/\*

Practica **1**

**1.** Crear un script para calcular el IMC (Índice de masa corporal) de cualquier persona, y determinar la composición corporal de dicha persona bajo los siguientes criterios:

Peso inferior al normal : IMC Menos de **18.5**

Normal : IMC **18.5** – **24.9**

Peso superior al normal : IMC **25.0** – **29.9**

Obesidad: IMC Más de **30.0**

formula = peso/(altura^**2**)

-Pedir la altura en metros

-Pedir el peso en kilos

-Aplicar la formula

-Evaluar el resultado en base a los criterios definidos

\*/

var altura = prompt("Digite su altura en metros")

var peso = prompt("Digite su peso en Kilos")

let formula = peso/(altura^**2**)

**if** (formula < **18.5**){

alert("Peso inferior al normal")

}**else** **if**(formula >= **18.5** && formula <= **24.9**){

alert("Peso normal")

}**else** **if**(formula >= **25.0** && formula <= **29.9**){

alert("Peso superior al normal")

}**else**{

alert("Presenta Obesidad")

}

/\*

Practica **2**

**2.** Usando como referencia los siguientes datos de estaturas promedio:

General Varón Mujer

México **169.9** **170.0** **160.8** cm's

Australia **172.6** **179.2** **165.9** cm's

Canadá **171.0** **178.1** **163.9** cm's

Brasil **167.3** **173.6** **160.9** cm's

Reino Unido **171.0** **177.5** **164.4** cm's

crear un script que me permita conocer si mi estatura es superior,

inferior o igual al promedio dependiendo de mi país y género.

Si mi país no se encuentra en la lista, indicar que no se cuenta

con el dato de estatura para ese país.

-recibir el dato **del** país

-recibir el dato de la estatura

-comprar los datos con base en la información brindada

-recibir el dato **del** pais

-recibir el dato de la estatura

-comparar los datos con base a la informacion brindada

condition ? result **if** true : result **if** false

gender === "masculino" ? averageHeight = **170** : averageHeight = **160.8**

encadenamoento de alternarios

averageHeight = gender === "masculino" ? **170** :

gender === "femenino" ? **160.8** :

"genero no valido"

\*/

let pais = prompt("Digita tu pais")

let Altura = prompt("Digita tu altura")

let genero = prompt("digita tu genero")

let altura\_media;

switch(pais){

case "mexico":

altura\_media = genero === "hombre" ? **170** : **160.8**

**if**(Altura < altura\_media ){

alert("Tu altura es menor al promedio")

}**else** **if**(Altura > altura\_media){

alert("Tu altura es mayor al promedio")

}**else**{

alert("Tu altura es promedio")

}

**break**

case "australia":

altura\_media = genero === "hombre" ? **179.2** : **165.9**

**if**(Altura < altura\_media ){

alert("Tu altura es menor al promedio")

}**else** **if**(Altura > altura\_media){

alert("Tu altura es mayor al promedio")

}**else**{

alert("Tu altura es promedio")

}

**break**

case "canada":

altura\_media = genero === "hombre" ? **178.1** : **163.9**

**if**(Altura < altura\_media ){

alert("Tu altura es menor al promedio")

}**else** **if**(Altura > altura\_media){

alert("Tu altura es mayor al promedio")

}**else**{

alert("Tu altura es promedio")

}

**break**

case "brazil":

altura\_media = genero === "hombre" ? **173.6** : **160.9**

**if**(Altura < altura\_media ){

alert("Tu altura es menor al promedio")

}**else** **if**(Altura > altura\_media){

alert("Tu altura es mayor al promedio")

}**else**{

alert("Tu altura es promedio")

}

**break**

case "reino unido":

altura\_media = genero === "hombre" ? **177.5** : **164.4**

**if**(Altura < altura\_media ){

alert("Tu altura es menor al promedio")

}**else** **if**(Altura > altura\_media){

alert("Tu altura es mayor al promedio")

}**else**{

alert("Tu altura es promedio")

}

**break**

default:

alert("Tu pais no se encuentra en la base")

}

/\*

practica **3**

lados de un triangulo con su perimetro y area

\*/

alert("definamos que tipo de triangulo es, digite 3 numeros enteros")

let lado1 = parseFloat(prompt("digite lado A"))

let lado2 = parseFloat(prompt("digite lado B"))

let lado3 = parseFloat(prompt("digite lado C"))

var perimetro = lado1+lado2+lado3

let s = perimetro/**2**

let area = Math.sqrt(s\*(s-lado1)\*(s-lado2)\*(s-lado3))

