Hola mundo

|  |
| --- |
| void main() { |
|  |  |
|  | /\* |
|  | print('Hola Mundo'); |
|  | asdasd |
|  | asd |
|  | a |
|  | sdasd |
|  | \*/ |
|  |  |
|  | String nombre; |
|  |  |
|  | print('Hola $nombre'); |
|  |  |
|  | } |

Tipos de datos

|  |
| --- |
| void main() { |
|  |  |
|  | // Números |
|  | int empleados = 10; |
|  | double pi = 3.141592; |
|  | var numero = 1.0; |
|  |  |
|  | print( '$empleados - $pi - $numero' ); |
|  |  |
|  | // String - Cadenas de caracteres |
|  | String nombre = 'Tony'; |
|  | print(nombre); |
|  | print(nombre[0]); |
|  | print(nombre[ nombre.length - 1 ]); |
|  |  |
|  | } |

|  |
| --- |
| void main() { |
|  |  |
|  |  |
|  | bool activado = true; |
|  | print(activado); |
|  |  |
|  |  |
|  | if ( !activado ) { |
|  | print('El motor esta funcionando'); |
|  | } else { |
|  | print('Está apagado'); |
|  | } |
|  |  |
|  |  |
|  |  |
|  | } |

Listas

|  |
| --- |
| void main() { |
|  |  |
|  |  |
|  | List<int> numeros = [1,2,3,4,5]; |
|  | print( numeros ); |
|  |  |
|  | numeros.add(6); |
|  | print( numeros ); |
|  |  |
|  |  |
|  | // Tamaño fijo |
|  | List masNumeros = List(10); |
|  | print( masNumeros ); |
|  | // masNumeros.add(1); Esto no es correcto |
|  | masNumeros[0] = 1; |
|  | print( masNumeros ); |
|  |  |
|  | } |

Map

|  |
| --- |
| void main() { |
|  |  |
|  | String propiedad = 'soltero'; |
|  |  |
|  |  |
|  | Map<String, dynamic> persona = { |
|  | 'nombre' : 'Carlos', |
|  | 'edad' : 32, |
|  | 'soltero': true |
|  | }; |
|  |  |
|  | print( persona['nombre'] ); |
|  | print( persona['edad'] ); |
|  | print( persona[ propiedad ] ); |
|  |  |
|  |  |
|  | Map<int, String> personas = { |
|  | 1: 'Tony', |
|  | 2: 'Peter', |
|  | 9: 'Strange' |
|  | }; |
|  |  |
|  | personas.addAll( { 4: 'Banner' } ); |
|  |  |
|  | print( personas ); |
|  | print( personas[2] ); |
|  |  |
|  |  |
|  | } |

Funciones

|  |  |
| --- | --- |
|  | void main() { |
|  |  |
|  | String mensaje = saludar2( nombre: 'Fernando', texto: 'Hola,' ); |
|  |  |
|  | print(mensaje); |
|  |  |
|  | } |
|  |  |
|  |  |
|  | String saludar({ String texto, String nombre }) { |
|  | // print('Hola'); |
|  | return '$texto $nombre'; |
|  | } |
|  |  |
|  | String saludar2({ String texto, String nombre }) => '$texto $nombre'; |

Clases

|  |
| --- |
| void main() { |
|  |  |
|  | final wolverine = new Heroe( |
|  | poder: 'Regeneración', |
|  | nombre: 'Logan' |
|  | ); |
|  |  |
|  | print( wolverine ); |
|  |  |
|  | } |
|  |  |
|  | class Heroe { |
|  |  |
|  | String nombre; |
|  | String poder; |
|  |  |
|  | // Heroe({ String nombre = 'Sin Nombre', String poder }) { |
|  | // this.nombre = nombre; |
|  | // this.poder = poder; |
|  | // } |
|  |  |
|  | Heroe({ this.nombre, this.poder }); |
|  |  |
|  | String toString() => 'nombre: $nombre - poder: $poder'; |
|  |  |
|  | } |

Constructores con nombre

|  |
| --- |
|  |
| import 'dart:convert'; |
|  |  |
|  | void main() { |
|  |  |
|  | // final wolverine = new Heroe('Logan', 'Regeneración'); |
|  | final rawJson = '{ "nombre": "Logan", "poder":"Regeneración" }'; |
|  | Map parsedJson = json.decode( rawJson ); |
|  |  |
|  | // print( parsedJson ); |
|  |  |
|  | final wolverine = new Heroe.fromJson( parsedJson ); |
|  |  |
|  | print(wolverine.nombre); |
|  | print(wolverine.poder); |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | class Heroe { |
|  |  |
|  | String nombre; |
|  | String poder; |
|  |  |
|  | Heroe( this.nombre, this.poder ); |
|  |  |
|  | Heroe.fromJson( Map parsedJson ) { |
|  | nombre = parsedJson['nombre']; |
|  | poder = parsedJson['poder']; |
|  |  |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |

