , phải có return}

const sum = (a,b)=>{

    let c = a+b;

    return c;

}

console.log(sum(11,22));

//trường hợp đặc biết của arrow function với 1 tham số

const showMessage = mess =>{

    console.log(mess);

}

showMessage("hello codegym")

//trương hợp code chỉ có 1 lệnh return

const sum = (a,b)=>a+b;

let array = [10,3,4,7,9];

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

    console.log(`${i}: ${array[i]}`);

}

//foreach để in

// chỉ là ví dụ về callback ở hàm chứ k phải in

function display (e,i){

    console.log(`${i}: ${e}`);

}

//display gọi là callback function

//truyền một function vào cho một function ;

// array kia là mảng gọi đến hàm forEach để duyệt với hàm forEach thì trong hàm có 1 arrow function để console log ra giá trị

//trong forEach nhận e làm giá trị của mảng và i là index

array.forEach((e,i)=>{

    console.log(`${i}: ${e}`);

});

//hàm map: thao tác từng phần tử trên mảng, cũng nhận tham số truyền vào là e,i

* Cú pháp: (tên mảng).map((e,i)=>{ return hành động với e});
* Và sẽ thường gán vào 1 mảng mới

let newArray = array.map((e,i)=>{

    return ++e;

});

/

//ví dụ về mảng sinh viên( gồm các đối tượng trong {} dùng I id: 1, (dấu phẩy nếu còn thuộc tính) và gán giá trị vào thuộc tính bằng :

        name: "chánh1"

const students = [

    {

        id: 1,

        name: "chánh1"

    },

    {

        id: 2,

        name: "chánh2"

    },

    {

        id: 3,

        name: "chánh3"

    }

]

// dùng hàm map và in ra một hang các thuộc tính của từng ptu (khác ở return)

const listStudent = students.map((e,i)=>{

    return (`<tr>

                 <td>${e.id}</td>

                  <td>${e.name}</td>

            </tr>`)

});

//in ra mảng mới có giá trị mỗi ptu là 1 hàng

console.log(listStudent);

//hàm filter: lọc mảng và gán vào mảng mới theo điều kiện

//cú pháp: const <tên mảng đã lọc> =<mảng cần lọc>.filter((e,i)=>điều kiện lọc);

const filterStudent = students.filter((e,i)=>e.id>=2);

console.log(filterStudent);

///default param : giá mặc định của tham số

// gán giá trị cho tham số  -> là default param

const sum=(a =1,b=2)=>a+b;

// nếu truyền vào giá trị thì tham số sẽ dc thay đổi khỏi giá trị mặc định

console.log(sum(10,20));

console.log(sum(5));

// Destructuring: phân rã mảng /đối tượng để lấy ra phân tử hoặc thuộc tính dễ dàng hơn

let mang1 =[23,2,5,7,34,6];

let mang2  = [2,4,6,8];

//...mang => phân rã mảng ra

//mang => là một đối tượng

let mang3 = [mang1,mang2]

//[ [ 23, 2, 5, 7, 34, 6 ], [ 2, 4, 6, 8 ] ] => 2 đối tượng -> như 2 phần tử của mảng mới

let mang3 = [...mang1,mang2]

//[ 23, 2, 5, 7, 34, 6, [ 2, 4, 6, 8 ] ] -> 1 mảng gộp vào và 1 đối tượng

let mang3 = [...mang1,...mang2]

// [

//   23, 2, 5, 7, 34,

//   6, 2, 4, 6,  8

// ] => gộp hẳn 2 mảng vào mảng

console.log(mang3)

//muốn lấy 2 phần tử đầu tiên của mảng

let a = mang[0];

let b = mang[1];

//cho a,b là 2 ptu đầu tiên và ...rest chính là mảng để lưu những giá trị còn lại trong mang1 => nó chính là 1 mảng động

const [a,b,...rest] = mang1; // cú pháp lưu giá trị trực tiếp của mảng const [biến hoặc mảng: ...rest]

// console.log(a)

// console.log(b)

// console.log(rest);

let student = {

    id: 1,

    name: 'chánh1'

}

//lưu giá trị trực tiếp của đối tượng const {thuộc tính chính xác của đối tượng} = đối tượng dc khởi tạo

const {name,id} = student;

// rest param

console.log(name)

console.log(id)

//gán trực tiếp có:

//mảng: const [a,b,...rest]=mang

//đối tượng: const {thuộc tính }= đối tượng

// có thể truy xuât trực tiếp

//tính tổng một mảng cần có giá trị truyền vào là 1 mảng(linh hoạt):

const sum = (...rest)=>{

    let tong =0;

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

        tong += rest[i];

    }

    return tong;

}

console.log(sum(3,4,6,7,8,9,1,3));

// toán tử spread ;

// thêm thuộc tính cho đối tượng(gán lại đối tượng)

student ={

    //gọi các thuộc tính ban đầu của đối tượng trc

    ...student,

    //thêm thuộc tính mới

    age:40

}

console.log(student);

Chú ý truy xuất trong đối tượng (thuộc tính có mảng, thuộc tính có đối tượng)

const person ={

firstName: 'John',

lastName: 'Doe',

age: 30,

gender: 'male',

occupation: 'developer',

nationality: 'America',

city: 'New York',

hobbies: ['reading', 'traveling', 'photography'],

**languages: ['English', 'spanish'],**

**education: {**

**degree: 'Bachelor',**

**major: 'Computer Science',**

**university: 'Harvard University'**

**}**

};

**//truy xuất gán đối tượng**

**const {firstName,gender,education: {degree},languages: [English]}=person;**

**//thuộc tính có mảng:**

**languages: [English]**

**//thuộc tính có đối tượng**

**education: {degree}**