



E' una UI Component library, utile a creare interfacce grafiche in applicazioni scritte in **Angular** (e non solo).

**Concetti essenziali:**

- 1) Typography
- 2) Components
  - **Bottoni e indicatori**
  - **Navigazione**
  - **Layout**
  - Form Controls
  - Popups e Modali
  - Data Tables

## Capitolo 1 – Installazione

1) `ng add @angular/material`

- Indigo-pink
- HammerJs
- Browser Animation

Installare anche questi ultimi per avere un'esperienza completa.

**N.B:** : Il processo di installazione cambia in modo notevole a partire dalla versione 6>.

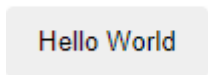
**Testiamo** la corretta installazione di **angular material** provando ad inserire un **bottone**

- 1) In App.module.ts -> import MatButtonModule
- 2) In app.component.html aggiungere V

```
<button mat-button>Hello World</button>
```

- 3) Nota se il bottone subisce differenze

Il bottone appare così:



## Capitolo 2 – Material module

La creazione di un modulo a se stante per importare i moduli del gruppo material permette una scrittura del codice molto più pulita e ordinata.

- Come importare un modulo material nel progetto

```

src > app > material > TS material.module.ts > ...
1  import { NgModule } from '@angular/core';
2  import { MatButtonModule } from '@angular/material';
3
4  const MaterialComponents = [
5    MatButtonModule
6  ]
7
8  @NgModule({
9    imports: [
10     MatButtonModule
11    ],
12    exports: [
13     MatButtonModule
14    ]
15  })
16  export class MaterialModule { }
17

```

1) Generare un nuovo modulo (material module), dentro cui utilizzare tutti gli altri moduli materiale (come, ad es. quello sul bottone creato in precedenza)

**ng g m material**

Questo genererà una cartella con un modulo dentro (material.module.ts).

2) Importarlo nel modulo app.module.ts

```
import { MaterialModule } from './material/material.module';
```

...

```
imports: [
  MaterialModule

```

...

Tutti i **moduli material** verranno inseriti nel **modulo material**, così da avere un import più pulito e definito.

## Capitolo 3 – Typography

### Modulo adibito alla gestione tipografica:

Sono varie le **classi** utilizzabili per definire stili tipografici nei nostri elementi html.

Name	Description
display-4	112px, one-off header, usually at the top of the page (e.g. a hero header).
display-3	56px, one-off header, usually at the top of the page (e.g. a hero header).
display-2	45px, one-off header, usually at the top of the page (e.g. a hero header).
display-1	34px, one-off header, usually at the top of the page (e.g. a hero header).
headline	Section heading corresponding to the <code>&lt;h1&gt;</code> tag.
title	Section heading corresponding to the <code>&lt;h2&gt;</code> tag.
subheading-2	Section heading corresponding to the <code>&lt;h3&gt;</code> tag.
subheading-1	Section heading corresponding to the <code>&lt;h4&gt;</code> tag.
body-1	Base body text.
body-2	Bolder body text.
caption	Smaller body and hint text.
button	Buttons and anchors.
input	Form input fields.

### Esempio

```
<div class="mat-display-4">Ciao sono un div!</div>
<div class="mat-display-3">Ciao sono un div!</div>
<div class="mat-display-2">Ciao sono un div!</div>
<div class="mat-display-1">Ciao sono un div!</div>
<div>Separatore</div>
<div class="mat-headline">Ciao sono un div!</div>
<div class="mat-title">Ciao sono un div!</div>
<div class="mat-subheading-2">Ciao sono un div!</div>
<div class="mat-subheading-1">Ciao sono un div!</div>
```

Risultato:

# Ciao sono un div!

## Ciao sono un div!

### Ciao sono un div!

#### Ciao sono un div!

Separatore

Ciao sono un div!

Ciao sono un div!

Ciao sono un div!

Ciao sono un div!

**Nota:** è possibile aggiungere un'unica classe ([mat-typography](#)) e poi inserire all'interno i classici tag html.

```
<div class="mat-typography">
  <h1>Ciao, sono un div!</h1>
  <h2>Ciao, sono un div!</h2>
  <h3>Ciao, sono un div!</h3>
  <h4>Ciao, sono un div!</h4>
  <h5>Ciao, sono un div!</h5>
  <h6>Ciao, sono un div!</h6>
</div>
```

## Capitolo 4 – Bottoni

Per poter usare i **material buttons** bisogna importare il modulo correlato.

Questo modulo utilizza un sistema ad **attributi** e non a classi.

