

***APUNTES DE FLUTTER***

**-------------------------------------------------------------------------------------------**

**ACCEDER A COMANDOS DE FLUTTER:**

Ctrl + Shift + P

**SI EN EL PROYECTO HAY ERRORES:**

1. -flutter clean-
2. -flutter pub get-

**CARGAR PROYECTO (APLICACIÓN) DE FLUTTER:**

-flutter doctor-

**CARGAR EMULADOR LOCAL:**

1. Acceder a los comandos.
2. Flutter: Launch Emulator
3. Pixel 4a API 34

**MOSTRAR NAVEGADORES DETECTADOS POR FLUTTER:**

-flutter doctor -v-

**CARGAR EMULADOR EN FLUTTER WEB:**

1. -flutter config --enable-web-
2. -flutter devices-
3. -flutter run -d chrome- **//** -flutter run -d edge- **//** -flutter run -d firefox- **//** -flutter run -d safari-
4. Esperar a que cargue la aplicación en una nueva pestaña de google:
   1. Abre DevTools (Ctrl+Shift+I o clic derecho > "Inspeccionar").
   2. Haz clic en el icono de Toggle Device Toolbar (parece un ordenador).
   3. En “Dimensions Responsive” ajusta el **ancho a 393px** y el **alto a 0 “851px” (Pixel 4a API 34)**.
   4. Se puede separar el panel de opciones en otra ventana con:

**Ctrl + Shift +D** para que no moleste a la vista principal.

**-------------------------------------------------------------------------------------------**

**APUNTES DE OPENWEBINARS DE DART**

**CÓMO HACER UN MAIN Y TIPOS DE VARIABLES:**

void main(List<String> args) {

var varNombre = "Un nombre";

print("hola mundo");

print(varNombre);

String nombreDos = "Messi";

String nombreTres = "Cristiano";

// CONCATENACIÓN DE STRINGS

print("$nombreDos" + " y " + "$nombreTres");

//UPPERCASE

print(nombreDos.toUpperCase());

//CARACTER DE VARIABLE

print(nombreDos[0]);

print(nombreDos[1].toUpperCase());

//LONGITUD DE LA CADENA

print(nombreDos.length);

// CONTAIN

String containFrase = "Esto es una línea";

print(containFrase.contains("Esto")); //true

print(containFrase.contains("Esta")); //false

print(containFrase.contains("ESTO")); //false

}

**VARIABLES INT Y DOUBLE:**

import 'dart:math' as math;

void main() {

//Sumar números enteros

int numero1 = 3;

int numero2 = 9;

int suma = numero1 + numero2;

print(suma);

//Retornar el valor suma como string

String suma2 = suma.toString();

print(suma2);

//Inicializar un número como si fuera un String y transformarlo en un Int

String strNumero = "5";

int strNumero2 = int.parse(strNumero);

print(strNumero2);

//Importar el paquete math de Dart: import 'dart:math' as math

//Calcular el perímetro de una circunferencia: Perimetro = 2\*PI\*r

int radioCir = 2;

double perimetroCir = 2 \* math.pi \* radioCir;

print(perimetroCir);

}

**BOOLEANOS Y CONDICIONAL “IF”:**

void main() {

bool encendido = false;

bool apagado = !encendido;

bool isAdmin = true;

bool estudiante = true;

print(apagado);

if (encendido == true) {

print("La luz está encendida");

} else {

print("¡Estamos a oscuras!");

}

if (estudiante == true && isAdmin == true) {

print("Puedes estudiar y crear cualquier curso");

} else if (estudiante == true && isAdmin == false) {

print("Sólo tienes acceso de lectura");

} else {

print("Suscríbete para tener acceso a los cursos");

}

}

**LISTAS (ARRAYLISTS):**

void main() {

// TIPOS DE LISTA

List<dynamic> listDynamic = ["hola", 2, 2.5, true];

List<int> listInt = [1, 2, 3];

List<double> listDouble = [1.0, 2.0, 3.9];

List<String> palabras = ["esta", "es", "una", "lista", "de", "Strings"];

// PINTA LA LISTA

print(listDynamic);

print(palabras);

print(listInt.first); // PINTA EL PRIMER ELEMENTO DE LA LISTA

print(listInt.last); // PINTA EL ÚLTIMO ELEMENTO DE LA LISTA

List<String> palabras2 = ["otras", "palabras"];

palabras.addAll(palabras2); // AÑADE LA LISTA "palabras2" AL FINAL DE LA LISTA "palabras"

print(palabras);

// AÑADE AL FINAL DE LA LISTA "palabras" UN NUEVO ELEMENTO

palabras.add("elemento 1");

palabras.add("elemento 2");

palabras.add("elemento 3");

print(palabras);

// ELIMINA EN LA LISTA "palabras" EL ELEMENTO INTRODUCIDO (SI EXISTE)

palabras.remove("esta");

print(palabras);

// ELIMINA DE LA LISTA "palabras" LOS ELEMENTOS DESDE EL "4" HASTA EL "FINAL"

palabras.removeRange(4, palabras.length);

print(palabras);

}

**SETS:**

void main() {

List<String> palabras = ["Bienvenido", "a", "Dart"];

palabras.add("Flutter");

palabras.add("Flutter");

palabras.add("Flutter");

palabras.add("Flutter");

palabras.add("Flutter");

palabras.add("Flutter");

