**Javascript Practice I**

**I. Functions**

No. 1 Tulislah sebuah function dengan nama teriak() yang mengembalikan nilai "Hello World!" yang kemudian dapat ditampilkan di console.

Output :

console.log(teriak()) // "Hello World!"

function teriak(){

    return "Hello World!";

    }

    console.log(teriak());

No. 2 Tulislah sebuah function dengan nama kalikan() yang mengembalikan hasil perkalian dua parameter yang di kirim.

Output:

var hasilKali = kalikan(num1, num2)

console.log(hasilKali) // 48

function kalikan(nilai1,nilai2){

    var hasil = nilai1 \* nilai2;

    return hasil;

}

var hasilkali = kalikan(2,24);

console.log(hasilkali);

No. 3 Tulislah sebuah function dengan nama introduce() yang memproses paramater yang dikirim menjadi sebuah kalimat perkenalan seperti berikut: "Nama saya [name], umur saya [age] tahun, alamat saya di [address], dan saya punya hobby yaitu [hobby]!"

Output:

var perkenalan = introduce(name, age, address, hobby)

console.log(perkenalan) // Menampilkan "Nama saya Ahmad, umur saya 24 tahun, dan saya punya hobby yaitu nonton Udemy!"

function perkenalan(name, age , address, hoby){

   var kalimat = "Nama saya "+name+", umur saya "+age+" tahun, alamat saya di "+address+", dan saya punya hobby yaitu "+hoby+"!";

   return kalimat;

}

console.log(perkenalan("nurholis", 27, "tasikmalaya", "main game"));

**II. Looping**

No. 1 Looping While

Pada tugas ini kamu diminta untuk melakukan looping dalam JavaScript dengan menggunakan syntax while. Untuk membuat tantangan ini lebih menarik, kamu juga diminta untuk membuat suatu looping yang menghitung maju dan menghitung mundur. Jangan lupa tampilkan di console juga judul 'LOOPING PERTAMA' dan 'LOOPING KEDUA'."

Output:

LOOPING PERTAMA

2 - I love coding

4 - I love coding

6 - I love coding

8 - I love coding

10 - I love coding

12 - I love coding

14 - I love coding

16 - I love coding

18 - I love coding

20 - I love coding

LOOPING KEDUA

20 - I will become a mobile developer

18 - I will become a mobile developer

16 - I will become a mobile developer

14 - I will become a mobile developer

12 - I will become a mobile developer

10 - I will become a mobile developer

8 - I will become a mobile developer

6 - I will become a mobile developer

4 - I will become a mobile developer

2 - I will become a mobile developer

var increment = 1;

while(increment <= 20) {

    if(increment%2==0){

    console.log(increment +"-I love coding");

}

    increment++; }

    var increment = 20;

    while(increment > 0) {

        if(increment%2==0){

        console.log(increment +"- I will become a mobile developer");

    }

        increment--; }

No. 2 Looping menggunakan for

Pada tugas ini kamu diminta untuk melakukan looping dalam JavaScript dengan menggunakan syntax for. Untuk membuat tantangan ini lebih menarik, kamu juga diminta untuk memenuhi syarat tertentu yaitu:

SYARAT:

A. Jika angka ganjil maka tampilkan Santai

B. Jika angka genap maka tampilkan Berkualitas

C. Jika angka yang sedang ditampilkan adalah kelipatan 3 DAN angka ganjil maka tampilkan I Love Coding.

OUTPUT

1 - Santai

2 - Berkualitas

3 - I Love Coding

4 - Berkualitas

5 - Santai

6 - Berkualitas

7 - Santai

8 - Berkualitas

9 - I Love Coding

10 - Berkualitas

11 - Santai

12 - Berkualitas

13 - Santai

14 - Berkualitas

15 - I Love Coding

16 - Berkualitas

17 - Santai

18 - Berkualitas

19 - Santai

20 - Berkualitas

for (let i = 1; i <= 20; i++) {

    if(i%3==0){

    console.log(i +"- I Love Coding");

    }

    else if(i%2==0){

        console.log(i +"-  Berkualitas");

        }

    else{

        console.log(i +"-   Santai");

    }

}

No. 3 Membuat Persegi Panjang #

Kamu diminta untuk menampilkan persegi dengan dimensi panjang x lebar dengan tanda pagar (#) dengan perulangan atau looping. Looping boleh menggunakan syntax apa pun (while, for, do while).

