순한 맛 시즌1

Flutter 로드맵

Flutter: Framework. Dart. Cross platform. Hummingbird(web project).

Setting

1.install flutter, setting environment path (add flutter sdk)

2.install android studio, visual studio code

3.flutter doctor, create project with VSCode, Android Studio

Error!

1. cmdline-tools component is missing

Run `path/to/sdkmanager --install "cmdline-tools;latest"`

See https://developer.android.com/studio/command-line for more details.

-answer: go android SDK Manager > SDK Tools > check Android SDK Command-line Tools(latest)

2. Android Licenses problems

-answer: input command flutter doctor –android-licenses

3. start emulator in VSCode

-answer: ctrl + shift + p > flutter select device > start “emulator”

\*statement

\*Riverpod, GetX

\*Fuchsia(OS)

\*Zircon(마이크로 커널)

위젯!!

기본: 독립적으로 실행되는 작은 프로그램. 그래픽이나 데이터 요소를 처리하는 함수를 가짐.

Flutter: UI를 만들고 구성하는 모든 기본 단위 요소. 눈에 보이지 않는 요소들까지 위젯. 모든 것이 위젯이다.

Stateless widget: stateful: 이전 상호작용의 어떠한 값도 저장하지 않음. 상태가 없는 정적인 위젯.스크린 상에 존재만 할 뿐 아무것도 하지 않음. 어떠한 실시간 데이터도 저장하지 않음. 어떤 변화를 유발시키는 value값을 가지지 않음.

Stateful widget: stateless: value 값을 지속적으로 추적 보존. 계속 움직임이나 변화가 있는 위젯. 사용자의 interaction에 따라서 모양이 변함. 데이터를 받아도 모양이 변함.

Inherited widget

Widget tree: widget들은 tree 구조로 정리될 수 있음. 부모 위젯(widget container)과 자식 위젯으로 구성. 예) root -> materialapp ->myhomepage -> scaffold(앱 화면과 빈 페이지) -> appbar, center -> text, column -> ,image,text

Pubspec.yaml: 프로젝트의 meta 데이터. 버전, 3rd party, 사용환경, dependency, library 등

Main.dart: import flutter/material 필수. Main => runApp(MyApp()). MyApp() 커스텀 위젯 만들고 build() 내부에 MaterialApp() 위젯 생성. Title: 말 그대로 app의 이름, theme, home: MyHomePage 커스텀 위젯 구현. MyHomePage build() 내부에 title: Text() 위젯을 통해 생성.

\*material

\*argument: 인자

\*swatch: 견본, primarySwatch: 색상 견본.

\*closing label: 위젯의 구분이 중요하기 때문에 주석으로 나온다.

\*scaffold: 무언가를 혼자 해낼 수 있도록 발판을 만들어 주다.

Error!

1. const 문제. Column 위젯은 const를 붙일 수 없다.

-answer: Parent Widget의 const를 지운다

1-1. const 문제. [] 내부의 Text() 위젯에 const를 붙일 수 없다.

-answer: const []로 해결.

1-2. const 문제. 파란줄로 뜨기 때문에 에러는 아니다. 하지만 매번 const를 못 붙이는 거를 구분해야 하기 때문에 불편하다.

-answer:

\*Axis: 세로 축.

클래스와 위젯

클래스: 설계도.

객체: 클래스가 메모리 상에 등록된 것.

인스턴스: 생성된 객체.

플러터 순한 맛 강좌 17 다시 보기

\*★BuildContext: widget tree에서 현재 widget의 위치를 알 수 있는 정보. 모든 widget은 자신의 BuildContext를 가진다, 이 BuildContext는 statelesswidget이나 state build 메서드에 의해서 리턴된 위젯의 부모가 된다.

Something.of(context): 현재 주어진 context에서 위로 올라가면서 가장 가까운 Something을 찾아서 반환해라.

Error!

1. MissingPluginException

텍스트이(가) 표시된 사진

자동 생성된 설명

\*margin: 위젯의 바깥쪽

\*padding: 위젯의 안쪽

\*container: 하나의 위젯을 가진다.

Error!

