Javascript

1 – Biến

// Khai báo 2 biến x, y nhận giá trị số bất kì

// Khai báo các biến khác và lần lượt gán giá trị tổng, hiệu, tích, thương của x, y cho từng biến.

2 – Kiểu dữ liệu

/\*\*

\* Sử dụng các kiến thức đã học trong bài khai báo các biến mô tả một người, vật nào đó mà bạn thích

\* Ví dụ dưới đây mô tả thời tiết ngày hôm nay

\* var today = 'Sunday';

\* var isCloudy = true;

\* var currentTemperature = 30;

\*/

3 – Object

/\*\*

\* Sử dụng kiến thức đã học khai báo object person gồm các property name, age, gender.

\* Ví dụ:

\* var person = {

\* name: 'Pham tan phat',

\* age: 25,

\* gender: 'male'

\* };

\*/

4 – Arraylist

/\*

Khai báo một array "listAnimal" chứa các object mô tả danh sách các con vật.

Mỗi con vật có 2 thuộc tính: name và age trong đó name có kiểu string, age có kiểu number.

\* {name}: string;

\* {age}: number;

\*/

5 – Arithmetic operator

Ví dụ :

var a = 10;

var h = 20;

/\*\*

\* a: cạnh đáy của 1 hình tam giác

\* h: chiều cao của tam giác đó

\* Yêu cầu: khai báo biến s bằng diện tích của hình tam giác này

\*/

Ví dụ :

var width = 10;

var height = 20;

var s = width \* height;

/\*\*

\* width: Chiều rộng của 1 hình chữ nhật

\* height: Chiều cao của hình chữ nhật

\* s: diện tích của hình chữ nhật (code mẫu)

\* Yêu cầu:

\* - Khai báo biến p là chu vi của hình chữ nhật này.

\*/

Ví dụ :

var r = 10;

var s = r \* r \* 3.14;

/\*\*

\* r: Bán kính của 1 hình tròn

\* s: Diện tích của hình tròn đó

\* Yêu cầu:

\* - Khai báo biến p là chu vi của hình tròn này.

\*/

6 – Phép tính tăng giảm

Ví dụ:

var a = 10;

var x = --a + a++;

console.log(x);

// Nhấn Run để chạy code.

// Giải thích vì sao kết quả là 18.

Ví dụ:

var a = 10;

var b = 18;

var x = a-- + b++ - ++b - ++a;

console.log(x);

// Nhấn Run để chạy code

// Giải thích tại sao kết quả lại như vậy. Gợi ý: Viết ra từng dòng để giải thích

7 – Phép gán

var a = 10;

/\*\*

\* Viết ra tất cả các cách khác nhau có thể (trong phạm vi bài học) để gấp đôi a

\*/

8 – Function

// Bài tập: Viết hàm tính thể tích khối hộp.

// Nhận vào các tham số như dưới, trả về thể tích của khối hộp.

/\*\*

\* Tính thể tích khối hộp

\* https://www.wikihow.com/Calculate-Volume-of-a-Box

\* @param {Number} w Chiều rộng

\* @param {Number} h Chiều cao

\* @param {Number} l Chiều dài

\* @returns {Number} Diện tích

\*/

function getBoxVolume(w, h, l) {

// Viết code ở đây

}

console.log(getBoxVolume(10, 20, 30)); // 6000

console.log(getBoxVolume(4, 5, 6)); // 120

9 – Object method

/\*\*

\* Khai báo biến `rectangle` là 1 object gồm có các property:

\* - width: chiều rộng, là 1 số bất kì

\* - height: chiều cao, là 1 số bất kì

\* Viết các method:

\* - getWidth: trả về chiều rộng

\* - getHeight: trả về chiều cao

\* - getArea: trả về diện tích

\*/

10 – Các phép so sánh

Ví dụ:

var a = [1, 2];

var b = [1, 2];

console.log(a === b);

// Chạy chương trình và tìm hiểu vì sao kết quả lại là false mặc dù giá trị của 2 array có vẻ giống nhau. Viết câu trả lời ở dưới đây.

Ví dụ:

var a = {

foo: 'bar'

};

var b = {

foo: 'bar'

};

console.log(a === b);

// Chạy chương trình và tìm hiểu vì sao kết quả lại là false mặc dù giá trị của 2 object có vẻ giống nhau. Viết câu trả lời ở dưới đây.

