

Style i motywy. Material Design

mgr inż. Stanisław Lota



Style i motywy

Style i motywy w systemie Android umożliwiają oddzielenie szczegółów projektu aplikacji od struktury i zachowania interfejsu użytkownika, podobnie jak w przypadku arkuszy stylów w projektowaniu stron internetowych.

Android umożliwia definiowanie wyglądu i stylu, na przykład kolorów i czcionek, składników systemu Android w plikach zasobów XML.

W ten sposób możemy ustawić atrybuty związane ze stylem w jednym centralnym miejscu w Android Studio w `res/values/` w pliku o nazwie `styles.xml`.

Styl

Styl określa atrybuty dla danego rodzaju widoku. Na przykład jeden styl może określać atrybuty przycisku.

Każdy atrybut określony w stylu jest atrybutem, który można ustawić w pliku układu.

Przykład definicji stylu

```
<resources>  
<style name="AppTheme"  
parent="Theme.AppCompat.Light.NoActionBar">  
<!-- Customize your theme here. -->  
<item name="colorPrimary">@color/colorPrimary</item> <item  
name="colorPrimaryDark">@color/colorPrimaryDark</item> <item  
name="colorAccent">@color/colorAccent</item>  
<item name="android:windowBackground">@color/colorBackground  
</item> </style>  
</resources>
```

Tworzymy styl

Aby zdefiniować zestaw stylów, musimy utworzyć nowy plik XML w katalogu **/res/values** w naszym projekcie.

Aby stworzyć **styl** w pliku XML, musimy wykonać poniższe kroki.

- Musimy dodać element **<style>** w pliku XML z atrybutem **name**, aby jednoznacznie identyfikować styl.
- Aby zdefiniować atrybuty **stylu**, musimy dodać elementy **<item>** o nazwie definiującej atrybut stylu i musimy dodać odpowiednią wartość do każdego **elementu <item>**.
- W systemie Android możemy zdefiniować wiele stylów za pomocą elementu **<style>** z atrybutem **name**, aby jednoznacznie identyfikować styl w pliku XML.

Przykład

```
<style name="MyAppStyle">  
  <item name="android:layout_width">wrap_content</item> <item  
  name="android:layout_height">wrap_content</item> <item  
  name="android:textColor">#434230</item>  
  <item name="android:background">#c6ecc6</item>  
  <item name="android:typeface">serif</item>  
  <item name="android:textSize">15sp</item>  
  <item name="android:textStyle">bold</item>  
  <item name="android:padding">10dp</item>  
</style>
```

Przykład definicji stylu

Widzimy tu trzy pozycje: `colorPrimary` o wartości `@color/colorPrimary`, `colorPrimaryDark` o wartości `@color/colorPrimaryDark` oraz `colorAccent` o wartości `@color/colorAccent`. Wszystkie kolory są zdefiniowane w pliku *colors.xml* w lokalizacji *app/res/values/*.

Jeżeli zajrzemy do tego pliku, to zobaczymy nasze trzy kolory

```
<resources>  
<color name="colorPrimary">#3F51B5</color>  
<color name="colorPrimaryDark">#303F9F</color>  
<color name="colorAccent">#FF4081</color>  
</resources>
```

Lista atrybutów

`android:typeface` — krój pisma, z czego do wyboru mamy: `normal` (zwykłą czcionkę), `serif` (czcionkę szeryfową (z ozdobnikami w postaci poprzecznie lub ukośnie zakończonych głównych kresek liter danego kroju pisma)), `sans` (czcionkę bezszeryfową (bez ozdobników)), `monospace` (czcionkę o stałej szerokości znaków).

`android:textStyle` — odmiana pisma, z czego do wyboru mamy: `normal` (tekst zwykły), `bold` (tekst pogrubiony), `italic` (tekst pochylony (kursywę)).

`android:textColor` — kolor tekstu,

`android:background` — kolor tła,

Motyw

Motyw jest stylem stosowanym do danej aktywności lub aplikacji, a nie do pojedynczego widoku. Definiuje zbiór nazwanych zasobów, które mogą być przywoływane przez style, układy widgety.

Motywy przypisują nazwy semantyczne, takie jak `colorPrimary`, do zasobów Androida.

Sposób, w jaki definiuje się tematy, jest identyczny jak w przypadku stylów w pliku `styles.xml`.

Przykład definicji motywu

```
<resources>  
<style name="AppTheme"  
parent="Theme.AppCompat.Light.NoActionBar"> <!-- Customize your  
theme here. --> <item  
name="colorPrimary">@color/colorPrimary</item> <item  
name="colorPrimaryDark">@color/colorPrimaryDark</item> <item  
name="colorAccent">@color/colorAccent</item> <color  
name="colorBackground">#42A5F5</color>  
</style>  
</resources>
```

Motyw

W pliku AndroidManifest.xml w katalogu app/manifests. ustawiony jest motyw.