**if**(lado1===lado2 && lado1 === lado3){

alert("equilatero")

}**else** **if**(lado1===lado2 || lado1 === lado3){

alert("escaleno")

}**else**[

alert("isoceles")

]

alert("perimetro = "+ perimetro )

alert("area = "+ area )

/\*

Practica **4**

Usando la frase "La mejor forma de predecir el futuro es creándolo" lograr los siguientes resultados:

**1.**- Mostrar la misma frase en snake\_case

**2.**- Mostrar la misma frase en kebab-case

**3.**- Mostrar la misma frase con todas las vocales en mayúscula

**4.**- Mostrar los primeros **10** caracteres de la frase

**5.**- Mostrar los últimos **10** caracteres de la frase

\*/

let frase = "La mejor forma de predecir el futuro es creándolo"

console.log(frase.replace(/ /g,"\_"))

console.log(frase.replace(/ /g,"-"))

console.log(frase.replace(/a|e|i|o|u/gi, function(change){

**return** change.toUpperCase();

}))

let diez = frase.charAt(**0**)

**for**(i=**1**;i<**10**;i++){

diez +=frase.charAt(i)

}

console.log(diez)

let ultdiez = frase.charAt(**39**)

**for**(i=**40**;i<=**48**;i++){

ultdiez += frase.charAt(i)

}

console.log(ultdiez)

let reverza = frase.charAt(**48**)

**for**(i=**47**;i>=**39**;i--){

reverza +=frase.charAt(i)

}

console.log(reverza)

/\*

Practica **5**

\*/

function dameUsuario(){

let numero = prompt("Digita los uruarios a registrar")

**for**(let i=**0**; i<numero; i++){

let usuario = prompt("digita el nombre de usuario")

let apellido = prompt("digita el primer apellido de usuario")

let apellido1 = prompt("digita el segundo apellido de usuario")

// console.log(usuario+" "+apellido+" "+apellido1)

let nArray = Array.of(usuario, apellido, apellido1)

//-> console.log(nArray)

}

}

//dameUsuario()

/\*

Practica **6**

\*/

let arrayVacio = new Array()

//->console.log(arrayVacio)

text = "jonathan"

//console.log(text)

//console.log(Array.from(text))

//console.log(Array.of(text))

//console.log(Array.of("jonathan","Reyes","Alatorre"))

//arrayNombres

let musicians = [

"Gustavo Cerati",

"Enrique Bunbury",

"Roy Orvison",

]

let painters = [

"Gustavo Cerati",

"Enrique Bunbury",

"Roy Orvison",

]

let writers = [

"Gustavo Cerati",

"Enrique Bunbury",

"Roy Orvison",

]

function dameIniciales(nombreCompleto){

//-> console.log(nombreCompleto)

let nombreDividido = nombreCompleto.split(" ")

let inicial = "";

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

let palabra = nombreDividido[i]

let inicial = palabra.charAt(**0**)

inicial += ". "

//-> console.log(inicial)

}

}

function imprimirInicial(arrayNombres){

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

dameIniciales(arrayNombres[i])

}

}

imprimirInicial(musicians)

console.log(" ")

imprimirInicial(painters)

console.log(" ")

imprimirInicial(writers)

console.log(writers)

delete writers[**1**];

writers = writers.filter((valor)=>{**return** valor != undefined;})

console.log(writers)

writers.splice(**0**,**1**,'DATO MODIFICADO')

writers.splice(**0**,**0**)//---> Dato eliminado

console.log(writers)

console.log(painters)

console.log(painters[painters.length-**1**])

console.log(painters[painters.length-**2**])

console.log(painters[painters.length-**3**])

/\*

Practica **7**

Teniendo en cuenta el siguiente array

-crear un nuevo array que tenga los nombres completos de todos los KoderList

\*/

var KoderList = [

[

"Fernanda",

"Palacios Vera"

],

[

"Jorge",

"Ochoa"

],

[

"Naomi",

"puertos"

],

[

"Rurick",

"maqueo Poissot"

]

]

//-crear un nuevo array que tenga los nombres completos de todos los KoderList

const crearNuevoArray = () => {

let nuevoArray = []

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

nuevoArray.push(KoderList[i].join(" "))

}

**return** nuevoArray

}

let verNuevoArray = crearNuevoArray()

console.log(verNuevoArray)

//-Ingresar un nombre

//-Validar si existe algun

// koder cuyo primer nombre coincida con el nombre escrito

const checarNombre = nombre => {

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

let primerNombre = verNuevoArray[i].split(" ")[**0**]

primerNombre === nombre && console.log("True")

