Báo Cáo Khóa học Futter

The Complete 2020 Flutter Development Bootcamp with Dart

Created by Dr. Angela Yu

1. Top tips for the Course:
   1. Go from lesson to lesson.
   2. Not Code Along. Understand all content of lesson. (Không code theo nên xem video 10’ sau đó tự code lại.)
   3. Practice typing skills.
   4. Cornell Note Taking System.
   5. Adjust Speed.
   6. Don’t Skip.
   7. Bookmark And Read them again usually.
   8. Practice = Mastery.
   9. Struggle is Good. Use StackOverflow.
2. I Am Rich – How to Create Flutter Apps from Scratch
3. Flutter

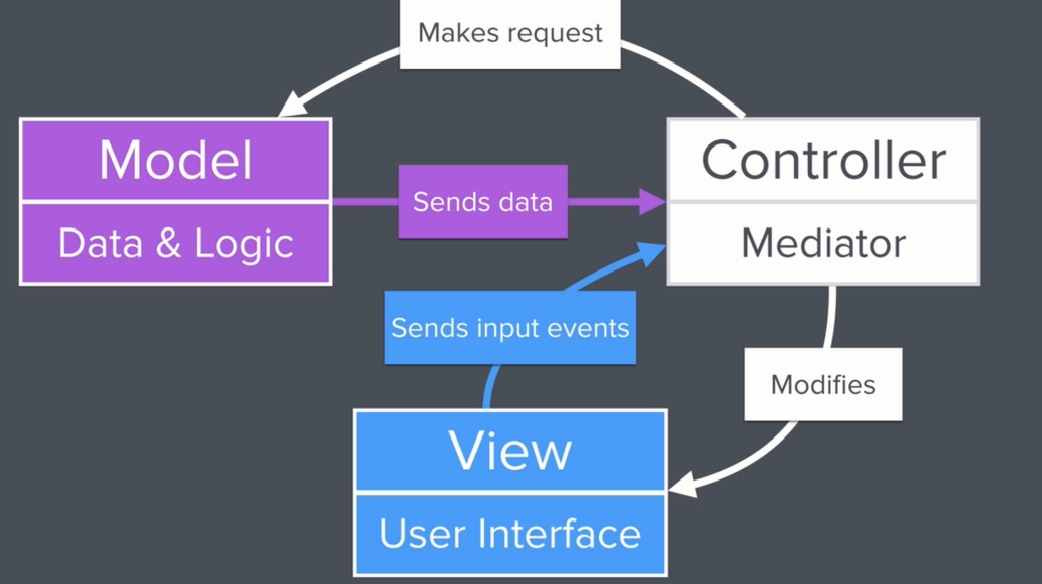
* By default all text widgets get aligned to the top left corner.
* Flutter apps are entirely built using widgets.
* Add a comma to the end of all of round brackets.
  + Because it makes easier to format.
* MaterialApp: Can access to all of colors material.
* Scaffold:
  + Provides APIs for showing drawers, snack bars, app bar, bottoms sheets, etc.
  + body: the primary content of the scaffold.
* YAML: YAML Ain’t Markup Language.
  + Be really careful about the indentation.(Tips)
* Assets:
  + Instead of specifying the precise file name, we can also just specify the folder
* Folder icon:
  + Android: app/src/main/res/…
  + IOS: Runner/Assets.xcassets/Appicon.appiconset/…
* Configue Image Asset (Android):
  + Click right at res folder.

1. Dart

* **=>**: fat arrow (Shorthand syntax) is used to define a single expression in a function.
* The **main** function is the starting/entrance point for all of Dart programs.
* We should work always with single quotes.
* All in Dart is Object.
* String interpolation: ‘$tetxt${variable}text’
* Type:
  + Statically Typed Language: <> dynamically typed programming language ( Javascript).
  + Dart infer the data type from the value.
  + If just declare “var” not have the value it will turn into dynamic.
  + Should create variables with the data type to start with.
* Random:
  + Random().nextInt(max): 0 -> max - 1.