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.devillecloud.myapplication">
```

```
<application
```

```
    android:allowBackup="true" android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name" android:supportsRtl="true"
    android:theme="@style/AppTheme">
```

```
    <activity android:name=".MainActivity" android:theme="@style/AppTheme">
```

```
        <intent-filter>
```

```
            <action android:name="android.intent.action.MAIN" />
```

```
            <category android:name="android.intent.category.LAUNCHER" /> </intent-filter>
```

```
        </activity> </application>
```

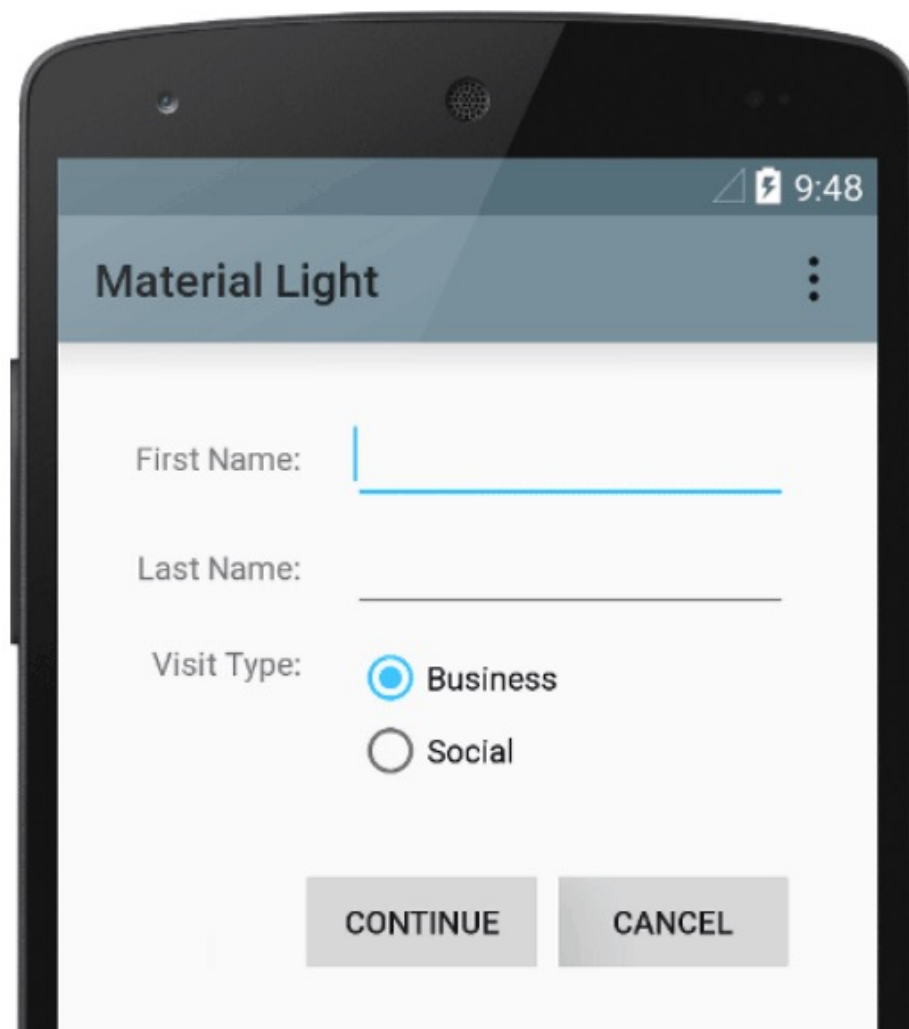
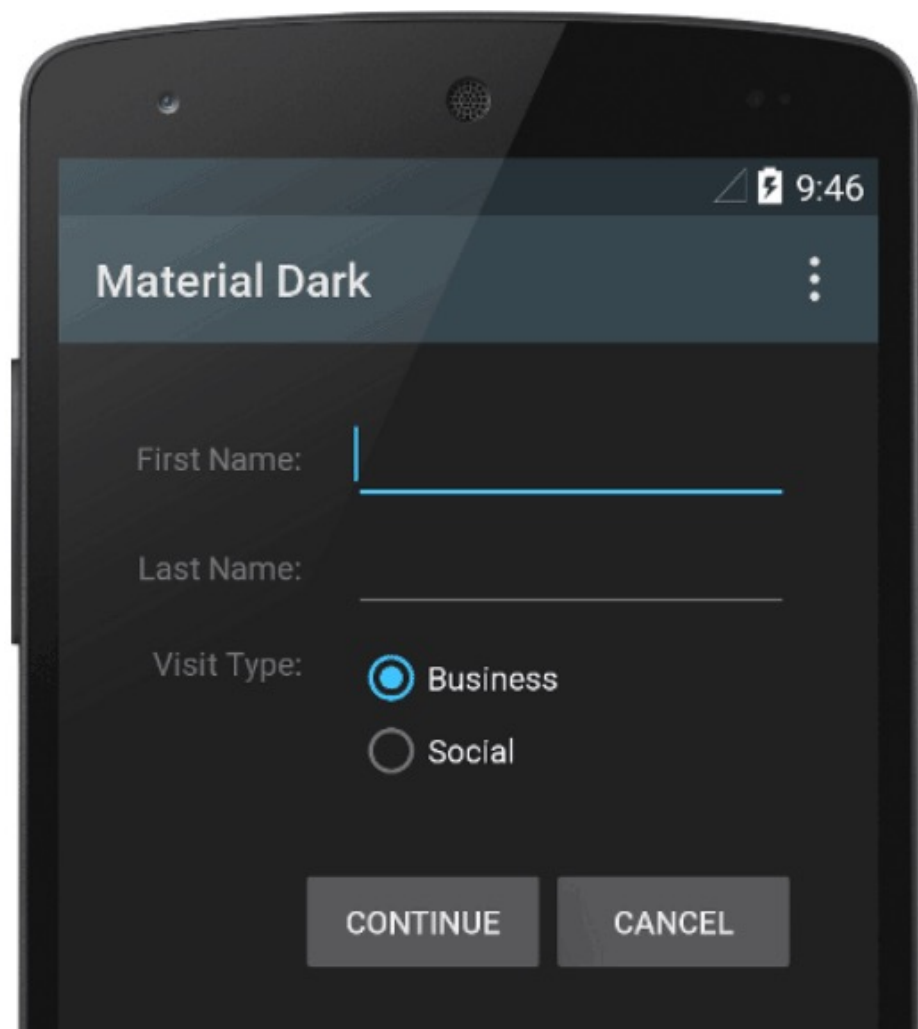
```
</manifest>
```

Styl a motyw

Jeśli wpis w pliku zasobów jest używany do nadania stylu widokowi, określa się go jako styl. Jeśli jest używany do stylizacji działania lub aplikacji, nazywany jest motywem.

Style i motywy obsługują dziedziczenie przy użyciu atrybutu nadrzędnego znacznika stylu. W ten sposób styl dziedziczy wszystkie ustawienia ze stylu macierzystego i może nadpisać wybrane atrybuty.

Styl i motyw



Material Design