}

}

checarNombre(prompt("digite nombre"))

/\*

Practica **8**

-quiero conocer la cantidad de productos en el array

-quiero conocer el costo total de todos los productos **del** array

-quiero conocer la cantidad de productos de cada categoría

(Vegetables, Cloths, Miscellaneous, Computers)

-quiero un nuevo array para cada categoría

-quiero un nuevo array de strings que tenga lo siguiente

[

"El { productName } tiene un costo de { price }"

]

\*/

var productsList = [

{

name:"producto 1",

price:**12.45**,

category:"Miscellaneous"

},

{

name:"producto 2",

price:**25.13**,

category:"Vegetables"

},

{

name:"producto 3",

price:**45.00**,

category:"Cloths"

},

{

name:"producto 4",

price:**12500.00**,

category:"Computers"

},

{

name:"producto 5",

price:**15500.00**,

category:"Computers"

},

{

name:"producto 6",

price:**8729.00**,

category:"Computers"

},{

name:"producto 7",

price:**150.00**,

category:"Cloths"

},{

name:"producto 8",

price:**100.00**,

category:"Miscellaneous"

},{

name:"producto 9",

price:**50.00**,

category:"Vegetables"

},{

name:"producto 10",

price:**200.00**,

category:"Cloths"

}

]

const cantidadDeProductos = array => array.length

let totalDeProductos = cantidadDeProductos(productsList)

console.log("Productos",totalDeProductos)

const productosPrecio = array =>{

let precioTotal = **0**

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

precioTotal += array[i].price

}

**return** precioTotal

}

let precioTotal = productosPrecio(productsList)

console.log("precio total", precioTotal)

const filtrarPorCategoria = (array, seleccionaCategoria )=> {

let resultado = []

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

array[i].category === seleccionaCategoria && resultado.push(array[i])

}

**return** resultado

}

let Buscar = filtrarPorCategoria(productsList, prompt("digita las opciones Miscellaneous, Vegetables, Cloths, Computers"))

console.log(Buscar)

console.log("Saber la cantidad de producto por categoria")

let cantidad = cantidadDeProductos(Buscar)

console.log("Miscellaneous "+cantidad)

console.log("Saber el precio total del producto por categoria")

let precio = productosPrecio(Buscar)

console.log("Miscellaneous costo total : $"+precio)

const crearArray = array =>{

let resultado = []

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

let frase = `El ${array[i].name} tiene un costo de ${array[i].price}`

resultado.push(frase)

}

**return** resultado

}

let frase = crearArray(productsList)

console.log(frase)

const names = ['jonathan','reyes','alatorre']

names.forEach((names, index) => {

console.log(names+" - "+index)

})

/\*

practica **9**

\*/

const imprimirNombre = () => {

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

let nombre = productsList[i].name

console.log(nombre)

}

}

imprimirNombre()

const imprimirNombre2 = () => {

productsList.forEach((productos, index) => console.log(productos.name+" "+index))

}

imprimirNombre2()

const crearArray = array => {

let resultado = []

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

let frase = `El ${array[i].name} tiene un costo de ${array[i].price}`

resultado.push(frase)

}

**return** resultado

}

const crearArray2 = () => {

let nuevoArray = productsList.map(producto => `El ${producto.name} tiene un costo de ${producto.price}`)

**return** nuevoArray

}

const filtrarCategoria = ( array, seleccionaCategoria) => {

let resultado = []

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

array[i].category === seleccionaCategoria && resultado.push(array[i])

}

**return** resultado

}

const filtrarCategoria2 = (array, seleccionaCategoria) => {

let resultado = array.filter(item => item.category === seleccionaCategoria)

}

//usando mentorsArray, realizar lo siguiente:

//-Obtener el score promedio de cada materia( HTML, CSS, JS, ReactJS )

//-Obtener el promedio individual de cada mentor

//-Obtener un array de strings con la siguiente forma:

// "Mi nombre es {nombre} y mi promedio es de {promedio}"

//-Obtener la lista de mentores cuyo promedio sea mayor a **9.5**

var mentorsArray = [

{

name:"Israel Salinas Martinez",

scores:[

{

signature:"HTML",

score:**8**

},

{

signature:"CSS",

score:**10**

},

{

signature:"JS",

score:**8**

},

{

signature:"ReactJS",

score:**8**

}

]

},

{

name:"David Cermeño Moranchel",

scores:[

{

signature:"HTML",

score:**8**

},

{

signature:"CSS",

score:**7**

},

{

signature:"JS",

score:**10**

},

{

signature:"ReactJS",

score:**10**

}

]