1. Others

* Websites:
  + Appicon.co, icons8.com, canva.com, vecteezy.com

1. I Am Poor – App challenge:
   1. Tips from Angela – Dealing with Distractions:
      1. Find a quiet time, quiet place.
      2. Put our phone into airplane mode and put it into a drawer.
2. MiCard – How to Build Beautiful UIs with Flutter Widgets.
   1. App History of Everything.
   2. Hot Reload:
      1. just working on stateless or stateful widget.
      2. Just call “build” method.
      3. Press Ctrl + S.
      4. Reduce time to develop the app.
      5. Won’t lose data (form).
   3. Hot Restart:
      1. Reset app (reset data, state).
   4. “build” method:
      1. Be called every time we make a change to one of the widgets inside that build method.
   5. Widgets:
      1. Container:
         1. Combines common painting, positioning, sizing widgets.
         2. Fit decoration to the narrowest parameter.
         3. When set alignment the container will expand to fill its parent’s width and height.
         4. Container with no children try to be as big as possible.
         5. Containers with children size themselves to their children.
         6. Margin: outside of widgets.
         7. Padding: inside of widgets.
      2. Safe Area:
         1. Put inside the safe area (without notch or an interactive area).
      3. Column/ Row
         1. Try to take up all of the space that is available vertically.
            1. Set mainExisSize = MainAxisSize.min if want to get minimum size of height.
         2. Width is limiting itslf to the size of its children.
         3. verticalDirection: Change vertical direction (up or down(default))
         4. mainAxisAligment:
         5. crossAxisAlignment:.
            1. Add max width of container to Column in order to use crossAxisAlignment.end.
            2. Stretch:
            3. baseLine: các dòng bằng nhau dù fontSize chênh lệch. (phải có textBaseLine).
            4. textBaseLine: xác định baseline
      4. SizedBox
         1. add space between 2 widgets.
      5. CircleAvatar
         1. backgroundImage: set image background.
      6. TextStyle:
         1. letterSpacing.
      7. Icon.
         1. Is Vector.
         2. More Flexible than Image(change color and size easily).
         3. Doesn’t become pixelated.
      8. Card
         1. Don’t have padding.
         2. Combine with ListTile.
      9. Padding.
         1. Add padding.
      10. ListTile.
          1. Add icon, title.
      11. Image:
          1. AssetImage: Image Provider.
      12. Expanded
          1. Make child of Row, Column, or Flex expand to full the available space in the main axis.
          2. If multiple children are expanded, the available space is divided among them according to the flex factor.
          3. Must be a descendant of a Row, Colum, or flex
   6. Flutter Layout Cheat Sheet: [https://medium.com/flutter-community/flutter-layout-cheat-sheet-5363348d037ehttps://medium.com/flutter-community/flutter-layout-cheat-sheet-5363348d037e](https://medium.com/flutter-community/flutter-layout-cheat-sheet-5363348d037e)
   7. Custom font
      1. Import the font files.
      2. Declare the font in the pubspec.
      3. Set a font as the default.
      4. Use a font in a specific widget.
   8. Stateful vs Stateless Widgets
      1. Stateless Widgets:
         1. Create a UI where the state of the widget isn’t going to change.
      2. Stateful Widgets:
         1. Create a UI where the state of the widget is going to change.
         2. setState() {}
            1. trigger to rebuild UI. (call method “build”);
            2. when we call setState we update our variables ,it will mark all of places where it’s being used as dirty. And we call build again everything inside gets re-evaluated, it’ll find the things that are marked dirty, and it will redraw.
   9. Tip from Angela:
      1. Use Pomodoro Technique: focused on 25’, relaxed on 5’ minutes.
3. Xylophone - Using Flutter and Dart Packages to Speed Up Development
   1. Flutter Packages:
      1. Are open source libraries of code that other people have created which we can incorporate into our project.