Material Design to oficjalny styl(język) graficzny stworzony przez Google i docelowo dedykowany platformie Android. Jest łatwy do dostosowania, czytelny i prosty. Łączy zasady dobrego projektowania z innowacyjnością.

Fundament Material Design opiera się na solidnych podstawach wypracowanych przez lata przez designerów oraz UI/UX designerów.

Material Design

Wytyczne dotyczące projektowania interfejsu użytkownika Androida zmieniły się na przestrzeni lat. Pierwsza duża zmiana w projektowaniu aplikacji na Androida pojawiła się wraz z wydaniem Androida 3.0 (Honeycomb), które wprowadziło styl Holo .

W 2014 roku Material Design ogłoszono oficjalnym stylem produkowanych przez Google aplikacji mobilnych(Wydanie Androida 5.0 (Lollipop)), między innymi takich jak Gmail, Google Drive, Google Docs. Szybko także inni twórcy zaczęli wykorzystywać jego założenia podczas tworzenia swoich projektów.

Material Design

Dokumentacja Material Design jest stale uaktualniana i zawiera mnóstwo sekcji omawiających podstawowe koncepcje i zastosowania takich jak:

- Kolorystyka i Typografia
- Ruch i Animacja
- Wzorce projektowania
- Tworzenie responsywnych interfejsów