Output:

########

########

########

########

 var tinggi= 4;

    for (baris = 0; baris <= tinggi ; baris++) {

    for (i = 1; i < 8; i++) {

    document.write("#"); }

    document.write("<br>"); }

No. 4 Membuat Tangga

Kali ini kamu diminta untuk menampilkan sebuah segitiga sama sisi dengan tanda pagar (#) dengan panjang sisi sesuai parameter yang diberikan . Looping boleh menggunakan syntax apa pun (while, for, do while).

Output:

#

##

###

####

#####

######

#######

    var tinggi= 8;

    for (baris = 0; baris <= tinggi ; baris++) {

    for (i = 1; i < 1 \* baris; i++) {

    document.write("#"); }

    document.write("<br>"); }

No. 5 Membuat Papan Catur

Buatlah suatu looping untuk menghasilkan sebuah papan catur dengan ukuran sisi yang ditentukan oleh parameter. Papan berwarna hitam memakai tanda pagar (#) sedangkan papan putih menggunakan spasi. Looping boleh menggunakan syntax apa pun (while, for, do while).

Output:

# # # #

# # # #

# # # #

# # # #

# # # #

# # # #

# # # #

# # # #

var tinggi= 9;

    for (baris = 0; baris <= tinggi ; baris++) {

     if(baris%2==0){

        document.write("");

     }else{

     document.write("<span>&nbsp;</span>");}

    for (i = 1; i < 5; i++) {

    document.write(" #"); }

    document.write("<br>"); }

**III. String**

Soal No. 1 (Membuat kalimat),

Terdapat kumpulan variable dengan data string sebagai berikut

var word = 'JavaScript';

var second = 'is';

var third = 'awesome';

var fourth = 'and';

var fifth = 'I';

var sixth = 'love';

var seventh = 'it!';

Buatlah agar kata-kata di atas menjadi satu kalimat . Output:

JavaScript is awesome and I love it!

Soal No.2 Mengurai kalimat (Akses karakter dalam string),

Terdapat satu kalimat seperti berikut:

var sentence = "I am going to be React Native Developer";

var exampleFirstWord = sentence[0] ;

var exampleSecondWord = sentence[2] + sentence[3] ;

var thirdWord; // lakukan sendiri

var fourthWord; // lakukan sendiri

var fifthWord; // lakukan sendiri

var sixthWord; // lakukan sendiri

var seventhWord; // lakukan sendiri

var eighthWord; // lakukan sendiri

console.log('First Word: ' + exampleFirstWord);

console.log('Second Word: ' + secondWord);

console.log('Third Word: ' + thirdWord);

console.log('Fourth Word: ' + fourthWord);

console.log('Fifth Word: ' + fifthWord);

console.log('Sixth Word: ' + sixthWord);

console.log('Seventh Word: ' + seventhWord);

console.log('Eighth Word: ' + eighthWord)

Buat menjadi Output berikut:

First word: I

Second word: am

Third word: going

Fourth word: to

Fifth word: be

Sixth word: React

Seventh word: Native

Eighth word: Developer

Soal No. 3 Mengurai Kalimat (Substring)

var sentence2 = 'wow JavaScript is so cool';

var exampleFirstWord2 = sentence2.substring(0, 3);

var secondWord2; // do your own!

var thirdWord2; // do your own!

var fourthWord2; // do your own!

var fifthWord2; // do your own!

console.log('First Word: ' + exampleFirstWord2);

console.log('Second Word: ' + secondWord2);

console.log('Third Word: ' + thirdWord2);

console.log('Fourth Word: ' + fourthWord2);

console.log('Fifth Word: ' + fifthWord2);

Uraikan lah kalimat sentence2 di atas menjadi kata-kata penyusunnya.

