

To do before next session

1 - Read articles

<https://docs.flutter.dev/ui/interactivity>

2 - Watch videos

<https://www.youtube.com/watch?v=AqCMFXEmf3w>

<https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddab9f2>

<https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddaba0a>

3 - Answer to following questions

- Why do we have a distinction between stateless and statefull widgets?
- When do I need use a stateless widget?
- When do I need use a stateful widget?

MOBILE DEVELOPMENT

W4-S2 – Stateful Widgets





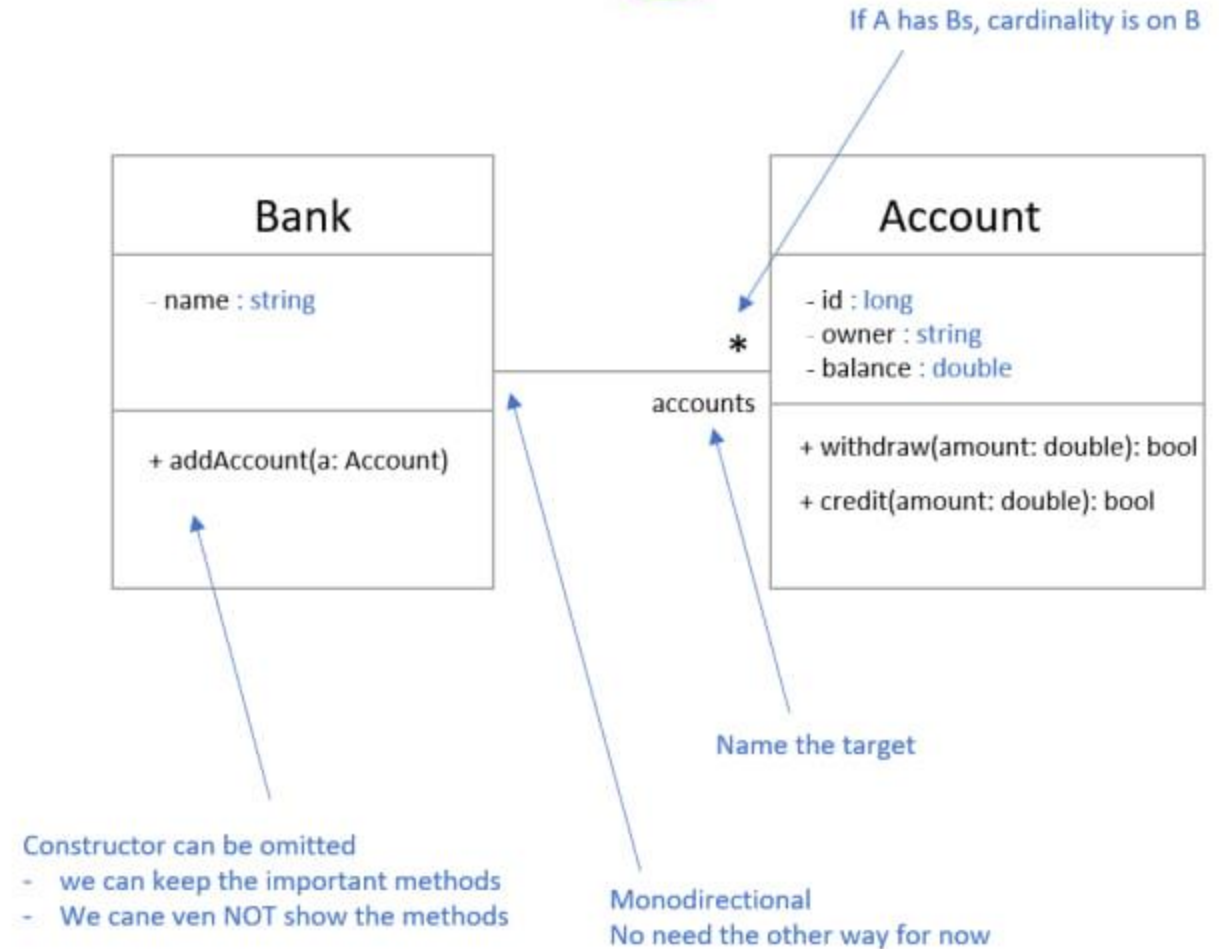
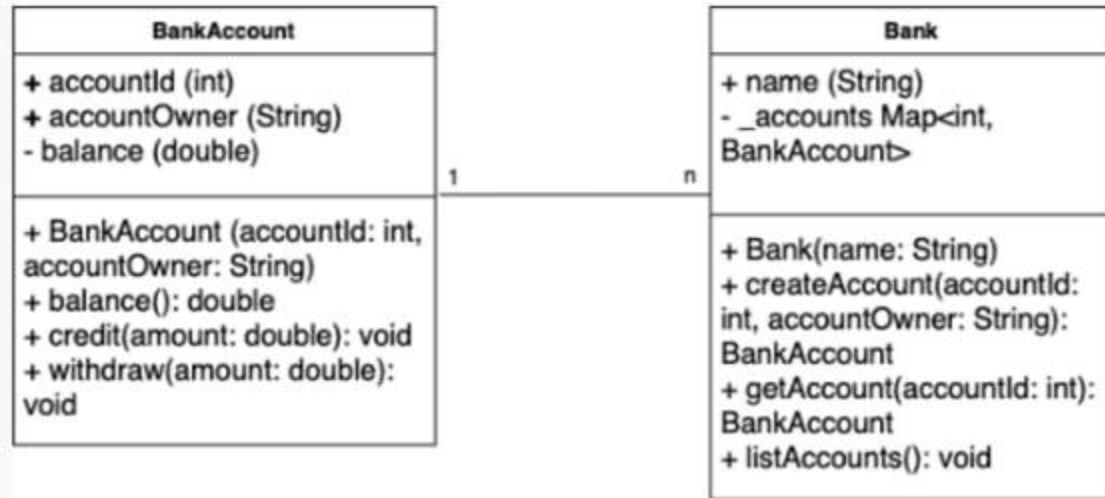
Course Objectives