- I wiele więcej...

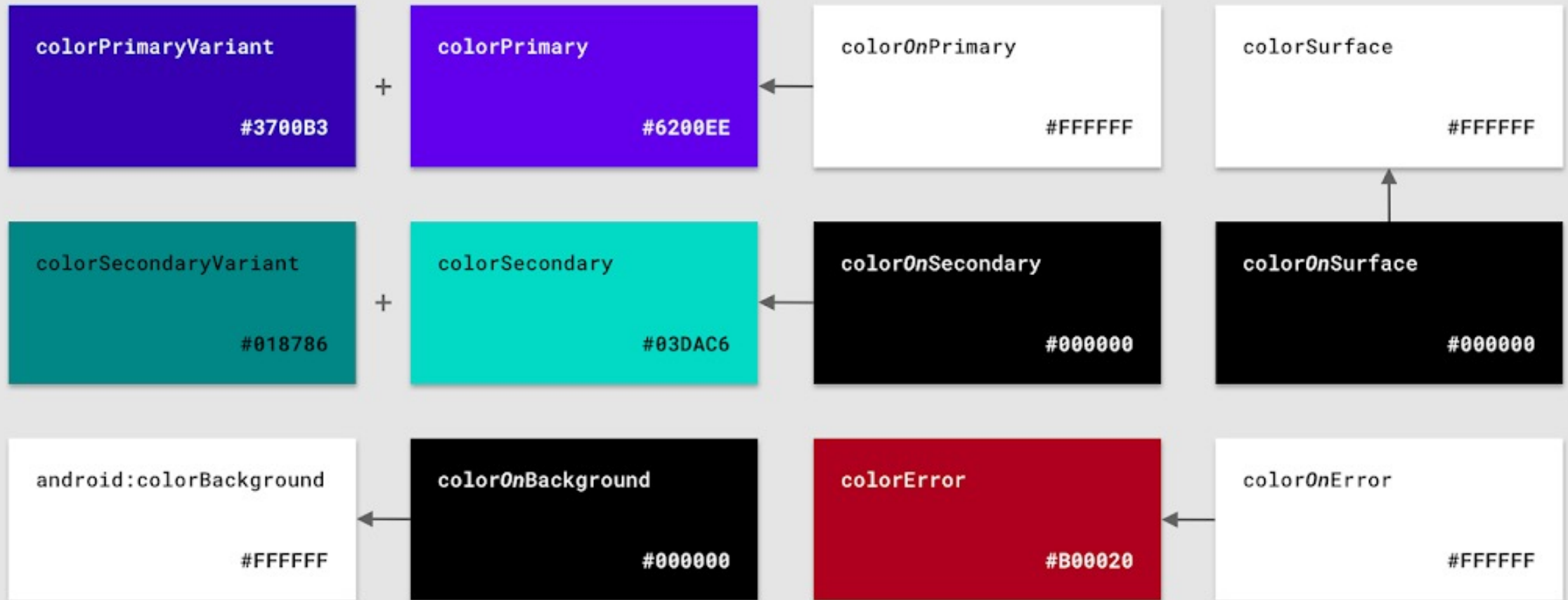
Dokumentację znajdziesz pod tym linkiem : <https://material.io/>

Kolory

Ważną cechą Material Design są kolory, które powinny być żywe oraz wyróżniające się.

Google udostępnia nam do dyspozycji palety dobrze dobranych kolorów, z których powinniśmy wybrać jeden podstawowy oraz drugi dodatkowy, dobrze odznaczający się na tle tego pierwszego, aby wyszczególnić ważniejsze elementy w aplikacji.

Kolory



Kolory

Do stworzenia palet kolorystycznych dla aplikacji i stron internetowych wykorzystuje się dwa główne kolory. Podstawowy stanowi bazę do tworzenia kolejnych jego wariacji i oznaczony jest jako 500. Na jego podstawie powinno się wygenerować kolory oznaczone od 50 – 900, gdzie te od 50 – 400 są od niego jaśniejsze, a 600 – 900 ciemniejsze

Dzięki takiemu podejściu aplikacja staje się żywsza, a ważne elementy dobrze odznaczają się na tle innych, wciąż pozostając z nimi w harmonii.

500	#F44336
50	#FFEBEE
100	#FFCDD2
200	#EF9A9A
300	#E57373
400	#EF5350
500	#F44336
600	#E53935
700	#D32F2F
800	#C62828
900	#B71C1C

A100	#FF8A80
A200	#FF5252
A400	#FF1744
