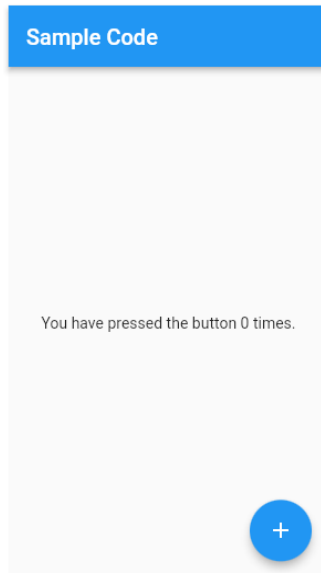


## Getting started (flutter) av Georgij

I flutter använder man Scaffold för att få en grundvy att utgå ifrån. Då får man en header och en body.



Sedan kan man rendera vad man vill i "body" komponenten hos en Scaffold. Den förväntar sig en Widget som är en typ av komponent som antingen kan vara en av flutters inbyggda widgets eller en egen custom widget eller en widget från något publikt paket. Widgets är som komponenter i React och Flutter är baserat på React. Row och Column används för att rendera en rad av Widgets i horizontal respektive vertikalled. Även här märker man inflytandet från React och JavaScript världen där innehållet i dessa placeras ut baserat på "Flexbox" från css. Exempelvis gör space-between att komponenterna lägger sig så långt ut mellan varandra som möjligt precis som i css. Vill man lägga Widgets över varandra istället så kan man använda en Stack som förväntar sig en lista av widgets som placeras (stackas) över varandra baserat på ordningen i listan.

De Widgets som har callbacks som kallas när ett viss event sker, exempelvis när en knapp trycks eller en input ändras, har ofta metoder som agerar som callbacks när dessa event sker. Exempelvis i flutters knapp Widget ElevatedButton finns det en metod som heter onPressed, som är en callback funktion som kommer kallas när någon trycker på knappen. Inuti callbackfunktionen kan man då skriva logik för vad man vill ska ske när någon trycker på knappen.

För att navigera mellan olika skärmar kan man använda sig av den inbyggda navigationen i Scaffold widgeten. Om man skapar två olika Widgets, som utgör två olika screens, som båda har egna Scaffolds som första Widget i sitt Widget träd så kan man enkelt navigera genom dessa via:

```
Navigator.push(  
  context,  
  MaterialPageRoute(builder: (context) => const SecondRoute()), );
```

, där `SecondRoute` är den `Screen` (Widget) man vill navigera till. Flutter via Scaffolden kommer kunna märka det och lägger då en tillbaka knapp på `SecondRoutes` header. `Navigator.push` kan kallas från en enkel knapp och flutter i bakgrunden sköter mycket av jobbet åt oss men vanligtvis vill man ju ha någon typ av lista där man kan se alla routes och från dessa navigera sig och även kunna se var man befinner sig. Ett enkelt sätt att sätta upp detta är att använda sig av den inbyggda `BottomNavigationBar` widgeten. Det är som en botten navigeringsmeny som de flesta mobilappar idag har. `BottomNavigationBar` skapas och används som property i en Scaffold. Man får dock skriva om sin kod lite grann för att få det att funka. `BottomNavigationBar` kommer ha ett index värde baserat på vilken route man tryckt på och man måste därför skapa en lista av Widgets, en för varje route i navigeringsmenyn, vars ett av elementen kan extraheras med det valda indexet som man kan sätta in i body på Scaffolden. Man får därmed ha en conditional body beroende på vad index-värdet från `BottomNavigationBar` har för värde för att kunna visa olika content för olika index för att visa rätt Widget/Screen för en viss route.