},

{

name:"Charles Silva",

scores:[

{

signature:"HTML",

score:**9**

},

{

signature:"CSS",

score:**9**

},

{

signature:"JS",

score:**10**

},

{

signature:"ReactJS",

score:**9**

}

]

},

{

name:"Michael Villalba Sotelo",

scores:[

{

signature:"HTML",

score:**8**

},

{

signature:"CSS",

score:**10**

},

{

signature:"JS",

score:**9**

},

{

signature:"ReactJS",

score:**10**

}

]

}

]

const ObtenerPromedio = dequemateria => {

let totalMateria = mentorsArray.reduce((acumulador, item) => {

let puntuacionMateria = item.scores.find(score => score.signature === dequemateria).score

**return** acumulador + puntuacionMateria

},**0**)

let promedio = totalMateria / mentorsArray.length

**return** promedio

}

let indicaMateria = ObtenerPromedio (prompt("Indica HTML, CSS, JS, ReactJS"))

console.log("Promedio :", indicaMateria)

// .find() devuelve el valor **del** primer elelemnto que cumpla una condicion dada

const obtenerPromedioMentor = dequementor => {

let seleccionaElMentor = mentorsArray.find( nombredelmentor => nombredelmentor.name === dequementor ).name

let seleccionaElMentor1 = mentorsArray.find( nombredelmentor => nombredelmentor.name === dequementor ).scores

let obtenerPromedio = seleccionaElMentor1.reduce( (acumulador, item) => {

**return** acumulador + item.score

},**0**) / seleccionaElMentor1.length

console.log("El promedio del mentor", seleccionaElMentor,"es :", obtenerPromedio)

**return** obtenerPromedio

}

let eligeMentor = obtenerPromedioMentor(prompt("indica mentor : Israel Salinas Martinez, David Cermeño Moranchel, Charles Silva, Michael Villalba Sotelo"))

console.log(eligeMentor)

const crearMatrizdeEtiquetas = () => {

let matriz = mentorsArray.map( variable => {

**return** `hola, soy ${variable.name} y mi promedio es de ${obtenerPromedioMentor(variable.name)}`

})

console.log(matriz)

}

crearMatrizdeEtiquetas()

const mejoresMentores = () => {

let resultado = mentorsArray.filter ( quementores => obtenerPromedioMentor( quementores.name) > **9**)

console.log(resultado)

}

mejoresMentores()

/\*

Practica **10**

funcion constructora

<h2>Challenge:</h2>

<p>💫 Generar una colección de Objetos de tipo Product que tenga las siguientes propiedades:</p>

<ul>

<li>nombre</li>

<li>description</li>

<li>precio de costo</li>

<li>clasificación</li>

<li>porcentaje de Ganancia</li>

<li>precio de venta</li> precio de costo + porcentaje de Ganancia

<li>precios de los ultimos **6** meses en una colección</li> precios aleatorios de **6** lengths

<li>Cantidad vendidas en la ultima semana</li> aleatorio en lengths

<li>Fecha de Caducidad</li>

</ul>

<p>💫 De igual forma es necesario realizar lo siguientes calculos por cada Producto:</p>

<ol>

<li>Calcular el precio de Venta y asignarlo a su correspondiente propiedad</li>

prescio de costo + porcentage de de ganancia = precio de venta

<li>Obtener los diás restantes para que caduque el producto</li>

<li>Obtener el promedio de precio de los ultimos **6** meses</li>

</ol>

<p>💫 Una vez obtenido la colleción de Productos obtener lo siguiente:</p>

<ol>

<li>Los productos que tengan mayor a **50** cantidades vendidas en la ultima semana</li>

<li>Los productos que tengan menor a **10** cantidades vendidas en la ultima semana</li>

<li>Los productos que se encuentren a menos de **15** dí**as** proximas a caducar</li>

</ol>

\*/

const Products = [

{

nombre: "leche",

descripcion: "Bote de leche de 1.5 lt tetrapac",

precio\_de\_costo: **15.50**,

clasificación: "Lacteo",

porcentaje\_de\_Ganancia: ".20",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **11**,

Fecha\_de\_Caducidad: "03/10/2022",

},

{

nombre: "crema",

descripcion: "Bote con 1lt de crema",

precio\_de\_costo: **15.50**,

clasificación: "Lacteo",

porcentaje\_de\_Ganancia: ".20",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **50**,