Output:

First Word: wow

Second Word: JavaScript

Third Word: is

Fourth Word: so

Fifth Word: cool

Soal No. 4 Mengurai Kalimat dan Menentukan Panjang String

var sentence3 = 'wow JavaScript is so cool';

var exampleFirstWord3 = sentence3.substring(0, 3);

var secondWord3; // do your own!

var thirdWord3; // do your own!

var fourthWord3; // do your own!

var fifthWord3; // do your own!

var firstWordLength = exampleFirstWord3.length

// lanjutkan buat variable lagi di bawah ini

console.log('First Word: ' + exampleFirstWord3 + ', with length: ' + firstWordLength);

console.log('Second Word: ' + secondWord3);

console.log('Third Word: ' + thirdWord3);

console.log('Fourth Word: ' + fourthWord3);

console.log('Fifth Word: ' + fifthWord3);

Output:

First Word: wow, with length: 3

Second Word: JavaScript, with length: 10

Third Word: is, with length: 2

Fourth Word: so, with length: 2

Fifth Word: cool, with length: 4

**IV. Tugas Conditional**

No. 1 If-else

Petunjuk : Kita akan memasuki dunia game werewolf. Pada saat akan bermain kamu diminta memasukkan nama dan peran . Untuk memulai game pemain harus memasukkan variable nama dan peran. Jika pemain tidak memasukkan nama maka game akan mengeluarkan warning "Nama harus diisi!". Jika pemain memasukkan nama tapi tidak memasukkan peran maka game akan mengeluarkan warning "Pilih Peranmu untuk memulai game". Terdapat tiga peran yaitu penyihir, guard, dan werewolf. Tugas kamu adalah membuat program untuk mengecek input dari pemain dan hasil response dari game sesuai input yang dikirimkan.

Petunjuk:

Nama dan peran diisi manual dan bisa diisi apa saja

Nama tidak perlu dicek persis sesuai dengan input/output

Buat kondisi if-else untuk masing-masing peran

Input:

var nama = "John"

var peran = ""

Output:

// Output untuk Input nama = '' dan peran = ''

"Nama harus diisi!"

//Output untuk Input nama = 'John' dan peran = ''

"Halo John, Pilih peranmu untuk memulai game!"

//Output untuk Input nama = 'Jane' dan peran 'Penyihir'

"Selamat datang di Dunia Werewolf, Jane"

"Halo Penyihir Jane, kamu dapat melihat siapa yang menjadi werewolf!"

//Output untuk Input nama = 'Jenita' dan peran 'Guard'

"Selamat datang di Dunia Werewolf, Jenita"

"Halo Guard Jenita, kamu akan membantu melindungi temanmu dari serangan werewolf."

//Output untuk Input nama = 'Junaedi' dan peran 'Werewolf'

"Selamat datang di Dunia Werewolf, Junaedi"

"Halo Werewolf Junaedi, Kamu akan memakan mangsa setiap malam!"

No.2 Switch Case

Kamu akan diberikan sebuah tanggal dalam tiga variabel, yaitu hari, bulan, dan tahun. Disini kamu diminta untuk membuat format tanggal. Misal tanggal yang diberikan adalah hari 1, bulan 5, dan tahun 1945. Maka, output yang harus kamu proses adalah menjadi 1 Mei 1945.

Gunakan switch case untuk kasus ini!

Contoh:

var hari = 21;

var bulan = 1;

var tahun = 1945;

// Maka hasil yang akan tampil di console adalah: '21 Januari 1945';

Skeleton Code / Code yang dicontohkan yang perlu diikuti dan dimodifikasi

var tanggal; // assign nilai variabel tanggal disini! (dengan angka antara 1 - 31)

var bulan; // assign nilai variabel bulan disini! (dengan angka antara 1 - 12)

var tahun; // assign nilai variabel tahun disini! (dengan angka antara 1900 - 2200)

**V. Array**

Soal No. 1 (Range)

Buatlah sebuah function dengan nama range() yang menerima dua parameter berupa number. Function mengembalikan sebuah array yang berisi angka-angka mulai dari angka parameter pertama hingga angka pada parameter kedua. Jika parameter pertama lebih besar dibandingkan parameter kedua maka angka-angka tersusun secara menurun (descending).

struktur fungsinya seperti berikut range(startNum, finishNum) {}

Jika parameter pertama dan atau kedua tidak diisi maka function akan menghasilkan sebuah array kosong []