- ✓ Attach **event** handlers using **anonymous functions** (closures) in Dart.
- ✓ **Differentiate** between **Stateless** and **Stateful** Widgets.
- ✓ Explain the **lifecycle of a Stateful Widget** in Flutter.
- ✓ **Identify use cases** where Stateful Widgets are needed.
- ✓ Create a **Stateful Widget** in Flutter.
- ✓ Use the **setState()** method to update the UI dynamically.

A traffic light **has a state** with 3 values : red, orange, green







Handling Events

Observe and understand the bellow code

```
void onPress(){\n  print("hello");\n}\n\nrunApp(\n  MaterialApp(\n    home: Column(\n      children: [\n        TextButton(\n          onPressed: onPress,\n          child: const Text("Press me"),\n        ),\n      ],\n    ),\n  ),\n);
```

What does this line means?

Why this function is not called with () ?

What is the type of onPressed ?

Handling Events

Observe and understand the bellow code

```
void onPress(){\n  print("hello");\n}\n\nrunApp(\n  MaterialApp(\n    home: Column(\n      children: [\n        TextButton(\n          onPressed: onPress,\n          child: const Text("Press me"),\n        ),\n      ],\n    ),\n  ),\n);
```

What does this line means?

*If the button is pressed,
the function **onPress**
will be called back*

Why this function is not
called with () ?

Because it is **NOT** a call
It s a **callback**

*A reference to a function
that should be called if we pressed
on the button*

What is the type of **onPressed** ? **VoidCallback**

*Which is the signature of callbacks that have no arguments
and return no data.*

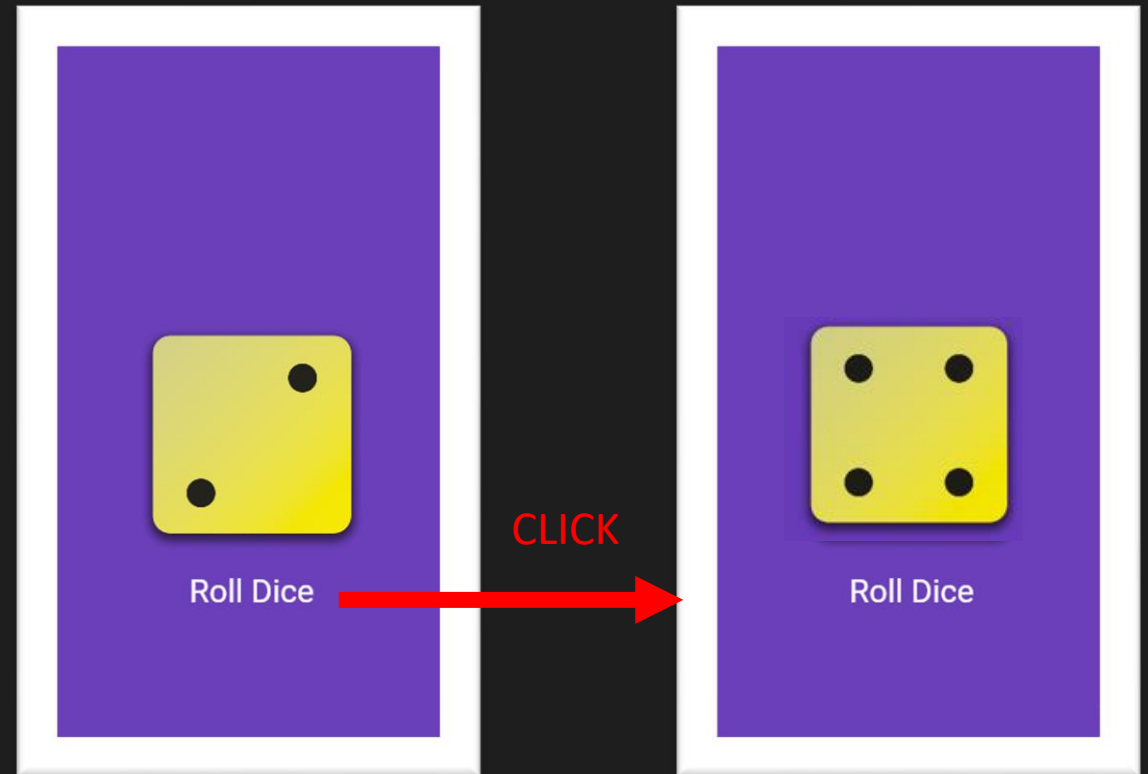
We want to change the dice image upon button pressed

