

```
Angular-Material-Light-dark-System-Color-mode > Project_Code > src > app > components > header > TS header.c
  40   </mat-toolbar>
  41   ,
  42   styles: [
  43     '@use \'@angular/material\' as mat;';
  44
  45     mat-toolbar {
  46       display: flex;
  47       align-items: center;
  48
  49       @include mat.toolbar-overrides(
  50         (
  51           container-background-color: var(--mat-sys-primary),
  52           container-text-color: var(--mat-sys-on-primary),
  53         )
  54       );
  55
  56       @include mat.icon-button-overrides(
  57         (
  58           icon-color: var(--mat-sys-on-primary),
  59         )
  60       );
  61     }
  62
  63     .selected-theme {
  64       @include mat.menu-overrides(
  65         (
  66           item-icon-color: var(--mat-sys-primary),
  67           item-label-text-color: var(--mat-sys-primary),
  68         )
  69       );
  70     }
  71
  72     .flex-stretch {
  73       flex: 1 0 auto;
  74     }
  75   ,
  76 }
  77 export class HeaderComponent {
  78   protected themeService = inject(ThemeService);
  79 }
```

TS content.component.ts X

Angular-Material-Light-dark-System-Color-mode > Project_Code > src > app > components > content > TS content.

```
92     styles: `<br>
93       .content {<br>
94         padding: 32px;<br>
95         min-height: 100vh;<br>
96       }<br>
97       background-color: var(--mat-sys-background);<br>
98       color: var(--mat-sys-on-background);<br>
99     }<br>
100    }<br>
101    .responsive-grid {<br>
102      display: grid;<br>
103      grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));<br>
104      grid-gap: 1rem;<br>
105    }<br>
106    }<br>
107    .flex-row {<br>
108      display: flex;<br>
109      flex-direction: row;<br>
110      gap: 1rem;<br>
111    }<br>
112    }<br>
113    mat-card {<br>
114      max-width: 400px;<br>
115    }<br>
116    }<br>
117    h3 {<br>
118      font-weight: 400;<br>
119    }<br>
120    ,<br>
121  })<br>
122 export class ContentComponent {<br>
123   private dialog = inject(MatDialog);<br>
124   }<br>
125   showDialog(): void {<br>
126     this.dialog.open(SampleDialogComponent, {<br>
127       width: '500px',<br>
128     });<br>
129   }<br>
130 }<br>
131 }
```

```
1  @use "@angular/material" as mat;  
2  
3  html {  
4      @include mat.theme(  
5          (  
6              color: mat.$azure-palette,  
7              typography: Roboto,  
8          )  
9      );  
10 }  
11  
12 .accent {  
13     @include mat.theme(  
14         (  
15             color: mat.$magenta-palette,  
16         )  
17     );  
18 }  
19  
20 .warn {  
21     @include mat.theme(  
22         (  
23             color: mat.$red-palette,  
24         )  
25     );  
26 }  
27  
28 html,  
29 body {  
30     height: 100%;  
31 }  
32 body {  
33     margin: 0;  
34     font-family: Roboto, "Helvetica Neue", sans-serif;  
35 }  
36  
37 h2,  
38 p {  
39     padding: 16px;  
40 }  
41
```

TS theme.service.ts ×

Angular-Material-Light-dark-System-Color-mode > Project_Code > src > app > services > TS theme.service.ts > ThemeService

```
1 import { Injectable, computed, effect, inject, signal } from '@angular/core';
2
3 export interface AppTheme {
4   name: string;
5   icon: string;
6 }
7
8 @Injectable({
9   providedIn: 'root',
10 })
11 export class ThemeService {
12   private appTheme = signal<'light' | 'dark' | 'system'>('system');
13
14   private themes: AppTheme[] = [
15     { name: 'light', icon: 'light_mode' },
16     { name: 'dark', icon: 'dark_mode' },
17     { name: 'system', icon: 'desktop_windows' },
18   ];
19
20   selectedTheme = computed(() =>
21     this.themes.find((t) => t.name === this.appTheme())
22   );
23
24   getThemes() {
25     return this.themes;
26   }
27
28   setTheme(theme: 'light' | 'dark' | 'system') {
29     this.appTheme.set(theme);
30   }
31
32   constructor() {
33     effect(() => {
34       const appTheme = this.appTheme();
35       const colorScheme = appTheme === 'system' ? 'light dark' : appTheme;
36       document.body.style.setProperty('color-scheme', colorScheme);
37     });
38   }
39 }
40
```

AngularDarkMode x Angular-Material-Light-dark-Sy - +

localhost:4200

Required YouTube Socet.io chat APP Movies PortFolio http://localhost:3000/ Home | Hostinger Header All Bookmarks

Angular Dark Mode

Buttons

Open Dialog Accent button Warn button

Input

Enter some information

Checkbox

Check this box!