Contoh:

node array range 1 10

Output: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

node array range 54 50

Output : [54, 53, 52, 51, 50]

node array range 1

Output : []

node array range

Output: []

Soal No. 2 (Range with Step)

Pada soal kali ini kamu diminta membuat function rangeWithStep yang mirip dengan function range di soal sebelumnya namun parameternya ditambah dengan parameter ketiga yaitu angka step yang menyatakan selisih atau beda dari setiap angka pada array. Jika parameter pertama lebih besar dibandingkan parameter kedua maka angka-angka tersusun secara menurun (descending) dengan step sebesar parameter ketiga.

struktur fungsinya seperti berikut rangeWithStep(startNum, finishNum, step) {}

Perintah: node array rangeWithStep <startNum> <finishNum> <step>

contoh perintah :

node array rangeWithStep 1 10 2

Output: [1, 3, 5, 7, 9]

node array rangeWithStep 5 2 1

Output : [5, 4, 3, 2]

Soal No. 3 (Sum of Range)

Kali ini kamu akan menjumlahkan sebuah range (Deret) yang diperoleh dari function range di soal-soal sebelumnya. Kamu boleh menggunakan function range dan rangeWithStep pada soal sebelumnya untuk menjalankan soal ini.

Buatlah sebuah function dengan nama sum() yang menerima tiga parameter yaitu angka awal deret, angka akhir deret, dan beda jarak (step). Function akan mengembalikan nilai jumlah (sum) dari deret angka. contohnya sum(1,10,1) akan menghasilkan nilai 55.

ATURAN: Jika parameter ke-3 tidak diisi maka stepnya adalah 1.

Perintah: node array sum <startNum> <finishNum> <step>

Contoh:

node array sum 1 10

Output : 55

node array sum 5 50 2

Output : 621

node array sum 1

Output: 1

Soal No. 4 (Array Multidimensi)

Sering kali data yang diterima dari database adalah array yang multidimensi (array di dalam array). Sebagai developer, tugas kita adalah mengolah data tersebut agar dapat menampilkan informasi yang diinginkan.

Buatlah sebuah fungsi dengan nama dataHandling dengan sebuah parameter untuk menerima argumen. Argumen yang akan diterima adalah sebuah array yang berisi beberapa array sejumlah n. Contoh input dapat dilihat dibawah:

//contoh input

var input = [

["0001", "Roman Alamsyah", "Bandar Lampung", "21/05/1989", "Membaca"],

["0002", "Dika Sembiring", "Medan", "10/10/1992", "Bermain Gitar"],

["0003", "Winona", "Ambon", "25/12/1965", "Memasak"],

["0004", "Bintang Senjaya", "Martapura", "6/4/1970", "Berkebun"]

]

Tulis data input di atas di dalam file library yang kamu. data tersebut digunakan sebagai input.

Tugas kamu adalah mengimplementasikan fungsi dataHandling() agar dapat menampilkan data-data pada dari argumen seperti di bawah ini:

Perintah: node array dataHandling

Output:

Nomor ID: 0001

Nama Lengkap: Roman Alamsyah

TTL: Bandar Lampung 21/05/1989

Hobi: Membaca

Nomor ID: 0002

Nama Lengkap: Dika Sembiring

TTL: Medan 10/10/1992

Hobi: Bermain Gitar

Nomor ID: 0003

Nama Lengkap: Winona

TTL: Ambon 25/12/1965

Hobi: Memasak

Nomor ID: 0004

Nama Lengkap: Bintang Senjaya

TTL: Martapura 6/4/1970

Hobi: Berkebun

Soal No. 5 (Balik Kata)

Kamu telah mempelajari beberapa method yang dimiliki oleh String dan Array. String sebetulnya bersifat mirip sebuah array karena kita dapat mengakses karakter-karakter pada sebuah string layaknya mengakses elemen pada array.

Buatlah sebuah function balikKata() yang menerima sebuah parameter berupa string dan mengembalikan kebalikan dari string tersebut.

Perintah node array balikKata <kata>

Contoh Perintah:

node array balikKata PadjadjaranLab

Output: baLnarajdajdaP

Dilarang menggunakan sintaks .split , .join , .reverse() , hanya gunakan looping!