Set y Get

|  |
| --- |
| void main() { |
|  |  |
|  | final cuadrado = new Cuadrado(); |
|  |  |
|  | cuadrado.lado = 10; |
|  |  |
|  |  |
|  | print( cuadrado ); |
|  | print( 'área: ${ cuadrado.area }' ); |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | class Cuadrado { |
|  |  |
|  | double \_lado; |
|  | // double \_area; |
|  |  |
|  |  |
|  | set lado( double valor ) { |
|  |  |
|  | if ( valor <= 0 ) { |
|  | throw('El lado no puede ser menor o igual a 0'); |
|  | } |
|  |  |
|  | \_lado = valor; |
|  |  |
|  | } |
|  |  |
|  |  |
|  | double get area => \_lado \* \_lado; |
|  |  |
|  |  |
|  | toString() => 'Lado: $\_lado'; |
|  |  |
|  | } |

Clase abstracta

|  |
| --- |
|  |
|  | final perro = new Perro(); |
|  | perro.emitirSonido(); |
|  |  |
|  | final gato = new Gato(); |
|  | gato.emitirSonido(); |
|  |  |
|  | } |
|  |  |
|  |  |
|  | abstract class Animal { |
|  |  |
|  | int patas; |
|  |  |
|  | void emitirSonido(); |
|  |  |
|  | } |
|  |  |
|  | class Perro implements Animal{ |
|  |  |
|  | int patas; |
|  | int colas; |
|  |  |
|  | void emitirSonido() => print('GUAUUUUU!!'); |
|  |  |
|  | } |
|  |  |
|  | class Gato implements Animal { |
|  |  |
|  | int patas; |
|  |  |
|  | void emitirSonido() => print('MIAUUUU!!'); |
|  |  |
|  | } |

Diferencia entre extends, implements y with

<https://stackoverflow.com/questions/55295782/extends-versus-implements-versus-with>

Extends

|  |
| --- |
| void main() { |
|  |  |
|  | final superman = new Heroe(); |
|  | superman.nombre = 'Clark Kent'; |
|  |  |
|  | final luthor = new Villano(); |
|  | luthor.nombre = 'Lex Luthor'; |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |
|  | abstract class Personaje { |
|  | String poder; |
|  | String nombre; |
|  | } |
|  |  |
|  |  |
|  | class Heroe extends Personaje { |
|  |  |
|  | int valentia; |
|  |  |
|  | } |
|  |  |
|  | class Villano extends Personaje { |
|  |  |
|  | int maldad; |
|  | } |

Mixins

|  |
| --- |
| abstract class Animal { } |
|  |  |
|  | abstract class Mamifero extends Animal { } |
|  |  |
|  | abstract class Ave extends Animal { } |
|  |  |
|  | abstract class Pez extends Animal { } |
|  |  |
|  |  |
|  | abstract class Volador { |
|  | void volar() => print('Estoy volando!!'); |
|  | } |
|  |  |
|  | abstract class Caminante { |
|  | void caminar() => print('Estoy caminando!!'); |
|  | } |
|  |  |
|  | abstract class Nadador { |
|  | void nadar() => print('Estoy nadando!!'); |
|  | } |
|  |  |
|  |  |
|  | class Delfin extends Mamifero with Nadador {} |
|  |  |
|  | class Murcielago extends Mamifero with Caminante, Volador {} |
|  |  |
|  | class Gato extends Mamifero with Caminante {} |
|  |  |
|  | class Paloma extends Ave with Caminante, Volador {} |
|  |  |
|  | class Pato extends Ave with Caminante, Volador, Nadador {} |
|  |  |
|  | class Tiburon extends Pez with Nadador {} |
|  |  |
|  | class PezVolador extends Pez with Nadador, Volador { } |
|  |  |
|  |  |
|  | void main() { |
|  |  |
|  | final pato = new Pato(); |
|  | pato.volar(); |
|  |  |
|  | final pezVolador = new PezVolador(); |
|  | pezVolador.nadar(); |
|  |  |
|  |  |
|  | } |

Explicacion de mixins

<https://medium.com/flutter-community/dart-what-are-mixins-3a72344011f3>

Futures

|  |
| --- |
| void main() { |
|  |  |
|  | print('Estamos a punto de pedir datos'); |
|  |  |
|  | httpGet('https://api.nasa.com/aliens').then( (data) { |
|  |  |
|  | print( data ); |
|  |  |
|  | }); |
|  |  |
|  |  |
|  | print('Ultima línea'); |
|  | } |
|  |  |
|  |  |
|  |  |
|  | Future<String> httpGet(String url) { |
|  |  |
|  | return Future.delayed( new Duration( seconds: 4 ), () { |
|  | return 'Hola Mundo'; |
|  | }); |
|  |  |
|  | } |

Async – Await

|  |
| --- |
| void main() async { |
|  |  |
|  | print('Estamos a punto de pedir datos'); |
|  |  |
|  | String data = await httpGet('https://api.nasa.com/aliens'); |
|  |  |
|  | print( data ); |
|  |  |
|  | print('Ultima línea'); |
|  | } |
|  |  |
|  |  |
|  |  |
|  | Future<String> httpGet(String url) { |
|  |  |
|  | return Future.delayed( new Duration( seconds: 4 ), () { |
|  | return 'Hola Mundo'; |
|  | }); |
|  |  |
|  | } |