Attributo basico è **mat-button** che definisce il **material** sul **bottone**.

Ecco alcuni esempi:

```
<button mat-button>Cliccami!</button>
<button mat-raised-button>Cliccami!</button>
<button mat-flat-button>Cliccami!</button>
<button mat-stroked-button>Cliccami!</button>

<button mat-icon-button>Icon</button>
<button mat-fab>Icon 2</button>
<button mat-mini-fab>Mini Fab Icon 2</button>

<button color="primary" mat-button>Primary</button>
<button color="accent" mat-button>Accent</button>
<button color="warn" mat-button>Warn</button>
```

Per evitare l'**animazione** di ripple è possibile aggiungere il **disableRipple**.

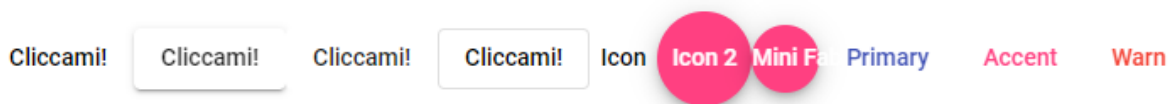


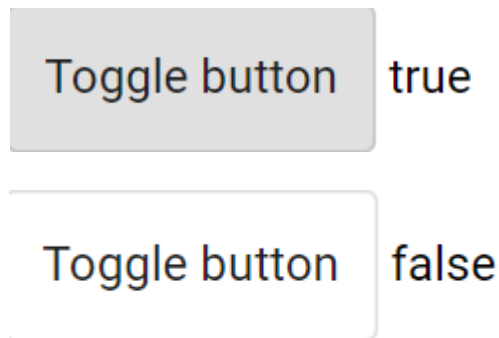
Figure 1 Esempi di bottoni con vari attributi, tra cui i colori

## Capitolo 5 – Button toggle

Animazione relativa il **bottone** che definisce uno **stato attivo** ed uno **inattivo**

```
<mat-button-toggle #toggleBtn>Toggle button</mat-button-toggle>
```

È possibile porre una variabile che si modifica in base al click



E' possibile definire un gruppo con dei toggle button interni

```
<mat-button-toggle-group #toggleGroup="matButtonToggleGroup">
  <mat-button-toggle value="Angular">Angular</mat-button-toggle>
  <mat-button-toggle value="React">React</mat-button-toggle>
  <mat-button-toggle value="Vue">Vue</mat-button-toggle>
</mat-button-toggle-group>

<h1>{{toggleGroup.value}}</h1>
```

Nota: è possibile usare l'attributo **multiple** per effettuare il toggle di più valori.

## Capitolo 6 – Icons

- 1) Import aggiungendo il link all'index.js dell'intero progetto

```
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
```

- 2) Import del modulo in MaterialModule

Definirne una in pagina è molto semplice

```
<mat-icon>grade</mat-icon>
```

Basta usare il tag è una keyword tra le quelle disponibili

## Capitolo 7 – Badges



### Cosa sono?

Piccoli descrittori giustapponibili ad elementi html

- Usano un attributo base con un valore inseribile:

```
matBadge="7"
```

- Un attributo per definire la sua posizione

```
matBadgePosition="before"
```

- Uno per la dimensione

```
matBadgeSize="large"
```

- Uno per il colore

```
matBadgeColor="green"
```

- Uno per non far andare il badge sopra il testo

```
matBadgeOverlap="false"
```

è possibile dare un valore dinamico al value del matBadge, usando un sistema di **[binding]**.

```
notifications: number = 2;
```

...

```
[matBadge]="notifications"
```

È possibile far scomparire il badge quando il valore di notifications è pari ad una certa condizione

```
[matBadgeHidden]="notifications === 0"
```



## Capitolo 8 – Progress Spinner



### Cosa sono?

Sono elementi che ruotano per indicare un caricamento

### Sintassi:

- Spinner progressivo a %

```
<mat-progress-spinner value="100"></mat-progress-spinner>
```

Click me!

- Spinner rotante

```
<mat-progress-spinner value="100"></mat-progress-spinner>  
<mat-spinner *ngIf="showSpinner"></mat-spinner>
```

## Capitolo 9 – Barra di navigazione

Tag adibito alla creazione di una **navbar**

```
<mat-toolbar color="primary" class="navbar">
```

Logo

Home About Services

## Capitolo 10 – Sidenav

Una **barra di navigazione laterale** rapportabile con un **contenuto relativo**

```
<mat-sidenav-container>
  <!-- Sidenav -->
  <mat-sidenav mode="push" [(opened)]= "opened">Sidenav</mat-sidenav>
  <!-- Contenuto -->
  <mat-sidenav-content>Main
    <button (click)="opened = !opened">Toggle Button</button>
  </mat-sidenav-content>
</mat-sidenav-container>
```

Esistono varie **modalità** di interazione tra contenuto e sidenav:

Mode:

- side
- push
- over (di default)

## Capitolo 11 – Menu

Cosa è?