**VI. Object Literal**

Soal No. 1 (Array to Object)

Buatlah function dengan nama arrayToObject() yang menerima sebuah parameter berupa array multidimensi. Dalam array tersebut berisi value berupa First Name, Last Name, Gender, dan Birthyear. Data di dalam array dimensi tersebut ingin kita ubah ke dalam bentuk Object dengan key bernama : firstName, lastName, gender, dan age. Untuk key age ambillah selisih tahun yang ditulis di data dengan tahun sekarang. Jika tahun tidak terdefinisi atau tahunnya lebih besar dibandingkan dengan tahun sekarang maka kembalikan nilai : "Invalid birth year".

Contoh: jika input nya adalah [["Abduh", "Muhamad", "male", 1992], ["Ahmad", "Taufik", "male", 1985]]

maka outputnya di console seperti berikut :

1. Abduh Muhamad : { firstName: "Abduh", lastName: "Muhamad", gender: "male", age: 28}

2. Ahmad Taufik : { firstName: "Ahmad", lastName: "Taufik", gender: "male", age: 35}

Untuk mendapatkan tahun sekarang secara otomatis bisa gunakan Class Date dari Javascript.

var now = new Date()

var thisYear = now.getFullYear() // 2020 (tahun sekarang)

Code

function arrayToObject(arr) {

// Code di sini

}

// Driver Code

var people = [ ["Bruce", "Banner", "male", 1975], ["Natasha", "Romanoff", "female"] ]

arrayToObject(people)

/\*

1. Bruce Banner: {

firstName: "Bruce",

lastName: "Banner",

gender: "male",

age: 45

}

2. Natasha Romanoff: {

firstName: "Natasha",

lastName: "Romanoff",

gender: "female".

age: "Invalid Birth Year"

}

\*/

var people2 = [ ["Tony", "Stark", "male", 1980], ["Pepper", "Pots", "female", 2023] ]

arrayToObject(people2)

/\*

1. Tony Stark: {

firstName: "Tony",

lastName: "Stark",

gender: "male",

age: 40

}

2. Pepper Pots: {

firstName: "Pepper",

lastName: "Pots",

gender: "female".

age: "Invalid Birth Year"

}

\*/

// Error case

arrayToObject([]) // ""

Soal No. 2 (Shopping Time)

Problem

Diberikan sebuah function shoppingTime(memberId, money) yang menerima dua parameter berupa string dan number. Parameter pertama merupakan memberIddan parameter ke-2 merupakan value uang (money) yang dibawa oleh member tersebut.

Toko X sedang melakukan SALE untuk beberapa barang, yaitu:

Sepatu brand Stacattu seharga 1500000

Baju brand Zoro seharga 500000

Baju brand H&N seharga 250000

Sweater brand Uniklooh seharga 175000

Casing Handphone seharga 50000

Buatlah function yang akan mengembalikan sebuah object dimana object tersebut berisikan informasi memberId, money, listPurchased dan changeMoney.

Jika memberId kosong maka tampilkan "Mohon maaf, toko X hanya berlaku untuk member saja"

Jika uang yang dimiliki kurang dari 50000 maka tampilkan "Mohon maaf, uang tidak cukup"

Member yang berbelanja di toko X akan membeli barang yang paling mahal terlebih dahulu dan akan membeli barang-barang yang sedang SALE masing-masing 1 jika uang yang dimilikinya masih cukup.

Contoh jika inputan memberId: '324193hDew2' dan money: 700000

maka output:

{ memberId: '324193hDew2', money: 700000, listPurchased: [ 'Baju Zoro', 'Sweater Uniklooh' ], changeMoney: 25000 }

Code

function shoppingTime(memberId, money) {

// you can only write your code here!

