FLUTTER I DART

Email: jordimassana@gmail.com

Como hacer un buen cliente de API para traer los datos-

**Firebase**: sistema de google de forma gratis: sistema de base de datos. Se guardan y extraen datos de forma authenticada (para chats, ..).

Flutter tambien sirve para web.

**Flutter** es framework, **Dart** es el lenguaje.   
framework: conjunto de herramientas para aconseguir un objectivo (herramientas: entre elles el lenguaje Dart).  
(.NET es framework de c#)

Flutter es para Android y IOS. (el nuevo os de google para movil es en flutter) -> habra continuacion de flutter en ppio.

Aplicaciones hibridas.

Apps se pueden hacer apps de forma nativa: (cada os tiene el suyo Linux ...otro paquete de codigo.)  
paquete de codigo para ios (ahora en lenguaje Swift) y  
otro paquete de codigo para android (ahora en lenguaje kotlin)

Flutter y React native son los dos frameworks que implementan elementos nativos a partir de la programacion de un solido codigo.

Flutter coge por ej appbar de flutter y en ios la pone con Swift y en kotlin para android.

Flutter se organiza en widgets:  
hay un container ( como un div en html) y los width height decoration y puede tenir container hijos.

CONTAINER(  
 width:300  
 height:300  
 decoration:  
 child: Text(“HOLA”))

Widgets natives hay muchos y muchas capes ej: Cupertino para IOS y Material para Android.

Redsys tiene plug in para android y ios, hay un bridge o algo asi que se llama para plugins en cada una de los os.

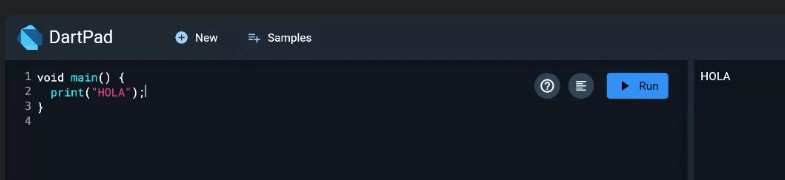
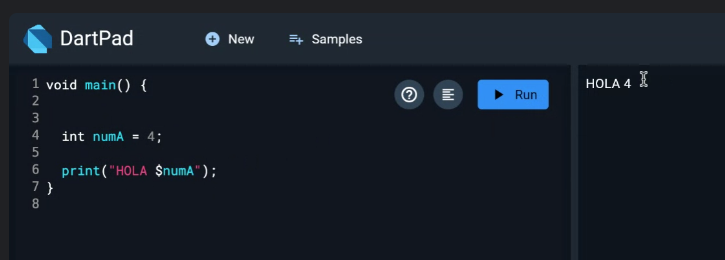
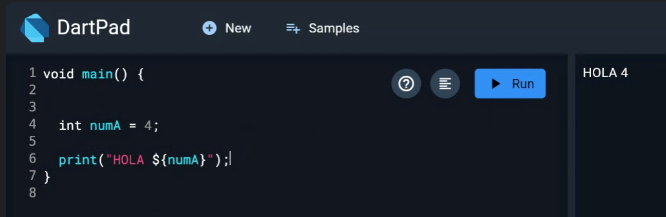
Todo es un widget en flutter. **Scaffold** (es como una pantall) es un widget que contiene barra tareas, cuerpo titulo, y una tapbar abajo de la pantalla. En flutter el primer widget serà un scaffold.   
Si queremos hacer un boton para reusar en varias pantalles serà un widget no scaffold.

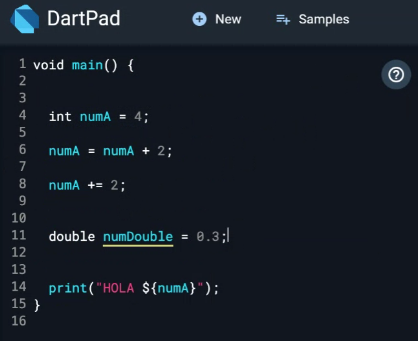
**Scaffold** tiene en **body** una variable y en **appbar** es otra zona.