Fecha\_de\_Caducidad: "2022-05-05",

},

{

nombre: "huevo",

descripcion: "Paquete de 12 blamquillos",

precio\_de\_costo: **31.80**,

clasificación: "ovino",

porcentaje\_de\_Ganancia: ".50",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **5**,

Fecha\_de\_Caducidad: "2022-03-30",

},

{

nombre: "azucar clara",

descripcion: "Bolsa de 1k",

precio\_de\_costo: **10.90**,

clasificación: "costales",

porcentaje\_de\_Ganancia: ".20",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **30**,

Fecha\_de\_Caducidad: "2022-08-01",

},

{

nombre: "azucar morena",

descripcion: "Bolsa de 1k",

precio\_de\_costo: **10.90**,

clasificación: "costales",

porcentaje\_de\_Ganancia: ".20",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **45**,

Fecha\_de\_Caducidad: "2022-03-10",

},

{

nombre: "Jugo de naranja",

descripcion: "bote de 600mlt",

precio\_de\_costo: **12.90**,

clasificación: "bebidas",

porcentaje\_de\_Ganancia: ".20",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **51**,

Fecha\_de\_Caducidad: "2022-04-08",

},

{

nombre: "prueba",

descripcion: "prueba",

precio\_de\_costo: **50**,

clasificación: "prueba",

porcentaje\_de\_Ganancia: ".50",

precio\_de\_venta: **0**,

precios\_de\_los\_ultimos\_6\_meses:[**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**],

Cantidad\_vendidas\_en\_la\_ultima\_semana: **50**,

Fecha\_de\_Caducidad: "1970-01-01",

}

]

function Product (nombre, descripcion, precio\_de\_costo, clasificación, porcentaje\_de\_Ganancia, precio\_de\_venta, precios\_de\_los\_ultimos\_6\_meses = [], Cantidad\_vendidas\_en\_la\_ultima\_semana, Fecha\_de\_Caducidad ){

this.nombre = nombre;

this.descripcion = descripcion;

this.precio\_de\_costo = precio\_de\_costo;

this.clasificación = clasificación;

this.porcentaje\_de\_Ganancia = porcentaje\_de\_Ganancia;

this.precio\_a\_la\_venta = function () { //precio de costo + porcentaje de Ganancia

let precio = this.precio\_de\_costo;

let porcentage = this.porcentaje\_de\_Ganancia;

let precio\_mas\_porcentage = Math.round( ( precio \* porcentage ) + precio )

console.log("$",precio\_mas\_porcentage)

**return** precio\_mas\_porcentage

};

this.precio\_a\_la\_venta = this.precio\_a\_la\_venta()

this.precios\_de\_los\_ultimos\_6\_meses = precios\_de\_los\_ultimos\_6\_meses; //precios aleatorios de **6** lengths

this.Cantidad\_vendidas\_en\_la\_ultima\_semana = Cantidad\_vendidas\_en\_la\_ultima\_semana; //aleatorio en lengths

this.Fecha\_de\_Caducidad = Fecha\_de\_Caducidad;

this.Diasrestantes = function() {

let caduca = new Date(Fecha\_de\_Caducidad)

let hoy = new Date();

let milSDia = **24**\***60**\***60**\***1000**;

let milSTransc = Math.abs( caduca-hoy )

let diasTranscurridos = Math.round( milSTransc/milSDia );

console.log("caduca en ",diasTranscurridos+**1**," Dias")

**return** diasTranscurridos

}

this.Dias\_para\_caducar = this.Diasrestantes()

this.promedio\_Meses = function(){

const promedio = this.precios\_de\_los\_ultimos\_6\_meses.reduce((acumulador, item)=>{

**return** (acumulador + item);

},**0**)/this.precios\_de\_los\_ultimos\_6\_meses.length

console.log("promedio de precio de los ultimos 6 meses",promedio)

**return** promedio

}

this.promedio\_6\_Meses = this.promedio\_Meses()

}

const Productos = Products.map( item => {

**return** new Product(item.nombre, item.descripcion, item.precio\_de\_costo, item.clasificación, item.porcentaje\_de\_Ganancia, item.precio\_de\_venta, item.precios\_de\_los\_ultimos\_6\_meses, item.Cantidad\_vendidas\_en\_la\_ultima\_semana, item.Fecha\_de\_Caducidad )

})

console.log(Productos)

const nuevoProducto = new Product("pan", "pan de caja", **50**, "Cereales", ".50", **50**, [**10.5**,**11.10**,**12.15**,**13.40**,**14.80**,**15.60**], **50**, "2022-04-08" )

console.log(nuevoProducto)