}

// TEST CASES

console.log(shoppingTime('1820RzKrnWn08', 2475000));

//{ memberId: '1820RzKrnWn08',

// money: 2475000,

// listPurchased:

// [ 'Sepatu Stacattu',

// 'Baju Zoro',

// 'Baju H&N',

// 'Sweater Uniklooh',

// 'Casing Handphone' ],

// changeMoney: 0 }

console.log(shoppingTime('82Ku8Ma742', 170000));

//{ memberId: '82Ku8Ma742',

// money: 170000,

// listPurchased:

// [ 'Casing Handphone' ],

// changeMoney: 120000 }

console.log(shoppingTime('', 2475000)); //Mohon maaf, toko X hanya berlaku untuk member saja

console.log(shoppingTime('234JdhweRxa53', 15000)); //Mohon maaf, uang tidak cukup

console.log(shoppingTime()); ////Mohon maaf, toko X hanya berlaku untuk member saja

Soal No. 3 (Naik Angkot)

Problem

Diberikan function naikAngkot(listPenumpang) yang akan menerima satu parameter berupa array dua dimensi. Function akan me-return array of object.

Diberikan sebuah rute, dari A - F. Penumpang diwajibkan membayar Rp2000 setiap melewati satu rute.

Contoh: input: [ ['Dimitri', 'B', 'F'] ] output: [{ penumpang: 'Dimitri', naikDari: 'B', tujuan: 'F', bayar: 8000 }]

Code

function naikAngkot(arrPenumpang) {

rute = ['A', 'B', 'C', 'D', 'E', 'F'];

//your code here

}

//TEST CASE

console.log(naikAngkot([['Dimitri', 'B', 'F'], ['Icha', 'A', 'B']]));

// [ { penumpang: 'Dimitri', naikDari: 'B', tujuan: 'F', bayar: 8000 },

// { penumpang: 'Icha', naikDari: 'A', tujuan: 'B', bayar: 2000 } ]

console.log(naikAngkot([])); //[]

Soal No 4. Nilai Tertinggi

Problem

Buatlah sebuah function nilaiTertinggi yang menerima input berupa array of object berisi data-data peserta sanbercode beserta nilai nya. function akan mengembalikan sebuah object seperti berikut:

{

<nama-kelas> : { name: <nama-siswa>, score: <nilai-siswa> },

<nama-kelas> : { name: <nama-siswa>, score : <nilai-siswa> }

}

function nilaiTertinggi(siswa) {

// Code disini

}

// TEST CASE

console.log(nilaiTertinggi([

{

name: 'Asep',

score: 90,

class: 'adonis'

},

{

name: 'Ahmad',

score: 85,

class: 'vuejs'

},

{

name: 'Regi',

score: 74,

class: 'adonis'

},

{

name: 'Afrida',

score: 78,

class: 'reactjs'

}

]));

// OUTPUT:

// {

// adonis: { name: 'Asep', score: 90 },

// vuejs: { name: 'Ahmad', score: 85 },

// reactjs: { name: 'Afrida', score: 78}

// }

console.log(nilaiTertinggi([

{

name: 'Bondra',

score: 100,

class: 'adonis'

},

{

name: 'Putri',

score: 76,

class: 'laravel'

},

{

name: 'Iqbal',

score: 92,

class: 'adonis'

},

{

name: 'Tyar',

score: 71,

class: 'laravel'

},

{

name: 'Hilmy',

score: 80,

class: 'vuejs'

}

]));

// {

// adonis: { name: 'Bondra', score: 100 },

// laravel: { name: 'Putri', score: 76 },

// vuejs: { name: 'Hilmy', score: 80 }

// }

**VII. ES6**

Buatlah Folder baru dengan nama "Tugas - ES6" di repositori tugas harian kalian.

Buatlah satu folder di dalam folder tersebut dengan nama folder "src". Kerjakanlah tugas berikut ini pada folder src dengan menggunakan Sintaks ES6 yang telah kalian pelajari.

Folder src tersebut dilakukan compile menggunakan Babel dengan output folder "dist". (Gunakan perintah git init untuk memulai project npm, lalu install library Babel js npm install @babel/core @babel/cli @babel/preset-env --save)

CATATAN: WAJIB menggunakan export & import module. struktur folder tugas terdiri dari entry file index.js dan folder library lib . function dibuat di file yang ada di folder lib lalu diexport dan diimport ke index.js. Perintah menjalankan soal harus sesuai perintah.

