

# Flet – Dropdown: esempi concreti e MVC

Questo documento mostra esempi pratici e realistici di utilizzo del Dropdown in Flet, prima in stile semplice e poi strutturato secondo il pattern MVC (Model-View-Controller).

## 1. Esempio base: Dropdown semplice

Scenario: seleziona un colore e mostro il valore selezionato.

```
import flet as ft
```

```
def main(page: ft.Page):
```

```
    def on_change(e):
        txt.value = f"Hai scelto: {dd.value}"
        txt.update()
```

```
    dd = ft.Dropdown(
        label="Scegli un colore",
        options=[
            ft.dropdown.Option("Rosso"),
            ft.dropdown.Option("Verde"),
            ft.dropdown.Option("Blu"),
        ],
        on_change=on_change
    )
```

```
    txt = ft.Text()
    page.add(dd, txt)
```

```
ft.app(target=main)
```

## 2. Dropdown dinamica da lista Python

Le opzioni vengono create iterando una lista.

```
valori = ["A", "B", "C", "D"]
```

```
dd = ft.Dropdown(  
    options=[ft.dropdown.Option(v) for v in valori]  
)
```

### 3. Dropdown con pattern MVC

Separiamo completamente logica, dati e interfaccia.

#### 3.1 Model

```
class Model:  
    def __init__(self):  
        self._scelta = None  
  
    @property  
    def scelta(self):  
        return self._scelta  
  
    @scelta.setter  
    def scelta(self, value):  
        self._scelta = value
```

#### 3.2 View

```
import flet as ft  
  
class View:  
    def __init__(self, page: ft.Page):  
        self.page = page  
        self.controller = None  
  
    def set_controller(self, controller):  
        self.controller = controller  
  
    def build(self):  
        self.dd = ft.Dropdown(  
            label="Scegli valore",  
            options=[  
                ft.dropdown.Option("A"),  
                ft.dropdown.Option("B"),  
                ft.dropdown.Option("C"),  
            ],  
            on_change=self.controller.on_change  
        )  
  
        self.txt = ft.Text()  
        self.page.add(self.dd, self.txt)  
  
    def update_text(self, value):  
        self.txt.value = f"Scelta attuale: {value}"  
        self.txt.update()
```

#### 3.3 Controller

```
class Controller:  
    def __init__(self, view, model):  
        self.view = view  
        self.model = model  
  
    def on_change(self, e):  
        self.model.scelta = e.control.value  
        self.view.update_text(self.model.scelta)
```

## 4. Flusso MVC riassunto

1. L'utente cambia selezione nel Dropdown
2. La View intercetta l'evento e chiama il Controller
3. Il Controller aggiorna il Model
4. Il Controller ordina alla View di aggiornarsi

Questo approccio è quello consigliato per progetti grandi, esami universitari e codice pulito.