Clase 1 Dart

void main() {

/\*

//final String nombre = "Tony";

//const apellido = "Stark";

//nombre = "Petter";

print("$nombre $apellido");\*/

//Numeros

int empleados = 10;

double salario = 1856.25;

print("El numero de empleados es: $empleados");

print("El salario de los empleados es: $salario");

//print(salario) para cuando solo es la variable y ya sin texto y sin comillas

bool? isActive = null; //la ? es para poder asignarle el valor null a cualquier variable.

//tmb pde ser bool? isActive; su valor seguiria siendo null;

if(isActive == null){

print("isActive esta a null");

} else if(isActive == true) {

print("isActive esta activo");

} else {

print("isActive esta inactivo");

}

}

Clase 2:

void main() {

//List numeros = [1,2,3,4,5,6,7,8,9,10];

//numeros.add(11);

//numeros.add("Galadriel"); si no especificamos tipo se pueden meter todos los tipos posibles en la lista

//Si especificamos el tipo ya solo se puede meter de ese mismo y no de otro;

/\*List<int> numeros = [1,2,3,4,5,6,7,8,9,10];

print(numeros);\*/

/\*final masNumeros = List.generate(100, (int index) => index+1); index representa las posiciones dentro de la lista.

print (masNumeros);\*/

/\*

Map persona = {

"nombre" : "Pedro",

"edad" : 35,

"soltero" : false,

};

persona.addAll({"apellido": "Chaves"});

print(persona);

print(persona["nombre"]);\*/

/\*Map<String, String> persona = {

"nombre" : "Pedro",

"edad" : "35",

"soltero" : "false",

};

print(persona);\*/

/\*Map<String, dynamic> persona = {

"nombre" : "Pedro",

"edad" : 35,

"soltero" : false,

};

print(persona);\*/

}

void main() {

saludar("Paco");

}

//Crear una funcion llamada saludar

//Le pasamos un string con un nombre

void saludar(String nombre){

print("Hola $nombre");

}

void main() {

final serie = "The Continental";

final productora = "Amazon";

registro(serie: serie, productora:productora);

}

//Crear una funcion llamada saludar

//Le pasamos un string con un nombre

/\*void registro([String nombre = "Desconocido"]){ // mediante los corchetes le asignamos un valor por defecto a la variable nombre por si no se introduce en el main

print("La serie se llama $nombre");

}\*/

/\*

void registro(String serie, {String? productora}){ // mediante las llaves y la interrogación ponemos un parámetro opcional el cual si no se le asigna valor es null

print("La serie se llama $serie");

print("La serie se llama $productora");

}\*/

void registro({required String serie, required String productora}){ // El tipo required obliga a poner esos parámetros al llamar la función, al llamarla se usa (parámetro de la funcion: variable del main)

print("La serie se llama $serie");

print("La productora se llama $productora");

}

//Constructor con nombre

void main() {

final spiderman = new Heroe(nombre: "Peter", poder: "Sentido aracnido");

spiderman.nombre = "Petter";

spiderman.poder = "sentido aracnido";

//print(spiderman.nombre);

//print(spiderman.poder);

print(spiderman);

}

class Heroe{

String nombre;

String poder;

Heroe({//si no usamos en el constructor el required tenemos que iniciar las variables con posibilidad de null (?)

required this.nombre,

required this.poder

});

@override

String toString(){

return "Nombre del heroe ${this.nombre}, poder ${this.poder}";

}

}

void main() {

final rawJson = {

"nombre": "Tony Stark",

"poder": "dinero"

};

}

class Heroe{

String? nombre;

String? poder;

Heroe({//si no usamos en el constructor el required tenemos que iniciar las variables con posibilidad de null (?)

required this.nombre,

required this.poder

});

Heroe.fromJson(Map<String, String> json){

this.nombre = json["nombre"]!;//la exclamacion dice que no va a asignarle valor nulo..(Confia en mi)

this.poder = json["poder"]!;

}

@override

String toString(){

return "Nombre del heroe ${this.nombre}, poder ${this.poder}";

}

}

void main() {

final rawJson = {

"nombre": "Tony Stark",

// "poder": "dinero"

};

final ironman = new Heroe.fromJson(rawJson);

print(ironman);

}

class Heroe{

String nombre;

String poder;

Heroe({//si no usamos en el constructor el required tenemos que iniciar las variables con posibilidad de null (?)

required this.nombre,

required this.poder

});

Heroe.fromJson(Map<String, String> json):

this.nombre = json["nombre"]?? "Desconocido",//la doble interrogación hace que si el valor de las variables es null le inserta el valor por defecto puesto después.

this.poder = json["poder"]??"No tiene poder";

@override

String toString(){

return "Nombre del heroe ${this.nombre}, poder ${this.poder}";

}

}

import "dart:math";

void main() {

final cuadrado = new Cuadrado(5);

print("Area: ${cuadrado.calculaArea()}");

print("Area get: ${cuadrado.area}");

print("Valor del lado: ${cuadrado.lado}");

}

class Cuadrado{

double lado;

//area= l\*l

Cuadrado (double lado):

this.lado = lado;

double get area{

return this.lado \* this.lado;

}

set setArea(double valor){

this.lado = sqrt(valor);

}

double calculaArea(){

return this.lado \* this.lado;

}

}

void main() {

final perro = new Perro();

final pajaro = new Pajaro();

//perro.emitirSonido();

sonidoAnimal(perro);

sonidoAnimal(pajaro);

}

void sonidoAnimal(Animal animal){

animal.emitirSonido();

}

abstract class Animal{

int? patas;

void emitirSonido();

}

class Perro implements Animal{

int?patas;

void emitirSonido(){

print("Guau");

}

}

class Pajaro implements Animal{

int?patas;

int?alas;

void emitirSonido(){

print("Pio");

}

}

void main() {

final tenis = new Tenis(false);

}

abstract class Deportes{

int? nJugadores;

}

abstract class DeportesDeBalon implements Deportes{

bool? balon;

}

abstract class DeportesDeRaqueta implements Deportes{

int? puntos;

}

class Tenis implements DeportesDeRaqueta{

int? puntos;

bool? saque;

int? nJugadores;

Tenis(bool saque){

this.saque = saque;

sacar(saque);

}

void sacar(bool saque){

if(saque){

print("Se ha sacado");

} else {

print("No saca");

}

}

}

class Futbol implements DeportesDeBalon{

int? nJugadores;

int? goles;

bool? balon;

}

class Baloncesto implements DeportesDeBalon{

int? nJugadores;

bool? balon;

int? puntos;

}

void main() {

final batman = Heroe ("Bruce Wane");

final joker = Villano ("Jack Osdwall");

print(batman);

print(joker);

print(joker.locura);

}

abstract class Personaje{

String nombre;

String? poder;

Personaje(this.nombre);

@override

String toString(){

return "$nombre - $poder";

}

}

class Heroe extends Personaje{

Heroe(String nombre): super(nombre);

}

class Villano extends Personaje{

Villano(String nombre): super(nombre);

int locura = 95;

}