Cómo se programa en Dart.

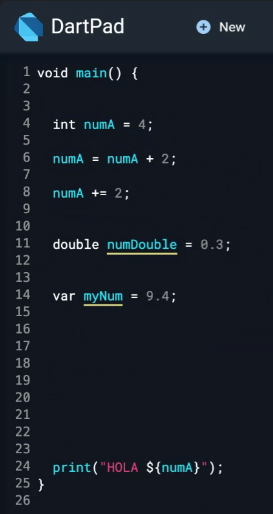
FRAMEWORK:  
-**Se usarà hay que instalar Android Studio. (tambien se puede hacer con Visual Studio Code)**-Hay que entrar en la web: DARTPAD. **-Hay que descargar flutter sdk**

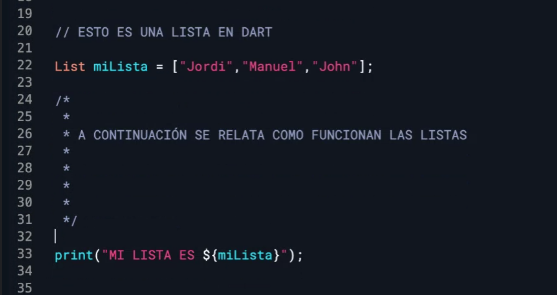
**Entramos en** [**https://dartpad.dev/**](https://dartpad.dev/) **mientras**

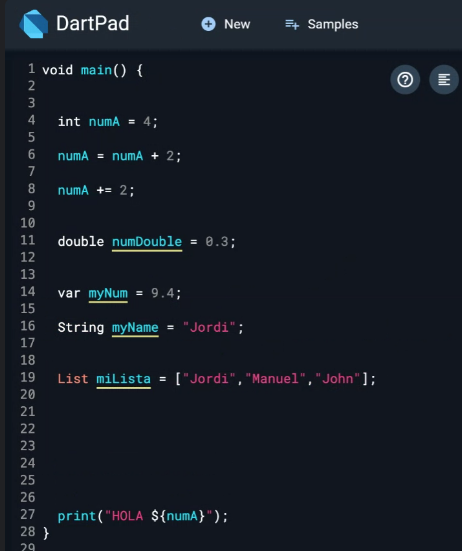
  
  
a veces hay que poner corchetes para hacer print de variable.

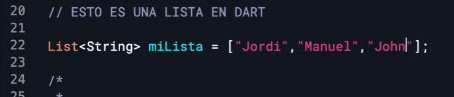
Float no existe en Dart sinó double.   


Fuertemente tipado puede ser Dart però podemos definir variables que se tipan en el momento de su asignación. Ej. Una variable puede ser de dos coses distintes, declarando con **var**.

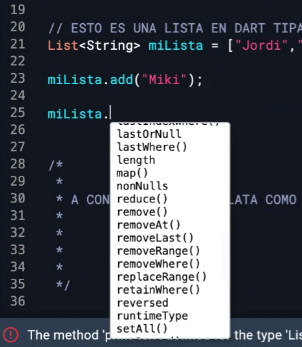


Tipos de variable: listas (cuando digamos arrays nos referimos a listas) string var double int   
comentarios con **//** en una linea:  
  