A700	#D50000

500	#E91E63
50	#FCE4EC
100	#F8BBD0
200	#F48FB1
300	#F06292
400	#EC407A
500	#E91E63
600	#D81B60
700	#C2185B
800	#AD1457
900	#880E4F

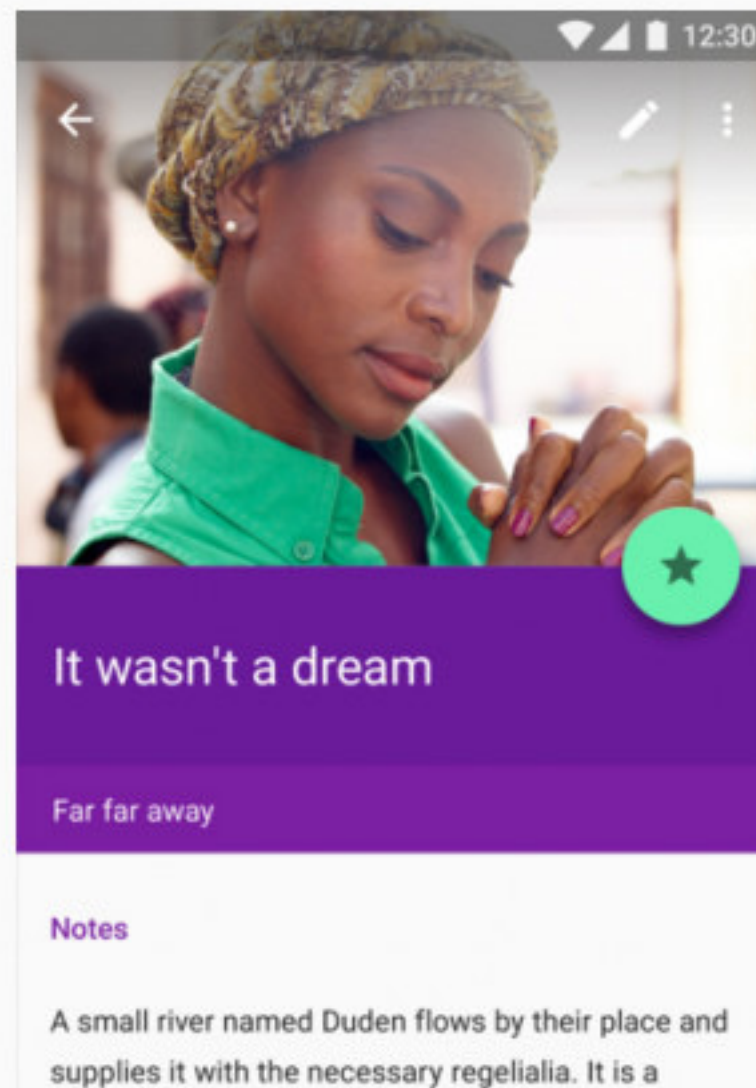
A100	#FF80AB
A200	#FF4081
A400	#F50057
A700	#C51162

500	#9C27B0
50	#F3E5F5
100	#E1BEE7
200	#CE93D8
300	#BA68C8
400	#AB47BC
500	#9C27B0
600	#8E24AA
700	#7B1FA2
800	#6A1B9A
900	#4A148C

A100	#EA80FC
A200	#E040FB
A400	#D500F9
A700	#AA00FF

Przykład

Primary — Purple	
500	#9B26AF
700	#7A1EA1
800	#691A99
Accent — Green	
A200	#68EFAD



Kolory

Drugi kolor (dodatkowy) powinien zdecydowanie odznaczać się na tle podstawowego i być wykorzystywany do wyróżniania istotnych elementów interaktywnych projektu, takich jak przyciski akcji czy linki. Powinien on także pochodzić z palety dodatkowej (oznaczonej na rysunku wyżej literami A) i być zdecydowanie bardziej nasycony.

Kolory

Zasady material design dotyczą także kolorów tekstów wykorzystywanych w interfejsie.

Podstawowy zawsze powinien bazować na czerni RGB (#000000) i być rozjaśniany za pomocą przezroczystości do 87% dla tekstu głównego, 54% dla dodatkowego i 38% dla nieaktywnego, przypisów oraz ikon.

