

Bài 5 - Cơ chế Render

Flutter có dùng button của android không??

Không

=> Tự vẽ toàn bộ giao diện bằng Engine riêng

=> Flutter App => Widget => Render => Skia Engine => GPU render

Chú ý: gần giống 1 game engine

=====

Widget là bản thiết kế giao diện (bản vẽ), Render Object sinh ra giao diện dựa trên Widget (Ngôi nhà)

Widget rất nhẹ, tạo, hủy liên tục => Flutter nhanh

=====

Hiển thị giao diện trong flutter

MaterialApp

Scaffold

AppBar

Body

Column

Text

Image

Button

=====

Stateless: Build và chạy 1 lần (không đổi dữ liệu, không lưu trạng thái)=> dùng khi sử dụng Text, Icon, UI tĩnh

=====

StateFull: Cấu trúc có 2 lớp

StatefulWidget: Cấu hình

State: Dữ liệu logic

Tại sao phải tách làm 2 lớp???

Vì flutter có thể hủy state và giữ lại giao diện hoặc ngược lại => tách 2 phần để tái sử dụng

=> tối ưu về mặt hiệu năng

=====

setState(): cơ chế để cập nhật giao diện

=> Khi nào giao diện thay đổi??? khi gọi setState (khi có dữ liệu thay đổi)

=> qui trình thay đổi giao diện:

Flutter đánh dấu cái cần thay đổi => gọi build lại => so sánh phần cũ và phần mới => chỉ vẽ lại phần thay đổi (chú ý: Không vẽ lại toàn bộ giao diện)

=====

Hàm build()

Có thể chạy nhiều lần

=> Build không được làm các việc sau:

-Không gọi API

-Không tạo DB

-Không tính toán nặng

=> được làm gì: nhanh, không nhiều hiệu ứng

=====

Render layout

Scaffold => body toàn màn hình => Column toàn màn hình (rộng bằng cha)=> Text vừa đủ nội dung

=====

Cách bố trí giao diện theo Column: mainAxisAlignment

Cách bố trí giao diện theo Row: crossAxisAlignment

=====

Widget:

Column/Row

Expanded/Flexible

Padding/Margin

=====

Trong app bán hàng

UI app bán hàng = Widget Tree

Mỗi màn hình = Widget

Dữ liệu thay đổi: setState

=====