Soal 1 Function Sapa

buatlah sebuah function yang menerima satu parameter berupa string nama. function tersebut me-return kalimat sapaan berdasarkan parameter nama yang diberikan.

Hint: gunakan arrow function, template literals

Perintah :

node dist sapa <nama>

Contoh Perintah:

node dist sapa Abduh

Output

"halo selamat pagi, Abduh"

Soal 2 Konversi Object

Buatlah sebuah function yang menerima tiga parameter yaitu nama, domisili, dan umur. function akan mengembalikan object berisi data dari parameter yang diberikan.

Perintah

node dist convert <nama> <domisili> <umur>

Contoh Perintah

node dist convert Bondra Medan 25

Output

{ "nama" : "Bondra", "domisili": "Medan", "umur": 25 }

Soal 3 Check Score

Buatlah sebuah function yang menerima satu parameter berupa string. string tersebut berisi data-data yang dipisahkan dengan tanda koma(,) dan titik dua (:). Function tersebut mengolah data parameter menjadi sebuah object berdasarkan data yang dikirim dari parameter.

Hint: Destructuring, Enhanced Objects, arrow functions

Perintah

node dist checkScore <string-data>

Contoh Perintah

node dist checkScore name:Ahmad,class:Adonis,score:89

Output

{ "name": "Ahmad", "class": "Adonis", "score": 89 }

Soal 4 Filter Data

Buatlah sebuah function yang menerima dua paramater berupa array of object berisi data peserta bootcamp Indonesia Mengoding, dan parameter string nama kelas. Function tersebut akan mengembalikan array of object berisi data peserta yang sudah terfilter menggunakan parameter nama kelas yang diberikan.

const data = [

{ name: "Ahmad", class: "adonis"},

{ name: "Regi", class: "laravel"},

{ name: "Bondra", class: "adonis"},

{ name: "Iqbal", class: "vuejs" },

{ name: "Putri", class: "Laravel" }

]

Perintah:

node dist filterData <nama-kelas>

Contoh Perintah:

node dist filterData adonis

Output:

[

{ name: "Ahmad", class: "adonis"} ,

{ name: "Bondra", class: "adonis"}

]

**VIII. Class Javascript & OOP**

Petunjuk Pengerjaan

Ikutilah langkah-langkah pengerjaan di bawah ini:‌

1. Menambahkan Folder Baru dan file baru

Buat folder baru dengan nama “Tugas - Berlatih Class”

2. Kerjakan Soal di bawah ini

kerjakan soal di bawah ini di dalam folder “Tugas - Berlatih Class”

Kita akan membuat program untuk mensimulasikan sebuah Bootcamp. Di dalam sebuah Bootcamp terdapat banyak pilihan Kelas yang dapat diikuti oleh banyak Student.

Release 0 Siapkan class Bootcamp dan class Kelas

Buatlah class Bootcamp di file dengan nama bootcamp.js dan class Kelas di kelas.js.

Tentukanlah property dan method apa saja yang terdapat di sebuah bootcamp. Misalkan properti nya \_name, \_classes dan method nya : getter name dan getter classes untuk mendapatkan data nama Bootcamp dan Kelas apa saja yang dimiliki. Selain itu ada method createClass() yaitu metode untuk membuat Kelas baru.

Tentukanlah property dan method apa saja yang terdapat di sebuah Kelas. Misalkan properti nya name yaitu nama kelas, level yaitu tingkat kesulitan kelas (beginner, intermediate, advanced), instructor yaitu nama instruktur di kelas tersebut. buatlah getter dan setter untuk semua property tersebut.

Hint: Kalian tentukan apakah Bootcamp dengan Kelas memiliki relasi Aggregation atau Composition lalu implementasikan pada class Bootcamp.

Contoh ketika kita membuat object Bootcamp baru dan membuat kelas Baru berdasarkan nama, level dan nama instrukturnya:

const sanber = new Bootcamp("sanbercode")

sanber.createClass("Laravel", "beginner", "abduh")

sanber.createClass("React", "beginner", "abdul")

console.log(sanber.classes)

Output console nya akan seperti berikut

[

Kelas {

\_name: 'Laravel',

\_students: [],

\_level: 'beginner',

\_instructor: 'abduh'

},

Kelas {

\_name: 'React',

\_students: [],

\_level: 'beginner',

\_instructor: 'abdul'

}

]

Release 1 Register Student

Suatu Bootcamp dapat membuka pendaftaran bagi setiap Student yang ingin mendaftar ke Kelas tertentu. Buatlah class Student di satu file dengan nama student.js. Tentukanlah property dan method yang ada pada object student.