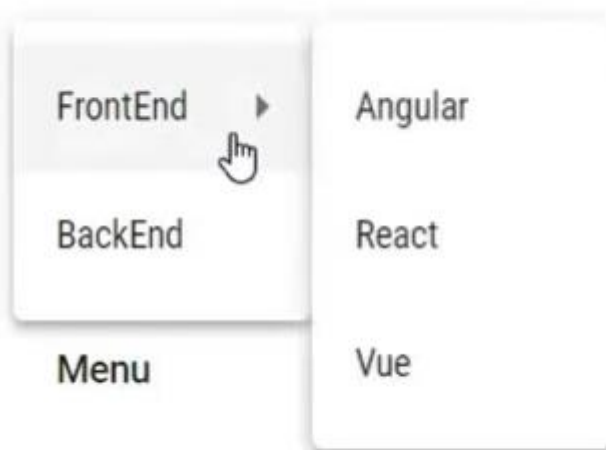
Indovina !?!

Utilizza il tag `<mat-menu>`

I bottoni interni usano l'attributo `mat-menu-item`

Usando un **binding** è possibile, togliendo un valore ed aprire/chidere il menu

```
<mat-menu #appMenu="matMenu">
  <button mat-menu-item>Front-End</button>
  <button mat-menu-item>Back-End</button>
</mat-menu>
<button mat-flat-button [matMenuTriggerFor]="appMenu" >Toggle Menu</button>
```



Oltre a ciò, è possibile anche creare un **sottomenu** (#submenu) da aprire in conseguenza di un **menu di gerarchia superiore**.

## Capitolo 12 – Liste

Modulo adibito alla gestione di liste

Tags utilizzabili:

```
<mat-list>
```

```
<mat-nav-list>
```

Tags/ attributi utilizzabili per inserire degli “items” interni

```
<mat-list-item>Item 1</mat-list-item>
```

```
<a mat-list-item href="#">Home</a>
```

**Nota:** esiste l’attributo **matLine** utilizzabile per mettere inline, “singolarmente”, un elemento.

- **matDivider** : Tags che permette di inserire una linea divisoria.

## Capitolo 13 – Grids

Permette la creazione di griglie customizzabili

È strutturato con un container che conterrà i tiles:

```
Container → <mat-grid-list cols="2">
```

Tiles

```
<mat-grid-tile rowspan="2">
```

Attributi per container:

**Cols:** definisce numero colonne della griglia.

**rowHeight:** altezza riga dei tiles.

**gutterSize:** equivalente al gap.

---

Attributi per items:

**Rowspan:** definisce quante righe deve prendere il tile specifico.

**Colspan:** definisce quante colonne deve prendere il tile specifico.

Esempio:

```
<mat-grid-list cols="2" rowHeight="50px">
  <mat-grid-tile rowspan="2">Tile 1</mat-grid-tile>
  <mat-grid-tile>Tile 2</mat-grid-tile>
  <mat-grid-tile colspan="2">Tile 3</mat-grid-tile>
  <mat-grid-tile>Tile 4</mat-grid-tile>
  <mat-grid-tile>Tile 5</mat-grid-tile>
  <mat-grid-tile>Tile 6</mat-grid-tile>
</mat-grid-list>
>
  <mat-grid-tile>Tile 5</mat-grid-tile>
  <mat-grid-tile>Tile 6</mat-grid-tile>
</mat-grid-list>
```

## Capitolo 14 – Expansion Panel

Pannello utilizzato per espandere un contenuto

Struttura:

Header

|

+ --- Titolo

+ --- Descrizione

Contenuto

Sezione azione

|

+ --- Bottone

```
<mat-expansion-panel>

  <!-- Testa visibile del pannello -->
  <mat-expansion-panel-header>
    <mat-panel-title>Titolo</mat-panel-title>
    <mat-panel-description>Descrizione</mat-panel-description>
  </mat-expansion-panel-header>

  <!-- Contenuto -->
```

```

<p>Questa è il contenuto dell'intero pannello</p>

<!-- Sezione dedicata alle azioni -->
<mat-action-row>
  <button mat-button>Cliccami, non mordo!</button>
</mat-action-row>

</mat-expansion-panel>

```

## Visione :

Titolo	Descrizione	
Questa è il contenuto dell'intero pannello		
		Cliccami, non mordo!