// Los productos que tengan mayor a **50** cantidades vendidas en la ultima semana

const encimaCincuenta = Products.filter(item => {

**return** item.Cantidad\_vendidas\_en\_la\_ultima\_semana > **50**;

})

console.log("Productos con mayor a 50 cantidades vendidas en la ultima semana",encimaCincuenta)

// Los productos que tengan menor a **10** cantidades vendidas en la ultima semana

const debajoDiez = Products.filter(item => {

**return** item.Cantidad\_vendidas\_en\_la\_ultima\_semana < **10**;

})

console.log("Productos con mayor a 50 cantidades vendidas en la ultima semana",debajoDiez)

// Los productos que se encuentren a menos de **15** dí**as** proximas a caducar

const menos\_de\_15\_días\_vigencia = Products.filter(item => {

**return** item.Dias\_para\_caducar < **15**;

})

console.log("menos de 15 días proximas a caducar",menos\_de\_15\_días\_vigencia)

/\*

Practica **11**

Teniendo la siguiente coleccion de Koders generar una coleccion de objetos de tipo Koder.

Agregando además las siguientes funciones:

Obtener la edad a partir de la fecha de nacimiento

Obtener promedio de sus scores

Colección de Koder que pertenezcan a JavaScript

Colección de Koder que pertenezcan a Python

\*/

const kodersCollection = [

{

name: 'Javier',

lastName: 'López',

birthday: '1996/06/24',

generation: **9**,

bootcamp: 'JavaScript',

scores: [

{

module: 'Frontend',

score: **90**

},

{

module: 'Backend',

score: **80**

},

{

module: 'Cloud',

score: **80**

},

]

},

{

name: 'Ale',

lastName: 'Paez',

birthday: '1998/05/12',

generation: **1**,

bootcamp: 'Python',

scores: [

{

module: 'Frontend',

score: **90**

},

{

module: 'Backend',

score: **100**

},

{

module: 'Cloud',

score: **100**

},

]

},

{

name: 'Juan Pablo',

lastName: 'Sánchez',

birthday: '1994/10/26',

generation: **10**,

bootcamp: 'JavaScript',

scores: [

{

module: 'Frontend',

score: **100**

},

{

module: 'Backend',

score: **95**

},

{

module: 'Cloud',

score: **80**

},

]

},

{

name: 'Oscar',

lastName: 'Ruiz',

birthday: '1996/06/24',

generation: **1**,

bootcamp: 'Python',

scores: [

{

module: 'Frontend',

score: **98**

},

{

module: 'Backend',

score: **100**

},

{

module: 'Cloud',

score: **70**

},

]

},

{

name: 'Javier',

lastName: 'López',

birthday: '1996/06/24',

generation: **9**,

bootcamp: 'JavaScript',

scores: [

{

module: 'Frontend',

score: **80**

},

{

module: 'Backend',

score: **90**

},

{

module: 'Cloud',

score: **100**

},

]

}

]

//Declaracion

function Koder(name, lastName, birthday, generation, bootcamp, scores){

this.name = name;

this.lastName = lastName;

this.birthday = birthday;

this.generation = generation;

this.bootcamp = bootcamp;

this.scores = scores;

this.saludar = function () {

console.log(`Hola mi nombre es ${this.name} y soy de la generacion ${this.generation} de ${this.bootcamp}`)

}

this.saludo =this.saludar()

this.hablar = function (alMentor, pregunta) {

console.log(`${alMentor} tengo duda sobre ${pregunta}`)

}

this.hola = this.hablar()

this.dameEdad = function (){

let hoy = new Date();

let birthday = new Date (this.birthday)

let Edad = hoy.getFullYear() - birthday.getFullYear()

console.log("tengo ", Edad," Años")

**return** Edad

}

this.edad = this.dameEdad()

this.damePromedio = function(){

const sumaScore = this.scores.reduce((acumulador, current)=>{

**return** acumulador += current.score

},**0**)

const promedio = Math.round(sumaScore / this.scores.length)

console.log("promedio", promedio ,"%")

**return** promedio

}

this.promedio = this.damePromedio()

}

const koderJonathan = new Koder("jonathan", "Reyes Alatorre", "10/09/1986", **11**, "JavaScript", [{module: 'Frontend',score: **98**},{module: 'Backend',score: **100**},{module: 'Cloud',score: **70**}] )

console.log(koderJonathan)

// Instanciar

console.log(koderJonathan.hablar( "Israel", "que es un array"))

console.log(".forEach")

const kodersArray = []

kodersCollection.forEach( (current, index) => {

let koder = new Koder(current.name , current.lastName, current.birthday, current.generation, current.bootcamp, current.scores )

kodersArray.push(koder)