What should we put on the rollDice() callback?

```
const String diceImage2 = 'assets/w4-s1/dice-2.png';  
const String diceImage4 = 'assets/w4-s1/dice-4.png';
```

```
String activeDiceImage = diceImage2;
```

```
class DiceRoller extends StatelessWidget {  
  const DiceRoller({super.key});  
  
  void rollDice() {  
    // Display the dice 4 !  
  }  
  
  @override  
  Widget build(context) {  
    return Column(  
      mainAxisAlignment: MainAxisAlignment.min,  
      children: [  
        Image.asset(  
          activeDiceImage,  
          width: 200,  
        ), // Image.asset  
        const SizedBox(height: 20),  
        TextButton(  
          onPressed: rollDice,  
          child: const Text('Roll Dice'),  
        ) // TextButton  
      ],  
    ); // Column  
  }  
}
```



Why is the widget not showing the updated value?

```
const String diceImage2 = 'assets/w4-s1/dice-2.png';  
const String diceImage4 = 'assets/w4-s1/dice-4.png';
```

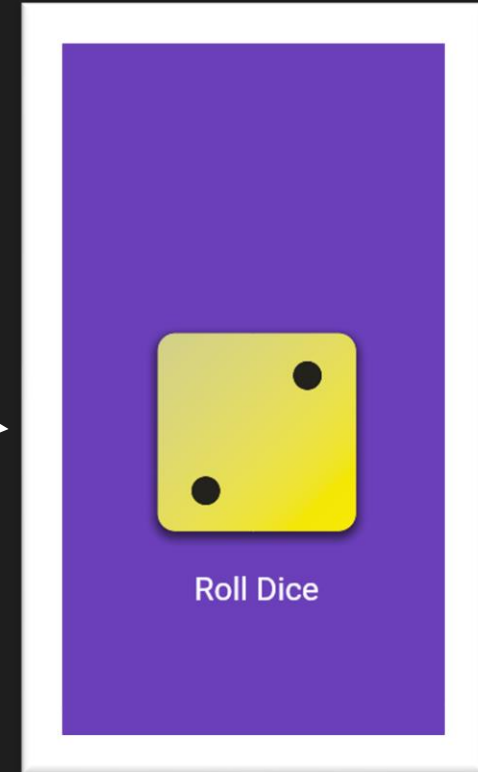
```
String activeDiceImage = diceImage2;
```

```
class DiceRoller extends StatelessWidget {  
  const DiceRoller({super.key});
```

```
  void rollDice() {  
    // Display the dice 4 !  
    activeDiceImage = diceImage4  
  }  
}
```

```
@override
```

```
Widget build(BuildContext) {  
  return Column(  
    mainAxisAlignment: MainAxisAlignment.min,  
    children: [  
      Image.asset(  
        activeDiceImage,  
        width: 200,  
      ), // Image.asset  
      const SizedBox(height: 20),  
      TextButton(  
        onPressed: rollDice,  
        style: TextButton.styleFrom(  
          foregroundColor: Colors.white,  
          textStyle: const TextStyle(  
            fontSize: 28,  
          ), // TextStyle  
        ),  
        child: const Text('Roll Dice'),  
      ) // TextButton  
    ],  
  ); // Column  
}
```



Because there is nothing telling the widget to rebuild itself when the “activeDiceImage” value changes.

A Stateless widget

will never **rebuild by itself**

- ✓ Stateless widgets **can't be changed once** they are built.
- ✓ These widgets are **immutable once they are built**
- ✓ Any change in data, widgets, icons, or variables do not change the state of the app or UI.
- ✓ They simply **override the build()** method and return a widget.



Let's refactor our widget to a
Stateful widget !

```
class DiceRoller extends StatefulWidget {
  const DiceRoller({super.key});

  @override
  State<DiceRoller> createState() {
    return _DiceRollerState();
  }
}
```

The stateful widget remains **unchanged over time**

The stateful widget is in charge of **creating the state object**

```
class _DiceRollerState extends State<DiceRoller> {
  String activeDiceImage = dice2;

  void rollDice() {
    setState(() {
      activeDiceImage = dice4;
    });
  }
}
```

The **state object** holds
 The active image
It's no longer final !