1. Inspector in VSCode

-answer: ctrl + shift + p > Flutter: Inspector or Dart Inspector 입력 후 사용. 둘 중 Flutter가 반응이 좀 더 좋은 듯하다.

\*route: 스마트폰 상에서 보여지는 하나의 페이지. Scaffold를 리턴할 때 사용하는 widget 하나하나가 다 route라고 할 수 있다.

\*매개변수

\*인자

\*Map: key: value 구조.

\*collection: 데이터들을 모아서 가지고 있는 자료구조.

\*generic: Collection이 가지고 있는 데이터들의 데이터 타입을 지정.

\*interpolation: 보간법. 문자열 사이에 $를 사용해 변수의 값을 출력하는 것

Dart – List: var number = new List.empty(growable:true); or [];

RaisedButton => ElevatedButton, FlatButton => TextButton, OutlineButton => OutlinedButton

조금 매운 맛

\*hot reload: 변경된 부분만을 확인해 바꾼다.

\*state: UI가 변경되도록 영향을 미치는 데이터.

\*stateless widget: state가 변하지 않는 위젯. rebuild만을 통해서 새로운 state 적용 가능.

\*tree: widget tree, element tree, render tree.

\*reload: frame은 그대로 둔 채 부수적인 요소들을 바꾸는 것, rebuild: 전체를 바꾸는 것.

setState()를 통해 상태 변화를 적용할 수 있다.

FloatingActionBar의 tooltip property는 long presses 시 해당 버튼에 대한 설명이 나온다.

Focus: FocusNode: 포커스를 받는 특정 위젯을 식별, FocusScope: 어떤 위젯들까지 포커스를 받는지 범위,

|  |  |
| --- | --- |
| Login app | |
| 텍스트, 모니터, 전자기기, 스크린샷이(가) 표시된 사진  자동 생성된 설명1차 목표 | 7.25 |
| 7.26  Error! Bottom overflowed by xx pixels  Solution   1. Add resizeToAvoidBottomInset: false in Scaffold() 2. Wrap body with SingleChildScrollView()   Problem! How can I get text in TextField  Solution  Using TextEditingController()  <https://eunjin3786.tistory.com/254> |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| Dice game app | |
| 텍스트, 전자기기, 스크린샷이(가) 표시된 사진  자동 생성된 설명2차 목표 | 7.26  Error! Image loading  Solution  경로 오류 assets/images/xx.png  Error! Remaining Snackbar  Solution  Wrap Scaffold with ScaffoldMessenger |
| 7.27    다음 목표는 플레이 버튼을 눌렀을 때 랜덤으로 주사위가 돌아가고 그에 맞는 이미지가 출력되는 것. | 7.30  랜덤으로 이미지를 불러오는 함수 getRandomDice()  Error! 버튼 클릭 시 이미지 변경  Solution  Stf widget lifecycle 중에 setState()를 이용한다.  <https://m.blog.naver.com/cdw0424/221867416467> |
|  |  |

Final: 오직 한 번만 설정할 수 있다. Run time constant. Rebuild.

Const: 컴파일 시에 상수 취급. Compile time constant. 선언과 동시에 초기화.  
접근제어자: private \_.

Future: 현재는 실제로 존재하지 않지만, 시간이 지나 구체적인 결과물로 나타난다는 약속. 클래스. Async. Await.

FutureBuilder: widget. Can return future.

For, for in, forEach.

List<>.generate(length,(index)=>)

Shuffle(), sublist()

Set<>

While

cascade표기법: ..

null safety: null(아직 값이 정해지지 않은 것). Null check를 할 필요가 없다. 전제1.모든 변수는 null이 될 수 없다. 전제2.Class 내의 변수는 생성과 동시에 초기화를 시켜야 한다. Type? == nullable type. Nullable은 non nullable에 저장될 수 없다. 그 때 ‘!’ 를 사용해 null값이 아니라고 알려준다.

Lazy initialization

Widget lifecycle

Api: 일련의 표준화된 명령어나 기능. 매개 역할자.

Exception handling

Json(JavaScript Object Notation): key 값과 value값 1대1 매치. vs XML