Figure 2 Pannello espanso



## Accordion

E' possibile unire 2 pannelli insieme in un unico macrocomponente, l'accordion, capace di "relazionare" i 2 componenti, facendoli collassare!

Titolo	Descrizione	
Titolo 2	Descrizione 2	
Questa è il contenuto dell'intero pannello 2		
		Cliccami, non mordo!

Figure 3 Accordion collapsing

## Capitolo 15 – Cards



### Cosa sono?

Blocchi adibiti a mostrare un contenuto, titolo e sezione “attiva”.

```
<mat-card>
  <mat-card-title>Titolo</mat-card-
title>
  <mat-card-content>Contenuto</mat-card-
content>
  <mat-card-actions align="end">
    <button mat-raised-
button>Cliccami, sono bello</button>
  </mat-card-actions>
</mat-card>
```

### Struttura:

**Titolo**

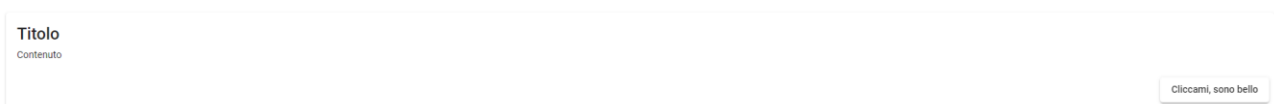
**Contenuto**

**Sezione azione**

|

+ --- **Bottone**

Attributo **align** → Permette di posizionare la sezione azione.



## Capitolo 16 – Tabs

### Cosa sono?

Blocchi adibiti alla separazione grafica di sezioni della UI

```
<mat-tab-group #tabRef (selectedTabChange)="log(tabRef.selectedIndex)">
  <mat-tab label="Angular">Angular</mat-tab>
  <mat-tab label="React">React</mat-tab>
  <mat-tab label="Vue">Vue</mat-tab>
</mat-tab-group>
{{tabRef.selectedIndex}}
// Semplice raggruppamento di 3 tabs
```

**Nota:** È possibile, in aggiunta, identificare l'indice della tab selezionata, aggiungendo un semplice #, oltre che allegarci una funzione.

## Esempi di moduli Material UI

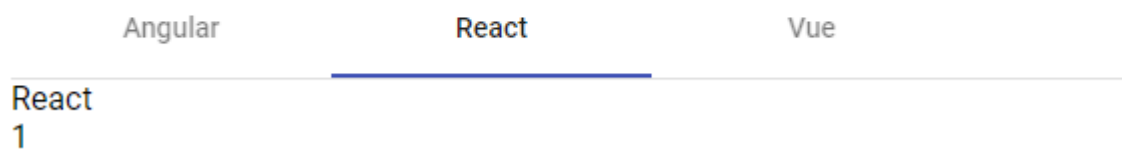


Figure 4 L'animazione consta di uno sliding orizzontale, con cambio dell'indice visualizzato

## Capitolo 17 – Stepper



Figure 5 Step by step (๖^\_๖)

### Cosa sono?

Gli stepper sono elementi che, tramite una transizione, verticale o orizzontale, permettono, **“passo dopo passo”**, di eseguire una serie di operazioni consequenziali, come una **registrazione**.



Può avere una struttura verticale o **orizzontale**

```
<mat-horizontal-stepper>
  <!-- Sezione 1 -->
  <mat-step label="Nome">
    <p>Nome</p>
  </mat-step>
  <!-- Sezione 2 -->
  <mat-step label="Cognome">
    <p>Cognome</p>
  </mat-step>
  <!-- Sezione 3 -->
  <mat-step label="Indirizzo">
    <p>Indirizzo</p>
  </mat-step>
</mat-horizontal-stepper>
```

// Es. di struttura orizzontale

## Esempi di moduli Material UI

1 Nome — 2 Cognome — 3 Indirizzo

Nome

È possibile inserire dei bottoni adibiti al passaggio alla prossima sezione o alla precedente, inserendo gli attributi **matStepperPrevious**, **matStepperNext**.

Attributo **linear**: permette di visualizzare come già lette le sezioni visitate.

Attributi **linear** e **completed**: permette di visualizzare come già lette le sezioni visitate, rispettivamente in stepper e step.

---

Fine sezione Layout