Teksty wykorzystywane na ciemnych tłach bazować powinny natomiast na bieli i być rozjaśniane odpowiednio do 100%, 70% i 50%.

Krój pisma

Podstawowym krojem pisma wykorzystywanym w większości produktów Google jest Roboto. Po raz pierwszy pojawił się w systemie Android Ice Cream Sandwich, a następnie zmodyfikowano go na potrzeby systemu Android Lollipop i w takiej formie obowiązuje do dziś.

Podstawowa skala hierarchii tekstu powinna opierać się na wielkościach 12, 14, 16, 20 i 34. W specyficznych sytuacjach można oczywiście wykorzystywać większe i mniejsze znaki, należy jednak wtedy odpowiednio manipulować grubością tekstu, aby zachować wizualną spójność z pozostałymi elementami.

THIN

Roboto

BLACK SMALL CAPS

SUNGLASSES

ITALIC

Self-driving robot ice cream truck

BOLD

Fudgesicles only 25¢

BOLD CONDENSED

ICE CREAM

MEDIUM

Marshmallows & almonds

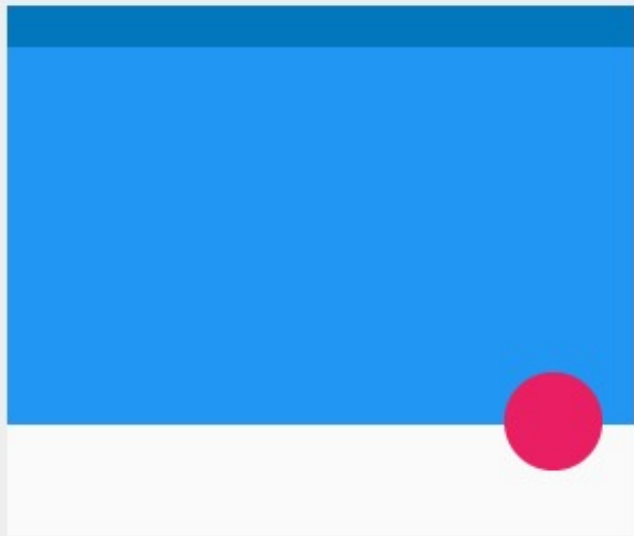
LIGHT

#9876543210

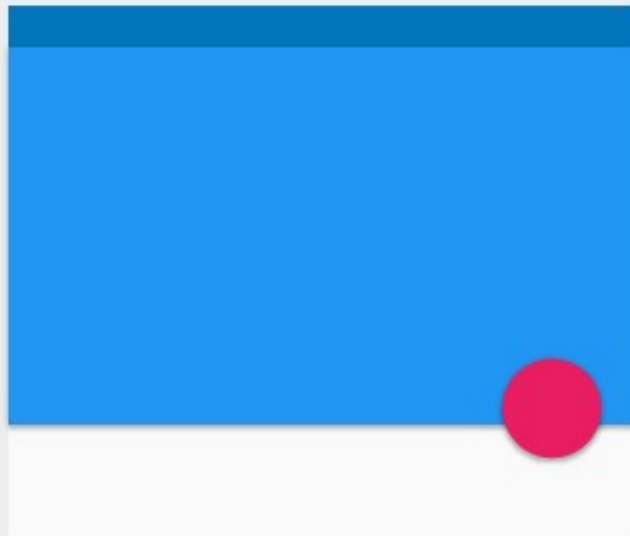
Cienie

Dość istotnym elementem Material Design są cienie, które mają wskazywać na położenie elementów w przestrzeni trójwymiarowej i zarazem określać ich hierarchię. Elementy, które wydają się znajdować wyżej od innych, są odbierane jako ważniejsze do podjęcia akcji.