      2. Với các widgets nhỏ ta dùng builder Widget (dùng function trả về widgets).
   2. Dart:
      1. Function:
         1. Named parameters: nameFunction({Type variables}).
            1. When have a lot of parameters.
4. Quizzler -Modularising & Organising Flutter Code
   1. Dart:
      1. List
         1. Insert: Inserts the object at position [index] in this list.
         2. indexOf: Returns the first index of [element] in this list.
      2. Other:
         1. Use “\” (backslash) if we want to use special characters in dart.
      3. Class
         1. Is a blueprint.
         2. Properties/fields.
         3. Method.
      4. OOP
         1. Abstraction
            1. Separated different jobs into class.
         2. Encapsulation
            1. Safe for app (not be able to change properties).
         3. Inheritance
         4. Polymorphism.
   2. Android Studio.
      1. VCS/ Local History: view history of codes.
5. Boss Level Challenge 2 – Destini
   1. Container:
      1. Set Background: decoration: BoxDecoration (DecorationImage).
   2. Visibility: set visibility of widgets.
6. BMI Calculator - Building Flutter UI for Intermediates
   1. Websites: <https://dribbble.com/> ColorZilla
   2. Theme:
      1. ThemeData:
      2. Also is a widget can wrap child widgets.
   3. Widgets
      1. Key class
         1. An identifier for widgets, elements, …
      2. Container:
         1. If have boxDecoration we have to move color into “decoration”.
         2. Width: double.infinity: max Width
      3. GestureDetector
         1. Detects gestures.
      4. Slider:
         1. Custom by SliderTheme(widgets).
      5. FloatingActionButton
         1. Use at most a **single** floating action button per screen.
      6. RawMaterialButton
         1. Dùng để custom button.
         2. Chỉ hiển thị elevation khi enable.
         3. Disable bằng cách set onpress = null;
   4. Principles
      1. DRY: Don’t Repeat Yourself.
   5. Flutter;
      1. @required: named parameters
   6. Dart
      1. Const vs final
         1. Const: compile-time constant.
            1. Vd: const time = DateTime.now(). Not working
         2. Final: runtime. (can be set only once).
            1. Vd: final time = DateTime.now(). Working
      2. Ternary:
         1. Condition ? DothisIfTrue : DoThisIfFalse.
         2. Sử dụng khi gán giá trị.
      3. String:
         1. toStringasFixed(number);
   7. Dart:
      1. Function:
         * 1. Is a First Class Objects: can be pass around like any other type. (Giống kotlin).
      2. Composition vs Inheritance:
         1. Flutter chủ yếu sử dụng composition (Android sử dụng inheritance các widgets chủ yếu được inheritance từ Text).
         2. Favor composition over inheritance.
      3. Map
         1. (key, value).
         2. Is basically a collection type.
   8. Routes
      1. Navigator
         1. Push:
            1. MaterialPageRoute
         2. Pop: return to the previous route:
         3. Named route:
            1. “*routes*”(MaterialApp).
            2. ‘/’: home screen.
            3. Dùng *initalRoute* thay cho *home.*
7. Clima - Powering Your Flutter App with Live Web Data
   1. Asynchronous:
      1. Future
      2. Async & await.
   2. Lifecycle:
      1. initState:
      2. Inactive
      3. Build
   3. Exception:
      1. Try
      2. Catch
      3. throw
   4. ?? (null aware operator).
      1. Check null: someVariable ?? defaultValue
   5. “By failing to prepare, you are preparing to fail” Benjamin Franklin.
   6. API (Application Programming Interface)
      1. Is a set of commands, functions, protocols, and objects that programmers can use to create software or interact with an external system.
   7. http:
      1. get (Future<Response>) response.body.
      2. codes cheat sheet:
         1. 1\*\* Hold on
         2. 2\*\* Here you go
         3. 3\*\* Go away
         4. 4\*\* You fucked up
         5. 5\*\* I fucked up
      3. Using *as* after import http package
   8. Widgets
      1. SpinKit (loading…)
      2. TextField
         1. Filled
         2. fillColor
         3. icon