o de varias lineas **/\*\*\*\*/** varias lineas

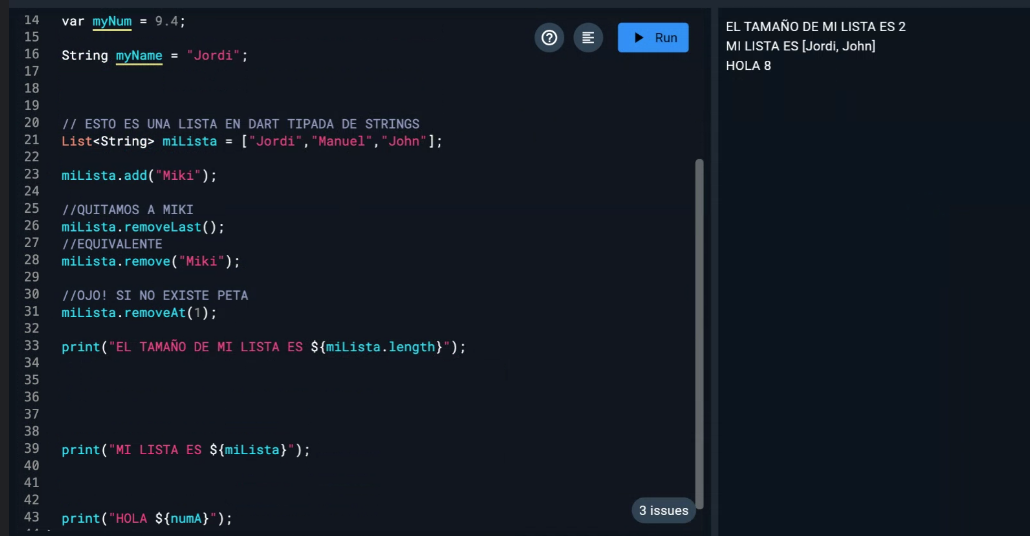


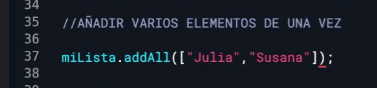
Lista string se declara como string una lista entera. Es muy peligroso mezclar tipus de datos en listas. Por eso tipamos las listas:   


Todas las listas son dinámicas y no hay que liberar memòria, no es lenguaje embedido.

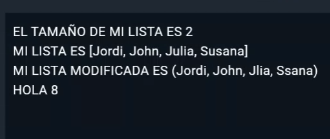
Listas tienen varias funciones (filtrar, iterar, añadir o quitar).  
  


  
el código peta si no existe el indice. El indice en la lista empieza por 0.



  
.add()  
.addAll()

Mapear una lista: convertir los elementos de una lista. Por ej en una lista de strings queremos eliminar la letra u de cada uno de esos elementos.

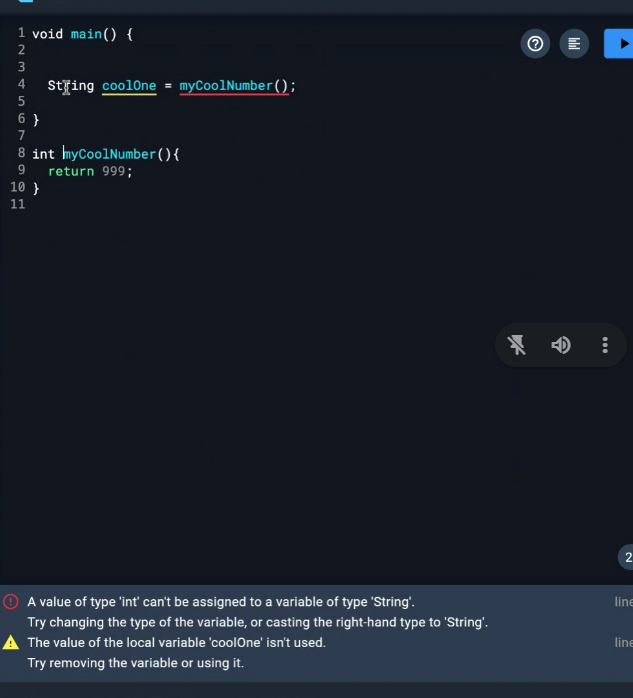
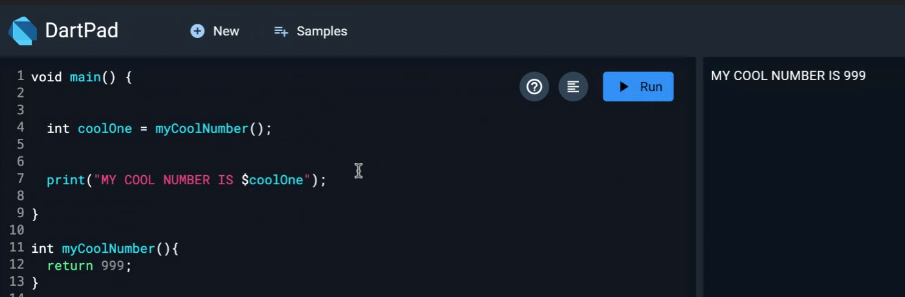
  
  
el map pasa por cada uno de los elementos y ejecuta una fx sobre cada uno de los elementos:  
quita las u por ejemplo.