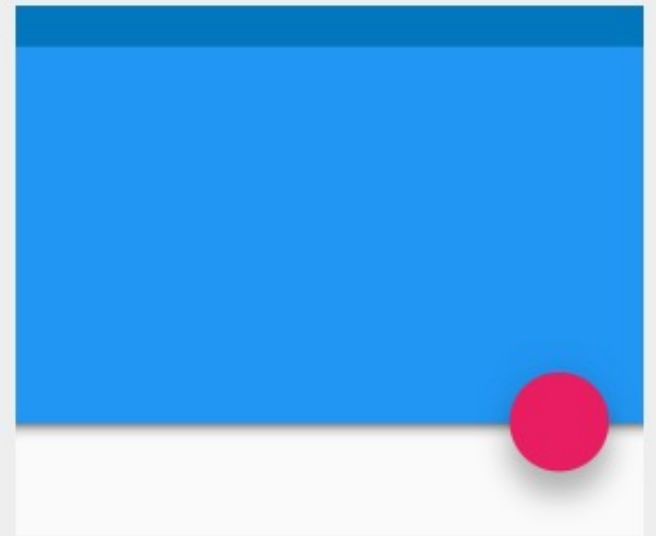
Cienie



Don't.



Don't.



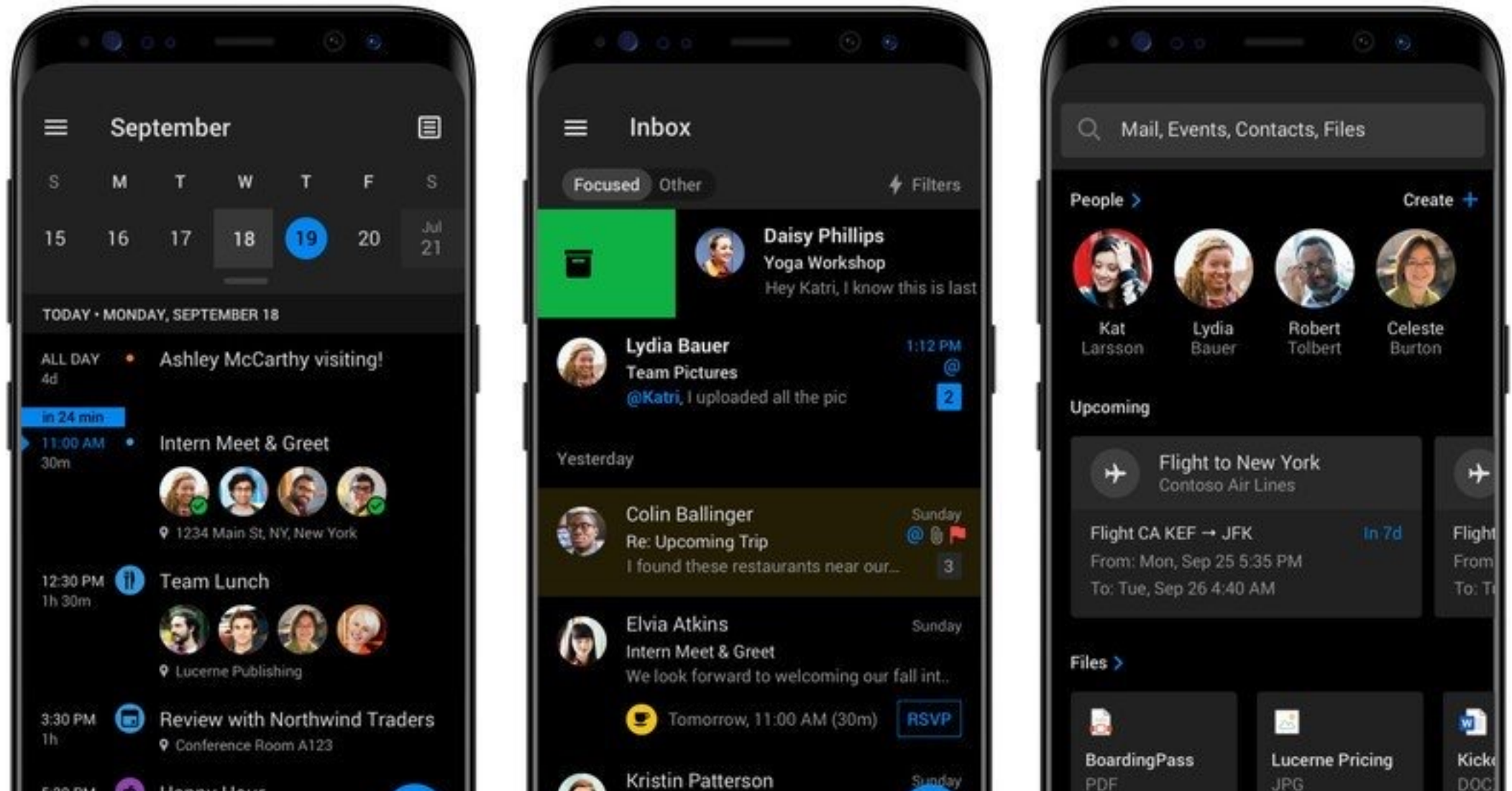
Do.

Dark Mode

Ciemny motyw jest dostępny od systemu Android 10 dotyczy zarówno interfejsu użytkownika systemu Android, jak i aplikacji uruchomionych na urządzeniu.

Głównymi zaletami jest to, że może znacznie zmniejszyć zużycie energii (w zależności od technologii ekranu urządzenia). Poprawia widoczność dla użytkowników słabo widzących i wrażliwych na jasne światło. Ułatwia każdemu korzystanie z urządzenia w słabym oświetleniu.

Dark Mode



Ikony

Aby jednak wszystkie ikony były spójne, należy opierać się na kilku podstawowych zasadach ich tworzenia:

- powinna zostać zachowana maksymalna prostota ikon,
- unikać powinno się wszelkich skomplikowanych kształtów, które mogłyby zostać wypaczone w małej rozdzielczości,
- wszelkie zakończenia linii powinny być proste,
- zaokrąglenia rogów ikon powinny opierać się na stałej proporcji określonej przez Google jako 2dp,

Ikony