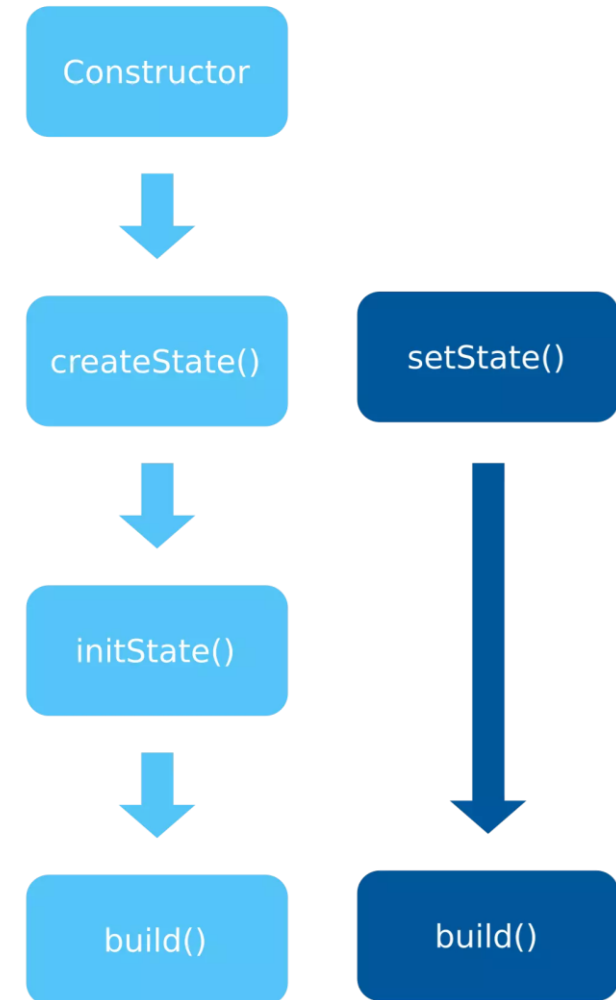
```
@override
Widget build(context) {
  return Column(
    mainAxisAlignment: MainAxisAlignment.min,
    children: [
      Image.asset(
        activeDiceImage,
        width: 200,
      ), // Image.asset
      const SizedBox(height: 20),
      TextButton(
        onPressed: rollDice,
        style: TextButton.styleFrom(
```

The state object is now in charge of **building the child widgets**

A Stateful widget

Will be rebuilt if **a state has changed**

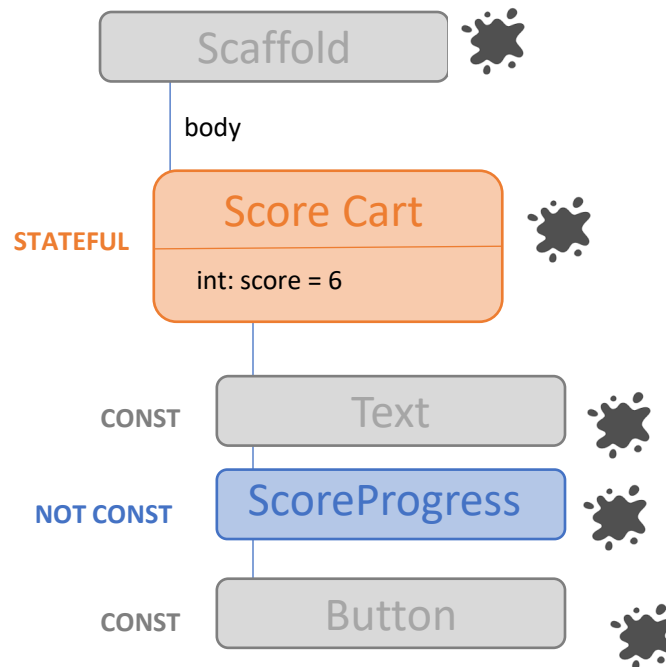
- ✓ Stateful widgets can rebuild its UI when its internal state changes
- ✓ It consists of two classes: a widget class and a state class.
- ✓ **The state is mutable**, and changes are triggered using the **setState()** method.
- ✓ It is used for **interactive elements** that need to be updated in case of dynamic changes.



When Widget are **tagged as dirty** ?

