# Listas

# Android Estudio V3.0.17/Flutter

Objetivo:

F.Creación: 3-4-2018

Programa:

Fuente: [Oficial](https://api.dartlang.org/stable/1.24.3/dart-core/List-class.html)

[Programa de Contactos](https://medium.com/@XensS/flutter-iii-listas-e-%C3%ADtems-818be6b9c226)

## Listas Fijas

rta(){  
 List<String> lFijo = **new** List(5);  
 lFijo[0]=**"Max"**;  
 lFijo[1]=**"Malva"**;  
 lFijo[2]=**"Amanda"**;  
 int numEl=lFijo.**length**;  
 String celda =lFijo[3];  
 **return " va el** $numEl**, [**$celda**]"**;  
}

retorna-> va el 5,[null]

lFijo.length = 0; // Error

lFijo.add(“Laura”); // Error

fixedLengthList[0] = “Laura”;

## Listas variables simples

String rta(){  
 List<String> lVar = **new** List();  
 lVar.add(**"Max"**);  
 lVar.add(**"Malva"**);  
 lVar.add(**"Amanda"**);  
 int numEl=lVar.**length**;  
 String celda =lVar[2];  
 **return " va el** $numEl**, [**$celda**]"**;  
}

Retorna🡪 va el 3,[Amanda]

fixedLengthList[0] = “Laura”; //Error

String celda =lVar[3];//Error

lVar.**length=0;**

## Lista de una clase

String rta(){  
 List<Pareja> lVar = **new** List();  
 lVar.add(**new** Pareja(**"C01"**,**"Codigo 01"**));  
  
 List<Pareja> par = <Pareja>[  
 **new** Pareja(**"c02"**,**"codigo02"**),  
 **new** Pareja(**"c03"**,**"codigo03"**)  
 ]

Pareja p=par[0];  
 String celda =par[1].**\_nombre**;  
 int numEl=par.**length**;  
 **return " va el** $numEl**, [**$celda**]"**;  
}

**class** Pareja{  
 String **\_nombre**;  
 String **\_descripcion**;  
  
  
 Pareja(**this**.**\_nombre**, **this**.**\_descripcion**);  
  
 String **get nombre** => **\_nombre**;  
  
 **set nombre**(String value) {  
 **\_nombre** = value;  
 }  
  
 String **get descripcion** => **\_descripcion**;  
  
 **set descripcion**(String value) {  
 **\_descripcion** = value;  
 }  
  
}

Imprime ->va el 2, [c03]

## Listas con Contructor de conjunto

String rta(){  
 **const** List<Choice> choices = **const** <Choice>[  
 **const** Choice(title: **'Car'**, icon: Icons.*directions\_car*),  
 **const** Choice(title: **'Bicycle'**, icon: Icons.*directions\_bike*),  
 **const** Choice(title: **'Boat'**, icon: Icons.*directions\_boat*),  
 **const** Choice(title: **'Bus'**, icon: Icons.*directions\_bus*),  
 **const** Choice(title: **'Train'**, icon: Icons.*directions\_railway*),  
 **const** Choice(title: **'Walk'**, icon: Icons.*directions\_walk*),  
 **const** Choice(title: **'Coche de guagua'**, icon: Icons.*child\_friendly*),  
  
 ];  
  
  
 int numEl=choices.**length**;  
 String celda=choices[6].**title**;  
 **return " va el** $numEl**, [**$celda**]"**;  
}  
  
**class** Choice {  
 **const** Choice({ **this**.**title**, **this**.**icon** });  
 **final** String **title**;  
 **final** IconData **icon**;  
}

Al pasarle los parámetros al constructor como opcionales, con los **{}**, nos permite luego pasar ambos por nombre

A la hora de crear el contacto usamos **const Choice** y no **new Choice** para que sea constante en tiempo de compilación.

Imprime 🡪 va el 7 [coche de guaguas]

## Iterable<E>

Una colección de variables que puede ser accesada secuencialmente.

Las listas y conjuntos son ambos iterables

String rta(){  
 String linea=**""**;  
 Map kidsBooks = {**'Matilda'**: **'Roald Dahl'**,  
 **'Green Eggs and Ham'**: **'Dr Seuss'**,  
 **'Where the Wild Things Are'**: **'Maurice Sendak'**};  
 **for** (**var** book **in** kidsBooks.**keys**) {  
 linea+=**'(**$book **was written by** ${kidsBooks[book]}**),'**;  
 }  
 **return** linea;  
}

## Implementación de un árbol

**class** Entry {  
 Entry(**this**.**title**, [**this**.**children** = **const** <Entry>[]]);  
 **final** String **title**;  
 **final** List<Entry> **children**;  
}  
  