- tą samą wielkość (2dp) powinny mieć także wszystkie światła wewnętrzne, wycięcia i znaki podziału pomiędzy elementami tworzącymi ikonę,
- należy unikać stylizowania ikon na trójwymiarowe, pochylania ich i nadawania im głębi,
- wszystkie kształty powinny wywodzić się z podstawowych kształtów geometrycznych, nie być zbyt skomplikowane.

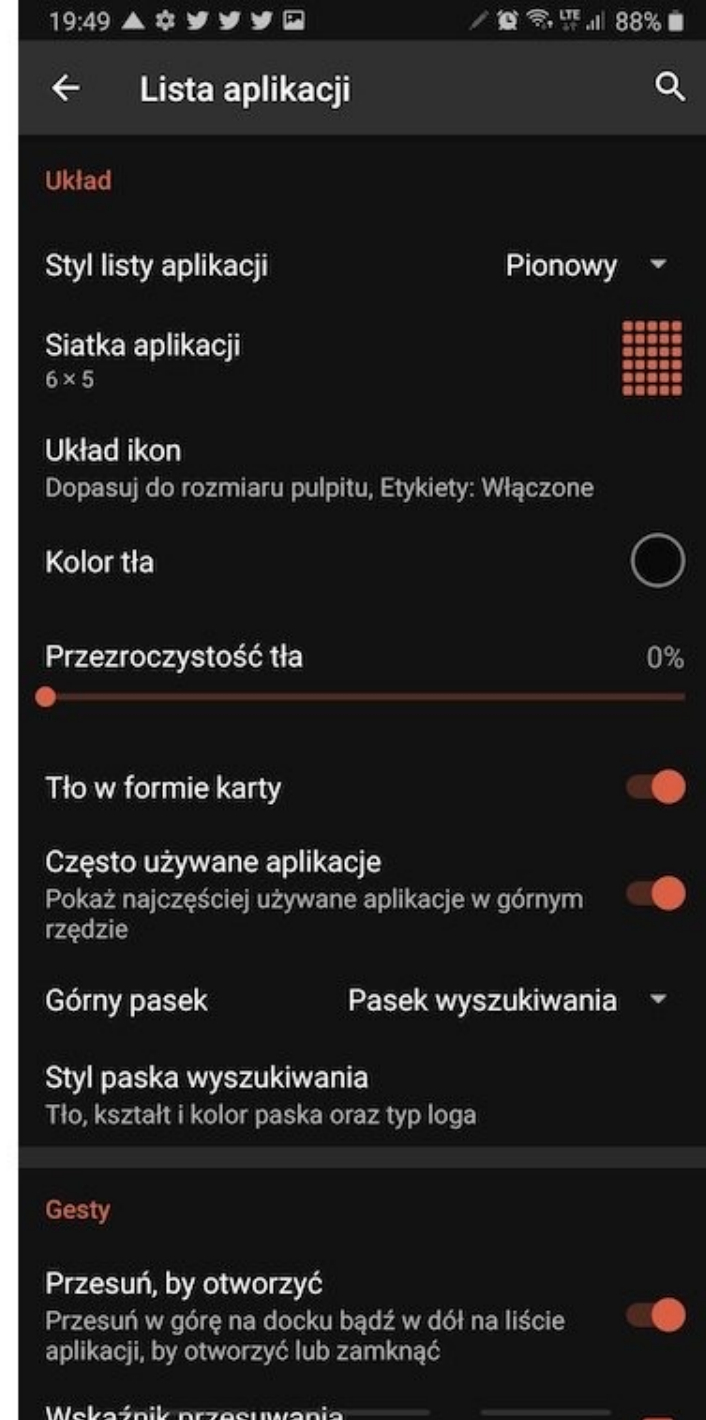
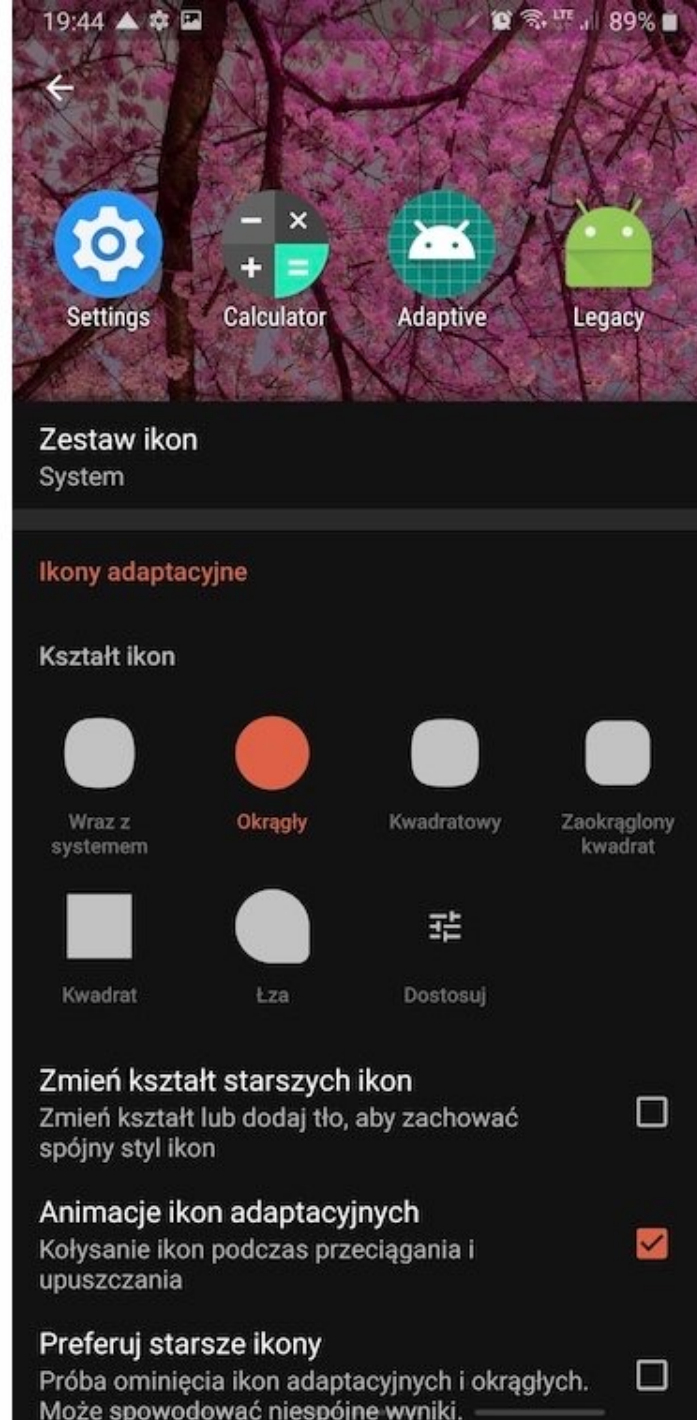
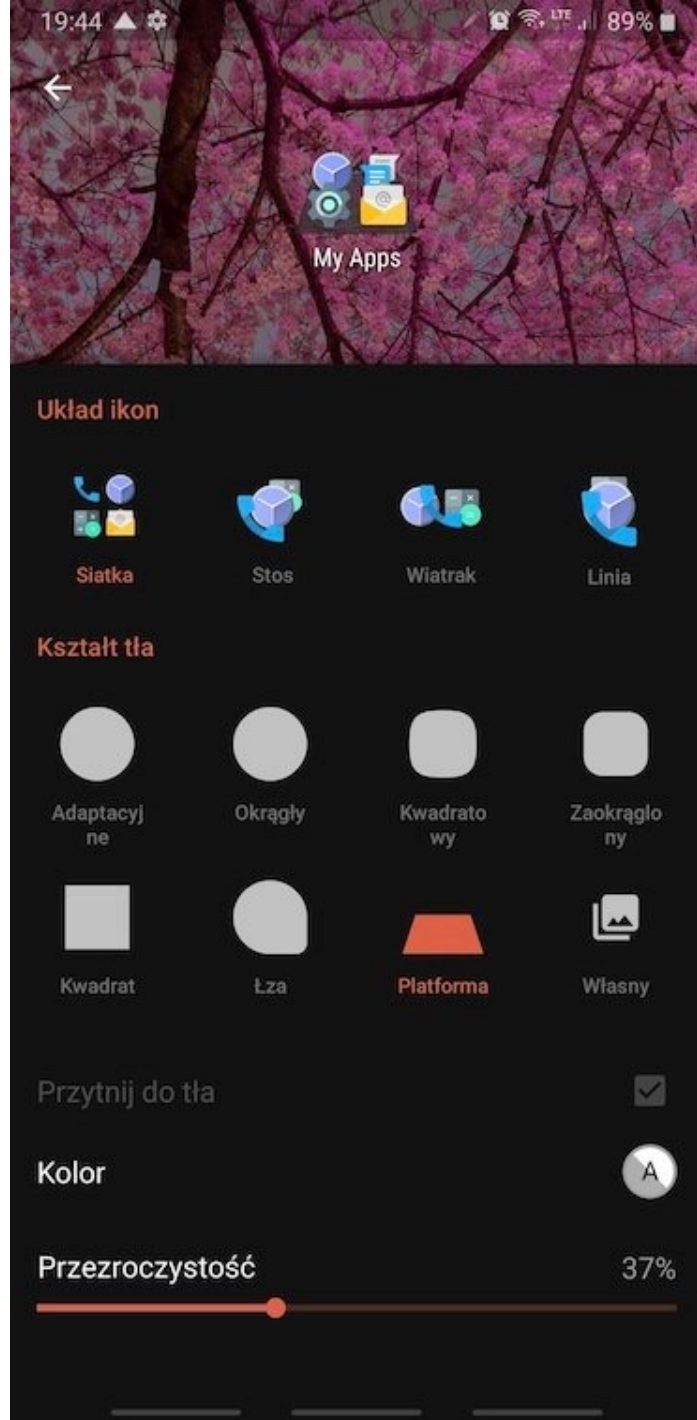
Adaptacyjne ikony

W wersji Android 8.0 (poziom API 26) wprowadzono adaptacyjne ikony uruchamiania, które mogą wyświetlać różne kształty w różnych modelach urządzeń.

Ikona adaptacyjnego programu uruchamiającego może wyświetlać okrągły kształt na jednym urządzeniu, a na innym urządzeniu wyświetlać ikonę w postaci kwadratu.

Adaptacyjne ikony uruchamiania są również używane np. w skrótach, aplikacji Ustawienia, oknach dialogowych.

<https://developer.android.com/design/downloads/index.html>



Zadanie

Wykonaj oficjalne ćwiczenie z Google Developer :

<https://developer.android.com/codelabs/kotlin-android-training-styles-and-themes#0>

Dodatkowe ćwiczenia :

<https://material.io/resources/tutorials#android-kotlin>