Ví dụ :

var a = '1000';

var b = '200';

console.log(a > b);

// Chạy chương trình và tìm hiểu vì sao kết quả lại là false mặc dù 1000 trông có vẻ lớn hơn 200. Viết câu trả lời ở dưới đây.

11 – Vòng lặp

Ví dụ

var a = [1, 2, 4, 8, 16];

/\*\*

\* Sử dụng vòng lặp for để hiển thị ra màn hình các phần tử của mảng a theo thứ tự ngược lại:

\* 16

\* 8

\* 4

\* 2

\* 1

\*/

Ví dụ

var a = [1, 2, 3, 4, 5];

var b = [10, 20, 30];

/\*\*

\* Sử dụng vòng lặp for để tính tích của các phần tử của array a với các phần tử của array b Kết quả mong muốn:

\* 10

\* 20

\* 30

\* ...

\* 20

\* 40

\* 60

\* ...

\* Gợi ý: Dùng nested for loops (các vòng lặp lồng nhau)

\*/

Ví dụ

/\*\*

\* Viết function trả về tích các số từ start đến end, không tính end

\* Ví dụ: console.log(calculate(2, 5)) sẽ hiển thị ra màn hình 24 (vì 2 \* 3 \* 4 = 24)

\*/

function calculate(start, end) {

}

12 – For in , for of

Ví dụ

/\*\*

\* Sử dụng vòng lặp for...of để tính tích của các phần tử trong một array gồm các số

\*/

function multiply(arr) {

}

console.log(multiply([2, 3, 4])); // expect: 24

Ví dụ

/\*\*

\* Sử dụng vòng lặp for...in để in ra tất cả các key trong object sau, in ra cả các key của các nested object (object con lồng bên trong object lớn)

\* Gợi ý: đây là bài tập khó, bạn nên tìm hiểu về recursive trước

\*/

var apartment = {

bedroom: {

area: 20,

bed: {

type: 'twin-bed',

price: 100

}

}

};

function getkey(object){

}

getkey(apartment)

/\*\*

\* Kết quả mong muốn:

\* bedroom

\* area

\* bed

\* type

\* price

\* Chú ý: không cần hiển thị ra đúng thứ tự như trên

\*/

13 – Array method

Ví dụ

function removeEnd(arr, n) {

}

console.log(removeEnd([2, 3, 1, 8, 9, 7], 3));

Ví dụ

Viết hàm JavaScript để lấy phần tử đầu tiên của mảng. Truyền tham số 'n' sẽ trả về các phần tử 'n' đầu tiên của mảng

function first(arr, n) {

}

console.log(first([1, 2, 3], 2)); // expect [1, 2]

14 – Callback tham số

Ví dụ

/\*\*

\* 1. Viết hàm double nhận vào 1 số bất kì, hiển thị ra số đó nhân đôi

\* 2. Viết hàm sumAndDo nhận vào 1 array và 1 callback function. Hàm sumAndDo làm nhiệm vụ tính tổng array đó, sau đó gọi callback function với tham số là kết quả tổng vừa tính

\*/

function double(num) {

}

function sumAndDo(nums, callback) {

}

sumAndDo([1, 2, 3], double);

Ví dụ

/\*\*

\* 1. Viết hàm transform nhận vào 1 array gồm các số và 1 function callback. Nội dung của hàm bao gồm:

\* - Khai báo 1 biến result có giá trị là 1 empty array

\* - Lặp qua từng phần tử trong array, với mỗi phần tử, gọi hàm callback và truyền vào phần tử đó, được kết quả bao nhiêu thì push vào array khai báo ở trên

\* - Trả về result array

\*/

function transform(numbers, callback) {

// Viết nội dung hàm ở đây

}

function double(num) {

return num \* 2;

}

var output = transform([2, 4, 6], double);

console.log(output);

// Expect: [4, 8, 12]

15 – Hàm map

Ví dụ

/\*\*

\* 1. Viết hàm tripple nhận vào 1 số và trả về số đó nhân 3

\* 2. Sử dụng map method và hàm tripple để nhân 3 các phần tử trong mảng dưới đây

\*/

// Khai báo tripple function ở đây

var numbers = [10, 5, 8];

// Gợi ý: numbers.map... (tự viết tiếp)