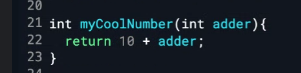
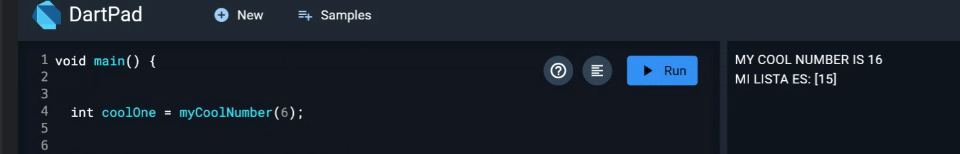
map es una funcion anònima.

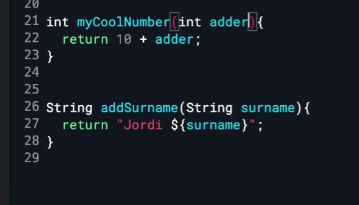
Funciones:

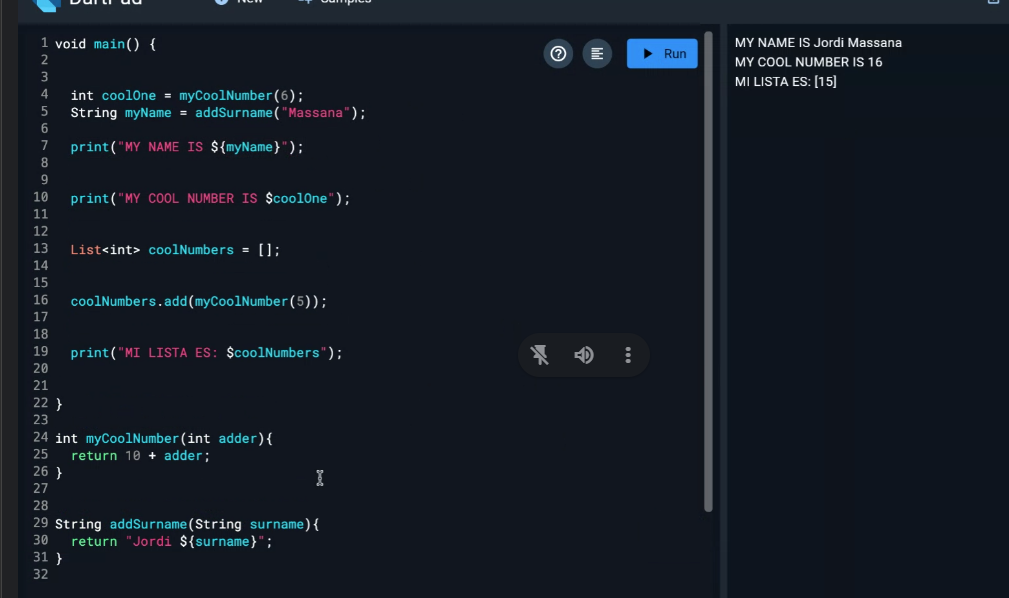
Una funcion tiene un parametro de retorno y un numero (mycoolnumber)  
llamamos a la funcion dentro de nuestro codigo void  

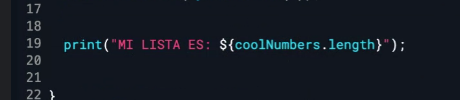

Mejor TIPARLO todo !   
  
tiene que ser int todo:

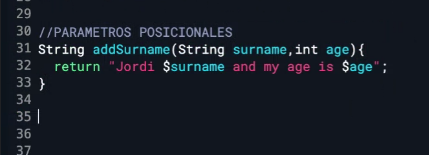
   


Pueden recibir parametros las funciones:  
  




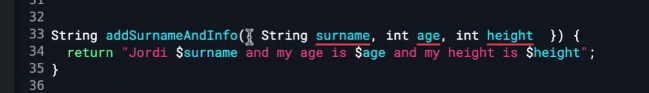
  
$ es interpolacion: cuando quieres poner una variable en una cadena de texto.

  
para llamar un metodo de la variable ... (osea cuando no es una unica palabra hay que poner las llaves).