         4. decoration: lineThrough
   9. StatefulWidget
      1. Have property ‘widget’ in State was linked with Widget.
   10. Navigator.
       1. Pop(context, data)
       2. Push() Future<T> : <T> is data of screen pop.
8. Boss Level Challenge 3 - Bitcoin Ticker
   1. Widgets
      1. DropdownButton< >
      2. CupertinoPicker
   2. Import
      1. Show Class ( just show this class in library).
      2. Hide Class (hide this class in library).
9. Flash Chat - Flutter x Firebase Cloud Firestore
   1. Static
      1. is useful for class-wide state and constants.
      2. Why have to have static when set property of class with **const?**
         1. **.**
   2. Animation
      1. Hero Animation
         1. 2 \* Hero Widgets
         2. A shared tag property
         3. Navigator-based screen transitions.
      2. Animation
         1. A ticker
         2. Animation controller
         3. Animation value
      3. Animation Controller
         1. Vsync: SingleTickerProviderStateMixin
            1. Which going to provide the ticker for animation controller.
         2. Forward: proceed animation forwards.
         3. addListener:
         4. duration / 60 : 60 tick / second
         5. reverse
      4. animation:
         1. addStatusListener
            1. dismissed: end of the reversed animation.
            2. Completed: end of the forward animation.
   3. Dispose
      * 1. Avoid memory leak.
   4. CurvedAnimation:
   5. ColorTween.
   6. Nên định nghĩa (generic) loại animation (Animation<T>) sẽ tránh gán nhầm value.
   7. Dart
      1. Mixins
         1. Are way of reusing a class’s code in multiple class hierarchies.
         2. Makes program easier to maintain. (shared capability split out in to a mixin).
         3. Should use instead of inherit.
      2. Is single inheritance pattern (only extends one class)
   8. animated\_text\_kit
   9. nếu muốn extractor 1 style (decoration) mà có 1 vài thuộc tính cần thay đổi ta dùng copywith.
   10. Widgets
       1. TextField
          1. keyboardType
          2. obscureText
       2. StreamBuilder
          1. Check hasData asyncSnapshot.
       3. Material:
          1. radiusBorder
          2. elevation.
       4. Flexible
          1. Creates a widget that controls how a child of a [Row], [Column], or [Flex] flexes.
       5. ListView
          1. Reverse: true
          2. Builder:
             1. Need to build a large number of children.
             2. Just create item that are actually visible.
       6. BottomSheet
          1. Muốn tạo border cho bottomSheet ta dùng 2 container chồng lên nhau, (Lớp 1: trùng màu nền khi hiện lên, lớp 2: màu tùy ý.)
   11. Firebase
       1. Khi sử dụng firebase ta add libray trong gradle android.
       2. Firebase Auth
       3. Cloud FireStore
       4. Lưu ý khi sử dụng phiên bản bên flutter và android.
   12. Stream.
   13. Context
       1. Is used to look up the Navigator and Theme.
   14. State
       1. Global state
          1. Use it across different parts of your app.
       2. Lifting state up.
   15. MVC
       1. 
   16. Imperative vs Reactive.
       1. Reactive:
          1. Prop Drilling.
             1. We have to essentially drill down through the tree in order to be able to reach the lower levels.
   17. State Management:
       1. BloC, setStateProvider, Scoped Model, Lifting State Up, Redux, MobX, MVC, InheritedWidget.
       2. Lựa chọn dựa trên Requirement Based.
   18. Provider
       1. Provider<T> (builder : return data)
       2. Provider.of<T>(context, listener : bool) : return data.
       3. ChangeNotifier (thuộc về Flutter).
          1. ChangeNotifierProvider<T>
       4. Dựa trên InheritedWidget.
       5. Khi muốn thay đổi data ta get listener = false;
       6. Consumer<T>(builder (context, T, child))
   19. InheritedWidget
       1. Allow us to pass data down the tree without having to rebuild the intervening(ở giữa) part of the tree.
   20. Ngăn chặn sự thay đổi từ bên ngoài thuộc tính (encapsulation) (Giống kotlin).
       1. List<Type> \_list;
       2. UnmodifiableListView<Type> list = UnmodifiableListView(\_list)