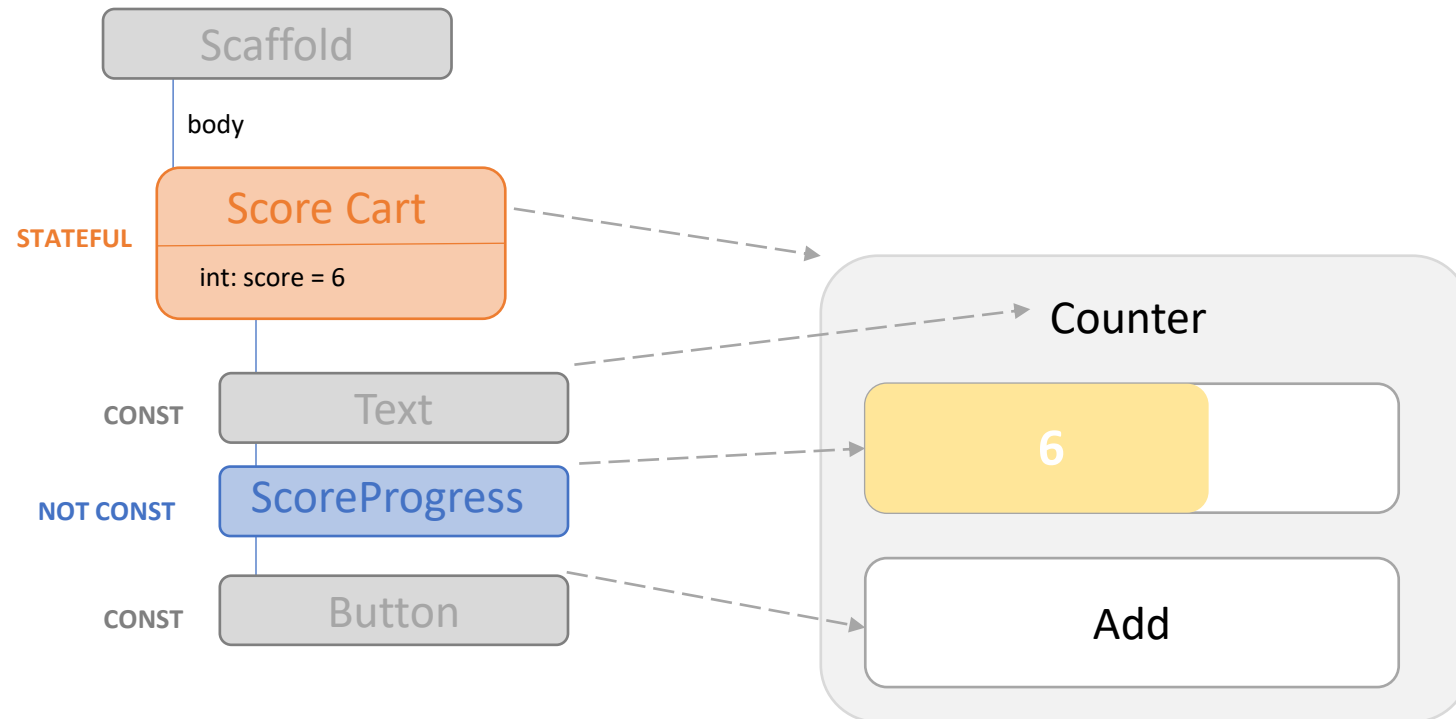
And need to be rebuild

- ✓ At the begging all **widgets are dirty**: they need to be paint for the 1st time