})

console.log(kodersArray)

console.log(".map")

const Koders = kodersCollection.map((current, index)=>new Koder( current.name , current.lastName, current.generation , current.birthday, current.bootcamp, current.scores ))

console.log(Koders)

console.log(".filter")

const koderJS = kodersCollection.filter((current)=>{

**return** current.bootcamp === 'JavaScript'

})

console.log(koderJS)

const KoderPython = kodersCollection.filter( (current) => current.bootcamp === 'Python' )

console.log(KoderPython)

//Otras Opciones

console.log(" ")

let nuevoArrayKoders = kodersCollection.map((koder)=>{

**return** new Koder(koder.name , koder.lastName, koder.birthday, koder.generation, koder.bootcamp, koder.scores )

})

console.log(nuevoArrayKoders)

let KoderJS = nuevoArrayKoders.filter ( koder => koder.bootcamp === "JavaScript" )

console.log(KoderJS)

let koderPython = nuevoArrayKoders.filter ( koder => koder.bootcamp === "Python" )

console.log(koderPython)

/\*

Practica **12**

Ronda **3**

Del siguiente arreglo llamada 'arraysNames':

Obtener de cada nombre la siguiente salida:

Ruben Flores R.B.

Andre Vazquez A.V.

Fernando Palacios F.P.

George Camarillo G.C.

Ale Paez A.P.

\*/

const arraysNames = [

["Ruben", "Flores"],

["Andre", "Vazquez"],

["Fernando", "Palacios"],

["George", "Camarillo"],

["Ale", "Paez"],

["Shari", "Andrade"],

["Erick", "Trujillo"],

["Arce", "Gutiérrez"],

["Juan", "Hernandez"]

]

function dameIniciales (array){

const separarNombres = array.split(' ')

console.log(separarNombres)

inicial = ''

separarNombres.forEach((palabra) => {

// console.log(palabra)

inicial += palabra.charAt(**0**) + ". "

// console.log(inicial)

})

**return** inicial

}

nombre = "Jonathan Reyes Alatorre"

//console.log(dameIniciales(nombre))

//arraysNames.forEach((array)=>console.log(array.join(' ')))

//arraysNames.forEach((array)=>console.log(dameIniciales(array.join(' '))))

arraysNames.forEach((array)=>console.log(array.join(' ')+' '+dameIniciales(array.join(' '))))

/\*

. Del siguiente arreglo

Crear un array de objetos de tipo koder que se encuentranen el array arrayKoders, Quedando de la siguiente manera:

newArrayKoders = [

{

name: "Ruben",

lasName: "Alvarez",

Generation: **1**,

bootcamp: "Python",

score: **90**

},

{

name: "Andre",

lasName: "Vazquez",

Generation: **1**,

bootcamp: "Python",

score: **90**

},

.

.

.

]

Una vez tenirndo el array de newArrayKoders obtener los siguienrtes datos:

-Obrtener los koders que se encuentren en el Bootcamp de Python

-Obrtener los koders que se encuentren en el Bootcamp de JS

-Obtener el promedio general de todos los kodersregistrados

-DOM->Pintar los koders en una tabla (Utilizando los metodos para manipular el DOM)

\*/

arraysKoders =[

[

["name","Ruben"],

["lasName","Alvarez"],

["Generation",**1**],

["bootcamp","Python"],

["score",**90**],

],

[

["name","Andre"],

["lasName","Alvarez"],

["Generation",**1**],

["bootcamp","Python"],

["score",**90**],

],

[

["name","Ale"],

["lasName","Paez"],

["Generation",**1**],

["bootcamp","Python"],

["score",**98**],

],

[

["name","Juan pi"],

["lasName","Lopez"],

["Generation",**10**],

["bootcamp","Javascript"],

["score",**95**],

],

[

["name","Cintia"],

["lasName","Gomez"],

["Generation",**10**],

["bootcamp","Javascript"],

["score",**85**],

],

[

["name","Shari"],

["lasName","Andrade"],

["Generation",**10**],

["bootcamp","Javascript"],

["score",**90**],

],

[

["name","Arce"],

["lasName","Gutierrez"],

["Generation",**10**],

["bootcamp","Javascript"],

["score",**90**],

],

[

["name","JC"],

["lasName","Hernandez"],

["Generation",**10**],

["bootcamp","Javascript"],

["score",**90**],

],

[

["name","Erick"],

["lasName","Trujillo"],

["Generation",**10**],

["bootcamp","Javascript"],

["score",**90**],

]