Property :

\_name : nama student

\_scores: [] nilai student sebuah array berisi 4 elemen dengan tipe data number. ke-empat elemen tersebut adalah nilai pada masing-masing quiz pekanan.

Hint: Kalian tentukan apakah di antara Bootcamp, Kelas dengan Student memiliki relasi Aggregation atau Composition lalu implementasikan pada program kalian.

Pada release ini daftarkanlah nama-nama student berikut ke dalam list array student pada masing-masing kelas "Laravel" dan "React" di Bootcamp. Cara menentukan kelas untuk masing-masing student adalah berdasarkan index array genap dan ganjil. Jika genap maka masuk ke kelas Laravel, jika ganjil maka masuk kelas React.

let names = ["regi", "ahmad", "bondra", "iqbal", "putri", "rezky"]

names.map((nama, index) => {

let newStud = new Student(nama)

let kelas = sanber.classes[index % 2].name

sanber.register(kelas, newStud)

})

// menampilkan data kelas dan student nya

sanber.classes.forEach(kelas => {

console.log(kelas)

});

Maka output console nya akan seperti berikut:

Kelas {

\_name: 'Laravel',

\_students: [

Student { \_name: 'regi', \_scores: [], \_finalScore: 0 },

Student { \_name: 'bondra', \_scores: [], \_finalScore: 0 },

Student { \_name: 'putri', \_scores: [], \_finalScore: 0 }

],

\_level: 'beginner',

\_instructor: 'abduh'

}

Kelas {

\_name: 'React',

\_students: [

Student { \_name: 'ahmad', \_scores: [], \_finalScore: 0 },

Student { \_name: 'iqbal', \_scores: [], \_finalScore: 0 },

Student { \_name: 'rezky', \_scores: [], \_finalScore: 0 }

],

\_level: 'beginner',

\_instructor: 'abdul'

}

Release 2 Graduate

Buatlah Method pada class Kelas untuk menentukan predikat kelulusan dari seluruh Student yang terdaftar pada kelas tersebut. methode ini mengembalikan sebuah object berisi tiga kategori lulusan : participant, completed, mastered.

kategori tersebut:

participated : nilai akhir kurang dari 60

completed: nilai akhir antara 60 dan 85

mastered: nilai akhir lebih dari 85

Contoh Output dari method graduate:

{

participant: [],

completed: [

Student { \_name: 'ahmad', \_scores: [Array], \_finalScore: 78 },

Student { \_name: 'iqbal', \_scores: [Array], \_finalScore: 74 },

Student { \_name: 'rezky', \_scores: [Array], \_finalScore: 70 }

],

mastered: []

}

Release 3 Method runBatch

Pada kelas Bootcamp terdapat method runBatch() untuk menjalankan satu batch bootcamp hingga lulus(graduate) di semua kelas. Di dalam method tersebut akan dilakukan perulangan sebanyak 4 kali yang menandakan 4 pekan pelaksanaan bootcamp. Di setiap perulangan, setiap Student akan ter-generate nilai secara random antara 50 - 100.

Di akhir method tersebut akan ditampilkan seluruh data kelas beserta data kelulusan pada kelas tersebut.

Contoh Output console nya seperti berikut :

graduated from Laravel: {

participant: [],

completed: [

Student { \_name: 'regi', \_scores: [Array], \_finalScore: 77 },

Student { \_name: 'bondra', \_scores: [Array], \_finalScore: 77 }

],

mastered: []

}

graduated from React: {

participant: [],

completed: [

Student { \_name: 'ahmad', \_scores: [Array], \_finalScore: 78 },

Student { \_name: 'iqbal', \_scores: [Array], \_finalScore: 74 },

Student { \_name: 'rezky', \_scores: [Array], \_finalScore: 70 }

],

mastered: []

}