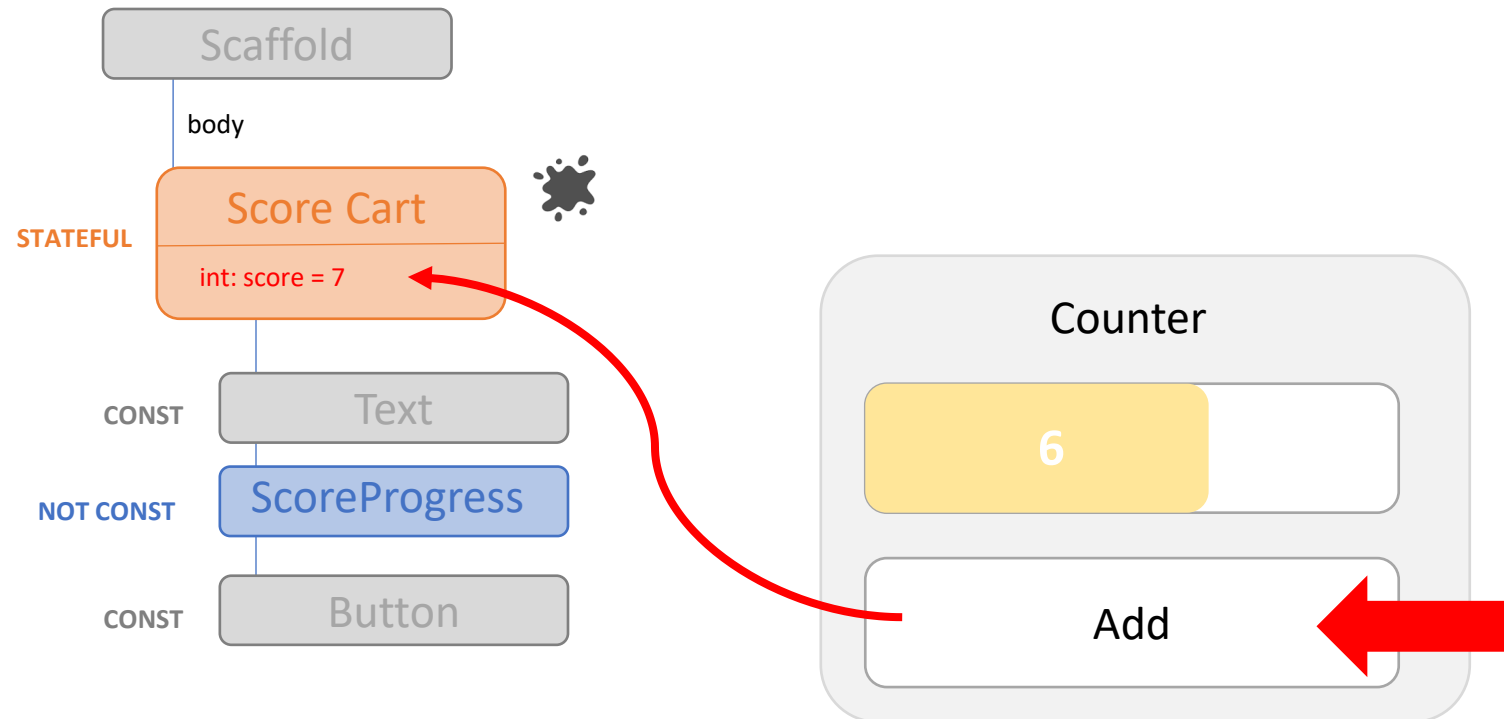
When Widget are **tagged as dirty** ? 🌟

- ✓ After building widgets, all widget are **now clean**



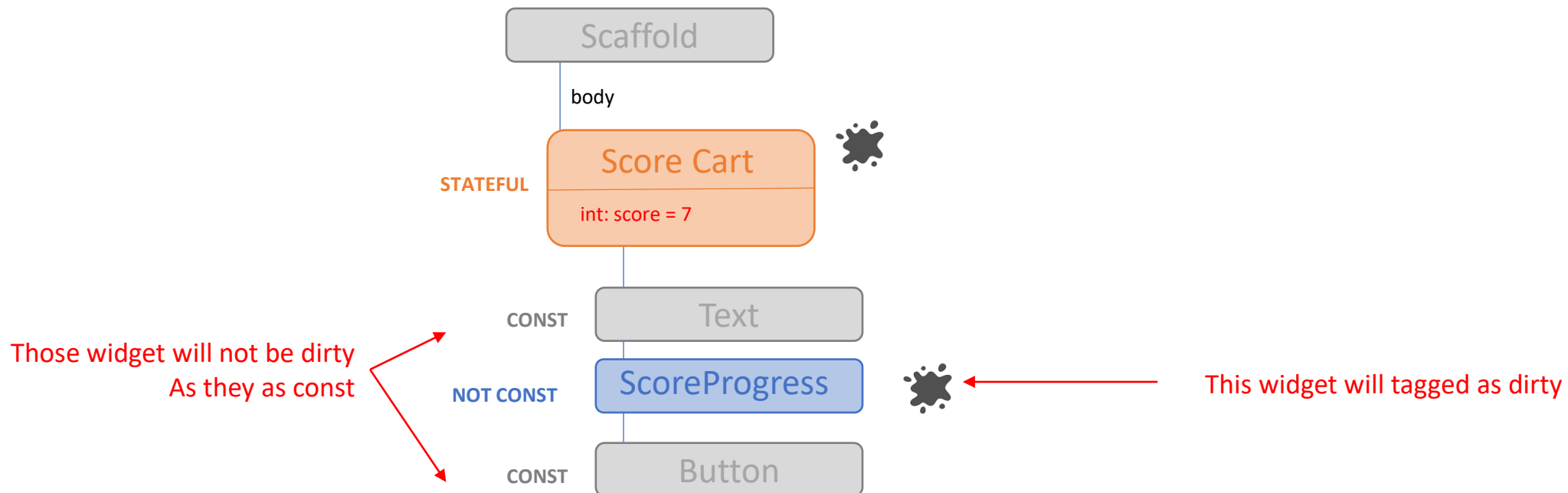
When Widget are tagged as dirty ?

- ✓ After clicking on Add, the **score state is increased**
- ✓ The **stateful widget is tagged dirty**