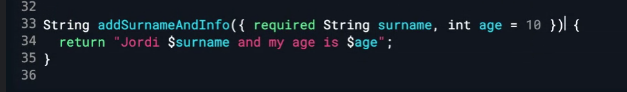
Parametros posicionales  


Parametros opcionales

(Llenguatjes **declaratius**: la funció ha de estar declarada abans de cridarla. Pero a Dart no és declaratiu, podem posar la funció on sigui. )

Sèrie de parametros que se ponen entre corchetes en una funcion son opcionales:   
  
  
Dart en la medida de lo possible intenta si hay un null da error:   
**Nullability\***: variable puede ser null.   
**Dart.safety**;

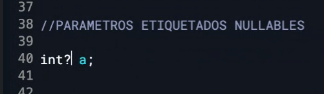
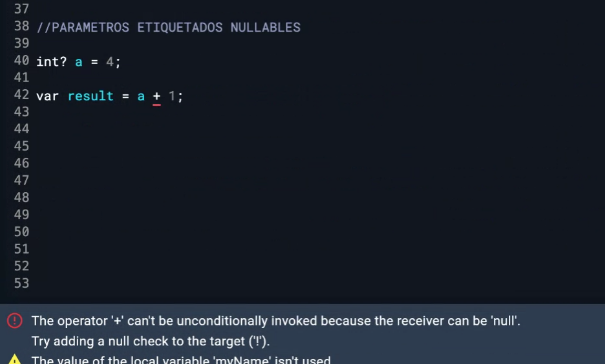
Required pide que se necessita argumento sí o sí.  

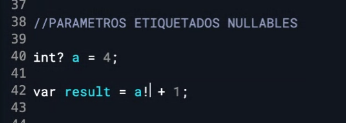
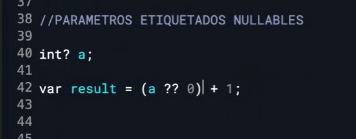

Hay parametros opcionales, puede no pasarse uno por ejemplo:   


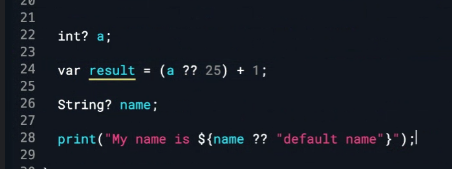
Age es opcional y su valor por defecto si se pasa serà 10 ( a no ser que se especifique otro como aquí: Además al estar **etiquetados** no hace falta que vayan en orden .



\*Nullability:

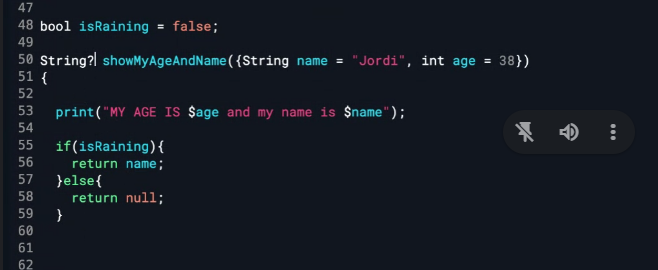
  
No se queja Dart: Le decimos esta variable a puede ser null, no es un error con **?**.   
Pero al poder ser null si hacemos una operacion ya no deja operar, para hacer esto hay que:   


  
con a! El **!** dice que te aseguras que a no es null. Pero esto dara error fàcil. La manera correcta de resolcer esto es :  
  
siempre que operemos con valor que puede ser null, le ponemos valor por defecto si a es null con **??** 0 (le damos valor 0 en este caso.



CONSTANTES:  
  
para que sean immutables

Una manera comun de declarar constantes es poner k minúscula primero.  


  
puede devolver null o string.