]

function koder(name, lasName, Generation, bootcamp, score){

this.name = name;

this.lasName = lasName;

this.Generation = Generation;

this.bootcamp = bootcamp;

this.score = score;

}

const nuevoArray = arraysKoders.map((propiedadesArray)=>{

const objetoKoder = propiedadesArray.reduce((objeto,propiedadesDelKoder)=>{

const llave = propiedadesDelKoder[**0**]

const valor = propiedadesDelKoder[**1**]

objeto[llave] = valor

**return** objeto

}, {})

**return** new koder(objetoKoder.name, objetoKoder.lasName, objetoKoder.Generation, objetoKoder.bootcamp, objetoKoder.score)

})

console.log(nuevoArray)

const Prueba = new koder("Jonathan", "Reyes", **11**, "JavaScript", **90**)

//console.log(Prueba)

let KoderJS = nuevoArray.filter ( koder => koder.bootcamp === "Javascript" )

console.log(KoderJS)

let koderPython = nuevoArray.filter ( koder => koder.bootcamp === "Python" )

console.log(koderPython)

const sumaScore = nuevoArray.reduce((acumulador, current)=>{

console.log(acumulador)

**return** acumulador += current.score

},**0**)

const promedio = Math.round(sumaScore / nuevoArray.length)

console.log("Promedio general de todos los koders registrados", promedio ,"%")

/\*

Practica 13

Ronda **1**

realizar una función que reciba como parámetro el nombre completo de un koder y retorne sus iniciales.

Ejemplo:

input: 'Fernanda Palacios Vera'

Output: f, p, v.

\*/

//forma **1**

var name = prompt("forma 1: dime un nombre completo")

let divideName = name.split(" ")

let suma=[]

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

let word = divideName[i]

let initial = word.charAt(**0**)+". "

suma = suma + initial

}

console.log(suma)

console.log(" ")

//forma **2**

const getFirstLetter = text => {

let newArr = text.split(" ");

let newText = "";

newArr.map(item => newText += console.log(item.charAt(**0**)+". "))

**return** newText

}

getFirstLetter(prompt("forma 2: dime un nombre completo"))

console.log(" ")

//forma **3**

function getInitial(name){

const namesplit = name.split(" ")

console.log(namesplit)

let initials = ''

namesplit.forEach((word)=>{

console.log(word.charAt(**0**))

initials += word.charAt(**0**) + '. '

})

**return** initials

}

getInitial(prompt("forma 3: dime un nombre completo"))

console.log(getInitial("Jonathan Reyes Alatorre"))

console.log(" ")

//forma **4**

function getInitial0(name0){

const namesplit0 = name0.split(" ")

console.log(namesplit0)

let initials0 = []

namesplit0.forEach((word)=>{

console.log(word.charAt(**0**))

initials0.push(word.charAt(**0**)+" ")

})

**return** initials0.join()

}

getInitial0(prompt("forma 4: dime un nombre completo"))

console.log(getInitial0("Jonathan Reyes Alatorre"))

console.log(" ")

/\*

Realiza una funcion que reciba n parametros de tipo string y y validar por cada string cuantas vocales tiene y retornal el resultado en un objeto.

Ejemplo:

input:

string1: 'Hola Mundo'

srting2: 'soy fullstack'

.

.

.

Output:

{

Hola Mundo: **4**,

soy fullstack:**3**

}

\*/

/\*let text0 = "The rain in SPAIN stays mainly in the plain";

let result = text0.match(/ain/g);

console.log(text0)

console.log(result)\*/

//forma **1** incompleta

const getVocal = text => {

let result = {

[text]: text.match(/[aeiou]/gi).length

}

**return** result;

}

console.log(getVocal("Hola Mundo"))

console.log(getVocal("soy fullstack"))

console.log(getVocal(prompt("Digita un texto con vocales")))

console.log(" ")

//forma **2**

//...string genera un arreglo por cada parametro

function getQuantityVowels(...string) {

console.log("string",string)

const object = []

string.forEach((str)=>{

console.log("str",str)

const quantity = str.match(/[aáeéiíoóuú]/gi).length

console.log("quantity",quantity)

object[str] = quantity

})

**return** object

}

console.log(getQuantityVowels(prompt("Digita un texto")))

console.log(getQuantityVowels("Hola Mundo", "soy fullstack" ))

console.log(getQuantityVowels("Jonathan", "Reyes", "Alatorre" ))

console.log(" ")