// SE AÑADEN TODOS LOS "Flutter"

print(palabras);

// CREAMOS UN SET QUE TIENE EL VALOR DE LA LISTA "palabras"

Set<String> palabras2 = palabras.toSet();

// PINTA EL SET "palabras2" como un LIST

print(palabras2.toList());

Set<String> palabras3 = {"Bienvenido", "a", "Flutter"};

palabras3.add("Dart");

palabras3.add("Dart");

palabras3.add("Dart");

palabras3.add("Dart");

palabras3.add("Dart");

// SE AÑADE "Dart" UNA VEZ PORQUE LOS SETS NO POSEEN VALORES REPETIDOS

print(palabras3);

}

**MAPS:**

void main() {

Map<dynamic, dynamic> personas = {

1: "Pedro",

2: "Maria",

3: "Lucia",

"nombre": "Open",

10: "Webinars",

"is@dmin": true,

};

Map<int, int> personas2 = {2: 10000, 12: 50000};

// SE SOBREESCRIBE EL VALOR "2" DEL MAP "personas" POR EL DEL MAP "personas2" Y SE AÑADE EL "12"

personas.addAll(personas2);

print(personas);

// SE PUEDE INTRODUCIR COMO VALORES: LISTS, SETS Y MAPS

Map<String, dynamic> aficiones = {

"deportes": ["crossfit", "piraguismo"],

"pintar": true,

"series": 10

};

Map<String, dynamic> jose = {"nombre": "jose", "aficiones": aficiones};

// PINTA DEL MAP "jose" LOS DEPORTES DEL MAP "aficiones" (QUE ESTÁ DENTRO DE "jose")

print(jose["aficiones"]["deportes"]);

}

**SCANNER DE DART Y PRINT EN TERMINAL:**

import 'dart:io';

void main() {

// PARA IMPRIMIR UNA FRASE EN LA TERMINAL (PRINT EN TERMINAL)

stdout.writeln('¡Hola! ¿Cuál es tu nombre?');

// SCANNER DE DART (LEER POR TECLADO)

String? nombre = stdin.readLineSync();

stdout.writeln('¡Encantado $nombre! ¡Bienvenido a OpenWebinars!');

}

**FUNCIONES CON ARGUMENTOS:**

import 'dart:math';

void main() {

String nombre = "Alfonso María";

String mensaje = "¡Ya no tengo que limpiar retretes!";

int random = Random().nextInt(8);

saludar(mensaje, nombre);

saludar2(nombre: nombre, mensaje: mensaje);

saludar3(mensaje: mensaje, veces: random);

}

// PARA INDICAR QUÉ HACE UNA VARIABLE EN LOS ARGUMENTOS SE ESCRIBE UN AVISO ENTRE CORCHETES

void saludar(String mensaje, [String nombre = '<insertar nombre>']) {

print('MENSAJE: $mensaje\nNOMBRE: $nombre');

}

// PARA INDICAR QUE UN PARÁMETRO ES OBLIGATORIO EN UN ARGUMENTO SE LE AÑADE "required" ANTES Y SE METE ENTRE LLAVES

void saludar2({required String nombre, required String mensaje}) {

print('MENSAJE: $mensaje\nNOMBRE: $nombre');

}

// PARA INDICAR QUE UN PARÁMETRO ES OBLIGATORIO PERO PUEDE SER NULO SE LE AÑADE "?" AL FINAL DEL TIPO.

void saludar3({String? nombre, required String mensaje, required int veces}) {

for (var i = 0; i < veces; i++) {

print('MENSAJE: $mensaje\nNOMBRE: $nombre');

}

}

**ENUM:**

enum Audio { BAJO, MEDIO, ALTO }

main() {

Audio volumen = Audio.ALTO;

switch (volumen) {

case Audio.BAJO:

print('No escucho nada.');

break;

case Audio.MEDIO:

print('Decibelios constantes.');

break;

case Audio.ALTO:

print('¿Estás sordo?');

break;

}

}

**FUTURE / PROMISE:**

void main() {

// SIRVE PARA EJECUTAR LA FUNCIÓN PASADO UN TIEMPO (INICADO EN EL ARGUMENTO)

Future<String> timeOut = Future.delayed(Duration(seconds: 3), () {

print('Cuando pasen 3 segundos...');

return '¡Aquí está!';

});

timeOut.then(print); //simplificación de la funcion

print('Fin del código');

}

**FUTURE CON ASYNC-AWAIT:**

import 'dart:io';

// SE ESCRIBE "async" PORQUE HAY UN "await"

void main() async {

String path = Directory.current.path + '/06\_otros\_tipos/assets/cursos.txt';

// SE ESCRIBE "await" PARA INDICAR QUE SE EJECUTE CUANDO SE HAYA REALIZADO LEER ARCHIVO

String texto = await leerArchivo(path);

print(texto);

print('Fin del main');

}

Future<String> leerArchivo(String path) async {

File file = new File(path);

return file.readAsString();

}

**CATCH-ERROR:**

void main() {

Future<String> timeOut = Future.delayed(Duration(seconds: 5), () {

if (1 == 1) {

throw '¡Corre lo mas rápido que puedas!';

}

return 'Retorno del future';

});

// EVITA QUE SE LANCE EL ERROR

timeOut.then(print).catchError((error) => print(error));

print('Fin del main');

}

flutter run -d chrome --web-renderer html