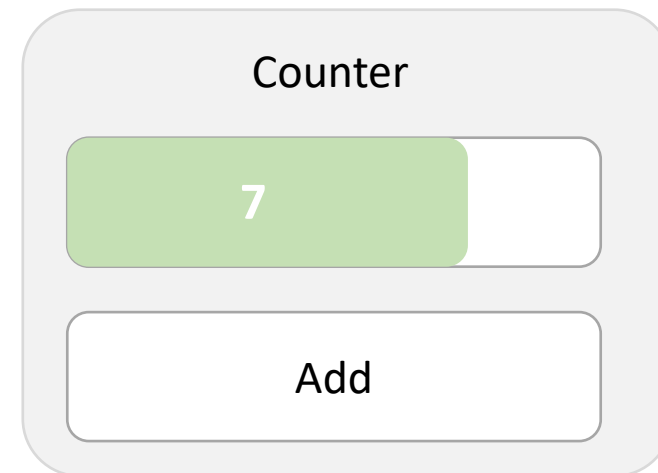
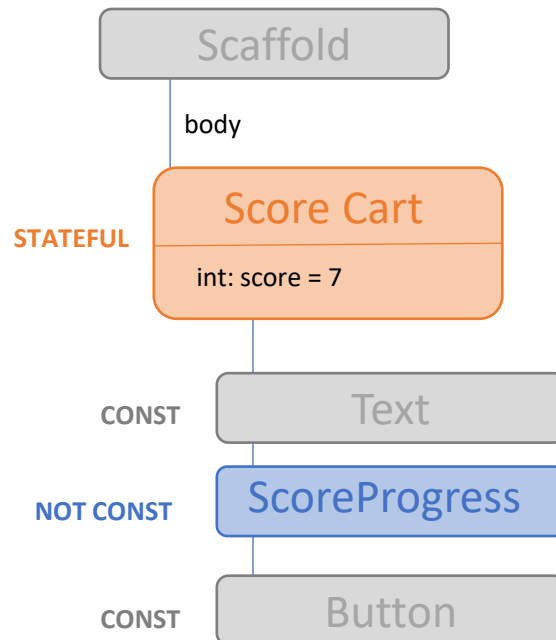
When Widget are tagged as dirty ?

- ✓ The children widgets will be marked as dirty too
- ✓ Expect the **const widgets**: as they never change !



When Widget are **tagged as dirty** ?

- ✓ The statefull widget and its sub widget **have been re-built**





Testing the **build()** calls

- ✓ Analyze when the build() is called on the 3 widgets (Counter, Label and Title)
- ✓ Remove the const on line 36 , and see the impact on the rebuild process

<https://zapp.run/edit/flutter-zeue06saeuf0?entry=lib/main.dart&file=lib/main.dart>

The screenshot displays the Flutter IDE interface. On the left, the `main.dart` file is open, showing the following code:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const MaterialApp(
6       title: 'My app',
7       home: Counter(),
8     ), // MaterialApp
9   );
10 }
11
12 class Counter extends StatefulWidget {
13   const Counter({
14     super.key,
15   });
16
17   @override
18   State<Counter> createState() => CounterState();
19 }
```

Below the code editor, the `LOGS` tab is active, showing the following messages:

```
[17:18:33] Counter widget - rebuilding...
[17:18:33] Label widget - rebuilding...
```

On the right, a browser window shows the live preview of the app at the URL `https://zeue06saeuf0.zapp.page/#/`. The app displays the text "Great counter" in a large blue font, followed by "Current value: 3" in a smaller grey font, and a blue "Increment" button below it.



Randomize the Dice !

- ✓ Get **the start code** and add the **assets** into you assets folders – Update your pubSpec !
- ✓ Change the code to **select a random dice** when clicking on the button
- ✓ *Extra : the roll should also work when clicking on the image*

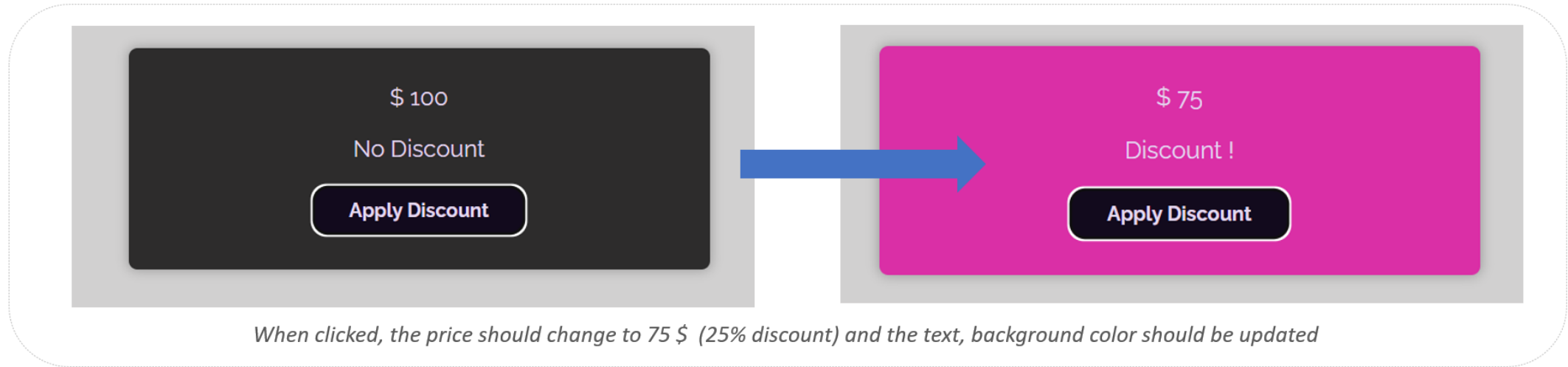


TIP : DART RANDOM

TIP : GESTURE DETECTOR

When clicking on the button, the new dice should be randomly selected (1 to 6)

How to design before coding?



STEP 1 – Identify **widgets** you need for this app

Widget name	Stateless or Stateful	(if stateful)	
Widget	Type	Arguments	State
PriceCard	Statefull	price:int	hasDiscount:bool

STEP 2 – Identify the type data for each UI element

UI Elements	Data
\$100	Computed from argument and state
No Discount	Computed from argument and state
Background color	Computed from argument and state

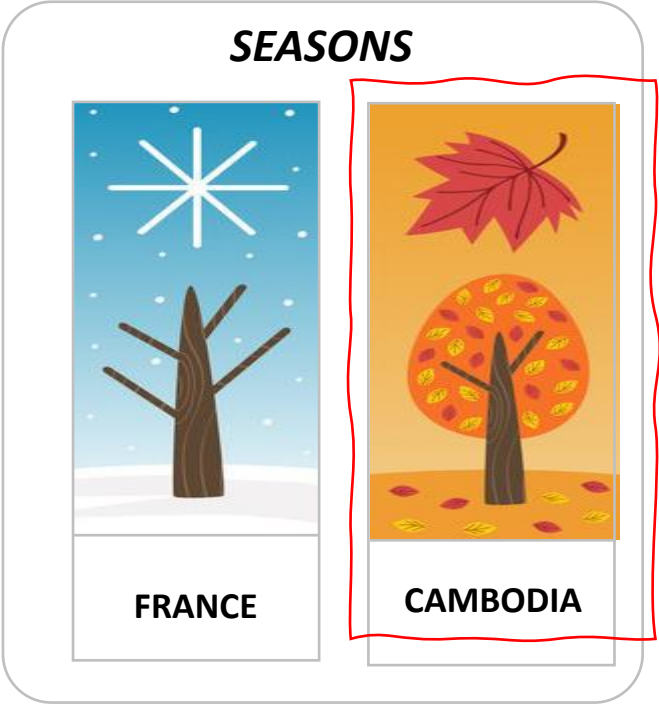
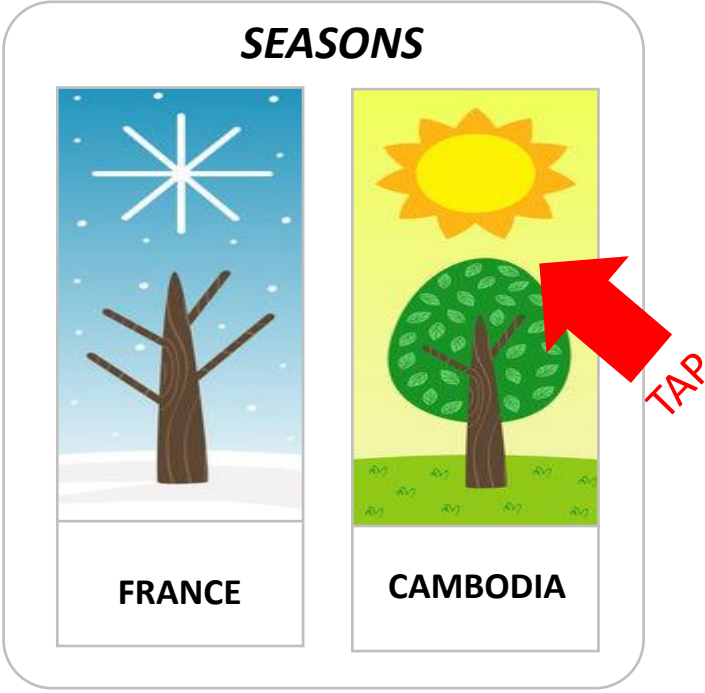
Do you know the Seasons cycle?

(winter, spring, summer, fall)





We want to display the current season in some countries



When we tap on the card, **the next season** should be display

STEP 1 – Identify **widgets** you need for this app

Widget	Type	Arguments	State

STEP 2 – Identify the type data for each UI element

UI Elements	Data
SEASONS	
CAMBODIA	
Season image	



Reflection in group

- ❓ **Why** does Flutter **make a distinction** btw stateless and statefull widgets?
- ❓ **When** do i need use a **stateless** widget or a **stateful widget** in my app?
- ❓ What if **every widgets** were **stateless.... Or stateful?**

CONGRATS !





What we have learnt today



- ✓ Attach **event** handlers using **anonymous functions** (closures) in Dart.
- ✓ **Differentiate** between **Stateless** and **Stateful** Widgets.
- ✓ Explain the **lifecycle of a Stateful Widget** in Flutter.
- ✓ **Identify use cases** where Stateful Widgets are needed.
- ✓ Create a **Stateful Widget** in Flutter.
- ✓ Use the **setState()** method to update the UI dynamically.

A traffic light **has a state** with 3 values : red, orange, green



For next week

1 Review the theory



<https://docs.flutter.dev/ui/interactivity>

<https://www.youtube.com/watch?v=AqCMFXEmf3w>

<https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddab9f2>

<https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddaba0a>

2 Code your statefull widgets



✓ Find an idea .. And code it !



Just an example !