Expansion panel

Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse >  X

Styles Computed Layout Event Listeners >

Filter :hover .cls + ↻

```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> ...
    <body class="mat-typography" style="color-scheme: light;">
      <app-root _ngHost-0-0-c2283532919 ng-version= "19.0.4" >...</app-root> == $0
        <script src="polyfills.js" type="module"></script>
        <script src="main.js" type="module"></script>
        <div class="cdk-overlay-container"></div>
    </body>
</html>
```

Inherited from body.mat-typography

style attribute { color-scheme: light; }

AngularDarkMode

Angular-Material-Light-dark-Sy

localhost:4200

Required YouTube Socet.io chat APP Movies PortFolio http://localhost:3000/ Home | Hostinger Header All Bookmarks

Angular Dark Mode

Buttons

Open Dialog Accent button Warn button

Input

Enter some information

Checkbox

Check this box!

Expansion panel

A card

Shiba Inu
Dog Breed

The screenshot shows a web browser window with the Angular Dark Mode application running at localhost:4200. The interface is dark-themed. On the left, there are sections for Buttons (with three buttons: Open Dialog, Accent button, Warn button), Input (with a text input field containing 'Enter some information'), and Checkbox (with a checked checkbox labeled 'Check this box!'). Below these is an Expansion panel section. On the right, there is a 'A card' component featuring a Shiba Inu dog standing in grass, with the text 'Shiba Inu' and 'Dog Breed' above it. A red box highlights the top-right corner of the browser window.

Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse > Styles Computed Layout Event Listeners >

Filter :hover .cls +

```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> ... </head>
    <body class="mat-typography" style="color-scheme: dark;">
      <app-root _ngcontent-ng-c2283532919 ng-version="19.0.4"> ... </app-root> == $0
        <script src="polyfills.js" type="module"></script>
        <script src="main.js" type="module"></script>
        <div class="cdk-overlay-container"></div>
      </body>
    </html>
```

element.style { color-scheme: dark; }

Inherited from body.mat-typography

style attribute { color-scheme: dark; }

This screenshot shows the developer tools' Elements tab with the DOM tree expanded to show the body element. A red box highlights the 'style="color-scheme: dark;"' attribute. The right-hand panel shows the 'Styles' tab with the CSS rule 'color-scheme: dark;' listed under 'Inherited from body.mat-typography'.

AngularDarkMode

Angular-Material-Light-dark-Sy

localhost:4200

Required YouTube Socet.io chat APP Movies PortFolio http://localhost:3000/ Home | Hostinger Header All Bookmarks

Angular Dark Mode

Buttons

Open Dialog Accent button Warn button

Input

Enter some information

Checkbox

A card

Shiba Inu
Dog Breed

The screenshot shows the same web browser window as the first one, but the interface is now light-themed. The left sidebar has the same sections: Buttons, Input, and Checkbox. The 'A card' component on the right also appears different, possibly due to a theme switch or a different component state. A red box highlights the top-right corner of the browser window.

Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse > Recorder Styles Computed Layout Event Listeners >

Filter :hover .cls +

```
<!DOCTYPE html>
<html lang="en"> scroll
  <head> ... </head>
    <body class="mat-typography" style="color-scheme: light dark;"> == $0
      <app-root _ngcontent-ng-c2283532919 ng-version="19.0.4">
        <app-header _ngcontent-ng-c2283532919 _ngcontent-ng-c2910022382> ... </app-header>
        <app-content _ngcontent-ng-c2283532919 _ngcontent-ng-c2533749858>
          <div _ngcontent-ng-c2533749858 class="content responsive-grid"> grid
            <div _ngcontent-ng-c2533749858>Buttons</div>
            <div _ngcontent-ng-c2533749858 class="flex-row"> ... </div> flex
            <br _ngcontent-ng-c2533749858>
            <br _ngcontent-ng-c2533749858>
            <h3 _ngcontent-ng-c2533749858>Input</h3>
          </div>
        </app-content>
      </app-root>
    </html>
```

element.style { color-scheme: light dark; }

body { margin: 0; font-family: Roboto, "Helvetica Neue", sans-serif; }
html, body { height: 100%; }
body { display: block; margin: 0px+; }

styles.scss:32
styles.scss:28
user agent stylesheet

This screenshot shows the developer tools' Elements tab with the DOM tree expanded to show the body element. A red box highlights the 'style="color-scheme: light dark;"' attribute. The right-hand panel shows the 'Styles' tab with the CSS rule 'color-scheme: light dark;' listed under 'element.style { color-scheme: light dark; }'. It also lists global styles for 'body' and 'html, body'.