

## Flutter: Checkbox and Radio Button widget:

A checkbox is a type of input component which holds the Boolean value. It is a GUI element that allows the user **to choose multiple options from several selections**. Here, a user can answer only in yes or no value. A marked/checked checkbox means yes, and an unmarked/unchecked checkbox means no value. Typically, we can see the checkboxes on the screen as a **square box** with white space or a tick mark. A label or caption corresponding to each checkbox described the meaning of the checkboxes.

we can have two types of checkboxes: a compact version of the Checkbox named "**checkbox**" and the "**CheckboxListTile**" checkbox, which comes with header and subtitle.

### Checkbox:

Attributes	Descriptions
value	It is used whether the checkbox is checked or not.
onChanged	It will be called when the value is changed.
Tristate	It is false, by default. Its value can also be true, false, or null.
activeColor	It specified the color of the selected checkbox.
checkColor	It specified the color of the check icon when they are selected.
materialTapTargetSize	It is used to configure the size of the tap target.

```
Checkbox(  
  value: this.showvalue,  
  onChanged: (bool value) {  
    setState() {  
      this.showvalue = value;  
    }  
  },  
),
```

```

    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        const Text(
          'Swimming:',
          textScaleFactor: 2,
        ), // Text
        Checkbox(
          // activeColor: Colors.amber,
          // checkColor: Colors.red,
          hoverColor: Colors.red,
          // focusColor: Colors.red,
          value: swimming,
          onChanged: (bool? value) {
            setState(() {
              swimming = value!;
            });
          }, // Checkbox
        ],
      ), // Row
    ],
  ), // Column
); // Scaffold
}

```

```

class _CheckBoxExampleState extends State<CheckBoxExample> {
  final _title = 'checkbox example';
  bool cycling = false;
  bool swimming = false;
  bool reading = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text(_title)),
      body: Column(
        // mainAxisAlignment: MainAxisAlignment.start,
        children: [
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text(
                'Cycling:',
                textScaleFactor: 2,
              ), // Text
              Checkbox(
                // activeColor: Colors.amber,
                // checkColor: Colors.red,
                hoverColor: Colors.red,
                // focusColor: Colors.red,
                value: cycling,
                onChanged: (bool? value) {
                  setState(() {
                    cycling = value!;
                  });
                }, // Checkbox
              ),
            ],
          ), // Row
        ],
      ),
    );
  }
}

```

GitHub link: [https://github.com/samsunk/checkbox\\_example.git](https://github.com/samsunk/checkbox_example.git)

## CheckboxListTile:

- A checkbox type which comes with **header** and **sub-title**.

### CheckboxListTile:

Attributes	Descriptions
value	It is used whether the checkbox is checked or not.
onChanged	It will be called when the value is changed.
title	It specified the main title of the list.
subtitle	It specified the subtitle of the list. Usually, it is used to add the description.
activeColor	It specified the color of the selected checkbox.
activeColor	It specified the color of the selected checkbox.
selected	By default, it is false. It highlights the text after selection.
secondary	It is the widget, which is displayed in front of the checkbox.

```
CheckboxListTile(  
  secondary: const Icon(Icons.abc),  
  title: const Text('demo mode'),  
  subtitle: Text('sub demo mode'),  
  value: this.subvalue,  
  onChanged: (bool value) {  
    setState(() {  
      this.subvalue = value;  
    });  
  },  
),
```

```

class _MyAppState extends State<MyApp> {
  bool _firstvalue = false;
  bool _secondvalue = false;
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: SafeArea(
          child: Container(
            margin: const EdgeInsets.only(top: 15),
            child: Column(
              children: [
                const Text(
                  'Checkbox with header and subtitle:',
                ), // Text
                CheckboxListTile(
                  secondary: const Icon(Icons.alarm),
                  title: const Text('Ringing at 4:30 AM everyday'),
                  subtitle: const Text('Ringing after 12 hours'),
                  value: _firstvalue,
                  onChanged: (bool? value) {
                    setState(() {
                      _firstvalue = value!;
                    });
                  },
                ), // CheckboxListTile
                CheckboxListTile(
                  secondary: const Icon(Icons.alarm),
                  title: const Text('Ringing at 5:00 AM everyday'),
                  subtitle: const Text('Ringing after 12 hours'),
                  value: _secondvalue,
                  onChanged: (bool? value) {
                    setState(() {
                      _secondvalue = value!;
                    });
                  },
                ), // CheckboxListTile
              ],
            ), // Column
          ), // Container
        ), // SafeArea
      ), // Scaffold
    ); // MaterialApp
  }
}

```

GitHub Link: [https://github.com/samsunk/checkboxlisttile\\_ex.git](https://github.com/samsunk/checkboxlisttile_ex.git)

## Flutter:RadioButton

A radio button is also known as the options button which holds the **Boolean value**. It allows the user to choose only one option from a predefined set of options. This feature makes it different from a checkbox where we can select more than one option and the unselected state to be restored. We can arrange the radio button in a **group of two or more** and displayed on the screen as **circular holes** with white space (for unselected) or a dot (for selected). We can also provide a **label** for each corresponding radio button describing the choice that the radio button represents. A radio button can be selected by clicking the mouse on the circular hole or using a keyboard shortcut.

In this section, we are going to explain how to use radio buttons in Flutter. Flutter allows us to use radio buttons with the help of '**Radio**', '**RadioListTile**', or '**ListTile**' Widgets.

The flutter radio button does not maintain any state itself. When we select any radio option, it invokes the **onChanged** callback and passing the value as a parameter. If the value and **groupValue** match, the radio option will be selected.

### Properties of Radio:

1. **Value**- It specifies the backhand value, which is represented by a radio button.
2. **group value** – It is used to specify the currently selected item for the radio button group. If the value and group value is same, it is assumed the button is checked.
3. **onChanged** – it is a callback method which is automatically invoked when the button is selected.

GitHub link - [https://github.com/samsunk/radio\\_example.git](https://github.com/samsunk/radio_example.git)