Ví dụ

// Make an array of strings of the names

function namesOnly(arr) {

}

namesOnly([

{

name: "Angelina Jolie",

age: 80

},

{

name: "Eric Jones",

age: 2

},

{

name: "Paris Hilton",

age: 5

},

{

name: "Kayne West",

age: 16

},

{

name: "Bob Ziroll",

age: 100

}

])

// ["Angelina Jolie", "Eric Jones", "Paris Hilton", "Kayne West", "Bob Ziroll"]

Ví dụ

function renderAcreage(arr) {

}

arrayHeightWidth([

{

witdh: 4,

height : 6

},

{

witdh: 5,

height : 7

},

{

witdh: 1,

height : 3

},

{

witdh: 5,

height : 10

},

{

witdh: 10,

height : 10

}

])

16 – Hàm filter

Ví dụ:

Cho một mảng các số, trả về một mảng mới chỉ có các số từ 5 trở lên

function fiveAndGreaterOnly(arr) {

}

Ví dụ:

Cho một mảng số, trả về một mảng mới chỉ bao gồm các số chẵn

function evensOnly(arr) {

}

Ví dụ:

// Viết hàm trả về số chẵn đầu tiên trong mảng sử dụng find method

// Nếu không tìm thấy số chẵn nào thì trả về undefined

function findFirstEvenNumber(array) {

}

Ví dụ:

/\*\*

\* viết hàm tìm số chia hết cho x đầu tiên có trong mảng

\* @params {array} array - Mảng các chữ số

\* @params {number} x - Số bị chia

\* Nếu không có trả về undefined

\* Example:

\* findDivisibleNum([1, 3, 4], 2) => 4

\*/

function findDivisibleNum(array, x) {

// viết code ở đây.

}

17 – Hàm reduce

Ví dụ

// Trả về số lượng người đã vote

function totalVotes(arr) {

}

var voters = [

{name:'Bob' , age: 30, voted: true},

{name:'Jake' , age: 32, voted: true},

{name:'Kate' , age: 25, voted: false},

{name:'Sam' , age: 20, voted: false},

{name:'Phil' , age: 21, voted: true},

{name:'Ed' , age:55, voted:true},

{name:'Tami' , age: 54, voted:true},

{name: 'Mary', age: 31, voted: false},

{name: 'Becky', age: 43, voted: false},

{name: 'Joey', age: 41, voted: true},

{name: 'Jeff', age: 30, voted: true},

{name: 'Zack', age: 19, voted: false}

];

totalVotes(voters) // 7

Ví dụ:

// Tính tổng giá trị các món bên dưới

function shoppingSpree(arr) {

}

var wishlist = [

{ title: "Tesla Model S", price: 90000 },

{ title: "4 carat diamond ring", price: 45000 },

{ title: "Fancy hacky Sack", price: 5 },

{ title: "Gold fidgit spinner", price: 2000 },

{ title: "A second Tesla Model S", price: 90000 }

];

shoppingSpree(wishlist) // 227005

Ví dụ:

function countOccurrences(arr) {

}

countOccurrences(["a", "b", "c", "b", "a", "a"]);

18 – Array tổng hợp

Ví dụ

/\* Filter các sản phẩm khi biết id danh mục sản phẩm

Ví dụ đây là 1 danh sách các danh mục sản phẩm,

khi người dùng muốn lọc các sản phẩm thuộc danh mục nào

thì filter các sản phẩm của danh mục đó.

var categories = [

{ id: 1, name: 'Đồ điện tử'},

{ id: 2, name: 'Đồ gia dụng'},

{ id: 3, name: 'Nội thất'}

];

\*/

var listProducts = [

{id: 1, categoryId: 1, name: 'Tivi'},

{id: 2, categoryId: 1, name: 'Tủ lạnh'},

{id: 3, categoryId: 3, name: 'Ghế sofa'},

{id: 4, categoryId: 1, name: 'Máy giặt'},

{id: 5, categoryId: 2, name: 'Chén bát'},

{id: 6, categoryId: 2, name: 'Nồi cơm điện'},

{id: 7, categoryId: 3, name: 'Cửa kính'},

{id: 8, categoryId: 1, name: 'Điều hoà'},

{id: 9, categoryId: 3, name: 'Bàn tròn'},

{id: 10, categoryId: 2, name: 'Lò vi sóng'},

]

function filterProductsByCategoryId(products, categoryId) {

// your code here!

}