*// The entire multilevel list displayed by this app.***final** List<Entry> data = <Entry>[  
 **new** Entry(**'Chapter A'**,  
 <Entry>[  
 **new** Entry(**'Section A0'**,  
 <Entry>[  
 **new** Entry(**'Item A0.1'**),  
 **new** Entry(**'Item A0.2'**),  
 **new** Entry(**'Item A0.3'**),  
 ],  
 ),  
 **new** Entry(**'Section A1'**),  
 **new** Entry(**'Section A2'**),  
 ],  
 ),  
 **new** Entry(**'Chapter B'**,  
 <Entry>[  
 **new** Entry(**'Section B0'**),  
 **new** Entry(**'Section B1'**),  
 ],  
 ),  
 **new** Entry(**'Chapter C'**,  
 <Entry>[  
 **new** Entry(**'Section C0'**),  
 **new** Entry(**'Section C1'**),  
 **new** Entry(**'Section C2'**,  
 <Entry>[  
 **new** Entry(**'Item C2.0'**),  
 **new** Entry(**'Item C2.1'**),  
 **new** Entry(**'Item C2.2'**),  
 **new** Entry(**'Item C2.3'**),  
 ],  
 ),  
 ],  
 ),  
];

## PageStorageKey<T> class

PageStorage se usa para guardar y restaurar valores que pueden sobrevivir al widget. Los valores se almacenan en un mapa por ruta cuyas claves están definidas por PageStorageKeys para el widget y sus antecesores. Para hacer posible que se encuentre un valor guardado cuando se recrea un widget, los valores de la clave no deben ser objetos cuya identidad cambiará cada vez que se crea el widget.

**import 'package:flutter/material.dart'**;  
  
**class** ExpansionTileSample **extends** StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 **return new** MaterialApp(  
 home: **new** Scaffold(  
 appBar: **new** AppBar(  
 title: **const** Text(**'ExpansionTile'**),  
 ),  
 body: **new** ListView.builder(  
 itemBuilder: (BuildContext context, int index) => **new** EntryItem(data[index]),  
 itemCount: data.**length**,  
 ),  
 ),  
 );  
 }  
}  
  
*// One entry in the multilevel list displayed by this app.***class** Entry {  
 Entry(**this**.**title**, **this**.**link**,[**this**.**children** = **const** <Entry>[]]);  
 **final** String **title**;  
 **final** String **link**;  
 **final** List<Entry> **children**;  
}  
*//*String miUrl=**"https://api.dartlang.org/stable/1.24.3/dart-core/List-class.html"**;  
*// The entire multilevel list displayed by this app.***final** List<Entry> data = <Entry>[  
 **new** Entry(**'Chapter A'**,**""**,  
 <Entry>[  
 **new** Entry(**'Section A0'**,**""**,  
 <Entry>[  
 **new** Entry(**'Item A0.2'**,miUrl),  
 **new** Entry(**'Item A0.3'**,miUrl),  
 ],  
 ),  
 **new** Entry(**'Section A1'**,miUrl),  
 **new** Entry(**'Section A2'**,miUrl),  
 ],  
 ),  
 **new** Entry(**'Chapter B'**,**""**,  
 <Entry>[  
 **new** Entry(**'Section B0'**,miUrl),  
 **new** Entry(**'Section B1'**,miUrl),  
 ],  
 ),  
 **new** Entry(**'Chapter C'**,**""**,  
 <Entry>[  
 **new** Entry(**'Section C0'**,miUrl),  
 **new** Entry(**'Section C1'**,miUrl),  
 **new** Entry(**'Section C2'**,**""**,  
 <Entry>[  
 **new** Entry(**'Item C2.0'**,miUrl),  
 **new** Entry(**'Item C2.1'**,miUrl),  
 **new** Entry(**'Item C2.2'**,miUrl),  
 **new** Entry(**'Item C2.3'**,miUrl),  
 ],  
 ),  
 ],  
 ),  
];  
  
*// Displays one Entry. If the entry has children then it's displayed  
// with an ExpansionTile.***class** EntryItem **extends** StatelessWidget {  
 **const** EntryItem(**this**.**entry**);  
  
 **final** Entry **entry**;  
  
 Widget \_buildTiles(Entry root) {  
 **if** (root.**children**.**isEmpty**)  
 **return new** ListTile(title: **new** Text(root.**title**));  
 **return new** ExpansionTile(  
 key: **new** PageStorageKey<Entry>(root),  
 title: **new** Text(root.**title**),  
 children: root.**children**.map(\_buildTiles).toList(),  
 );  
 }

Programa tiles

@override  
 Widget build(BuildContext context) {  
 **return** \_buildTiles(**entry**);  
 }  
}  
  
**void** main() {  
 runApp(**new** ExpansionTileSample());  
}

## Cargar una lista de fotos

List<Container> \_buildGridTileList(int count) {

return new List<Container>.generate(

count,

(int index) =>

new Container(child: new Image.asset('images/pic${index+1}.jpg')));

}