1.5 Цвета

Цвета

# Значение цвета

В Flet существуют 2 способа задания цветового свойства: через шестнадцатеричное значение и через именованные цвета.

### Шестнадцатеричное значение

Шестнадцатеричное значение должно быть в формате #aarrggbb (0xaarrggbb) или #rrggbb (0xeeggbb). В случае, если aa (прозрачность) не указана, она устанавливается как ff (не прозрачный).

c1 = ft.Container(bgcolor='#ff0000')

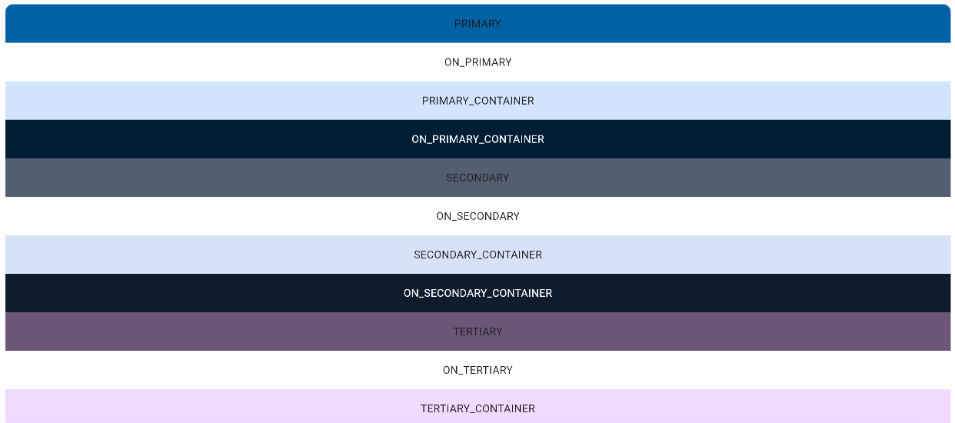
### Именованные цвета

Именованные цвета - это цвета темы *Material Design* и цветовые палитры. Их можно задать с помощью строкового значения или используя модуль flet.colors .

c1 = ft.Container(bgcolor=ft.colors.YELLOW)

c2 = ft.Container(bgcolor='yellow')

# Цвета темы

В имеется 30 именованных цветов темы, которые

theme.color\_scheme

генерируются на основе свойства color\_scheme\_seed (значение по умолчанию -

"blue" ("синий")).

page.theme = theme.Theme(color\_scheme\_seed='green')

page.update()

Любой из 30 цветов можно переопределить, в таком случае они будут иметь абсолютное значение, которое не будет зависеть от seed .

page.theme = ft.Theme(

color\_scheme=ft.ColorScheme(

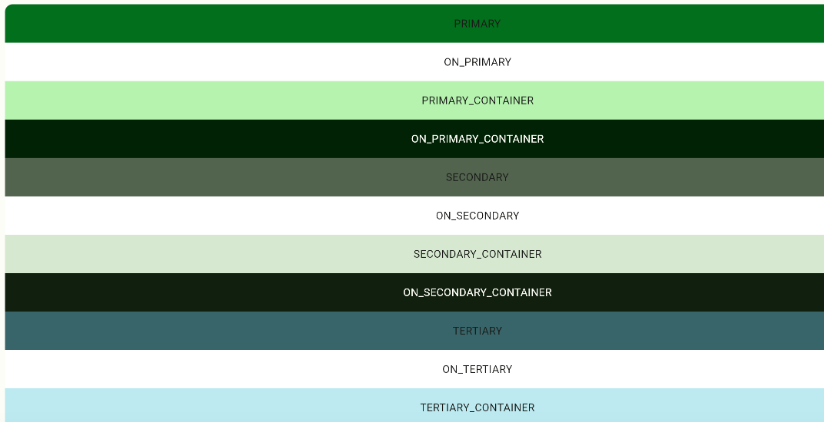
primary=ft.colors.GREEN,

primary\_container=ft.colors.GREEN\_200

# ...

),

)

Цвета темы определяют цвета по умолчанию для большинства элементов управления Flet.

# Цветовые палитры

Цветовые палитры, изначально созданные Material Design в 2014 году, состоят из цветов, предназначенных для гармоничного сочетания друг с другом.

Цветовые образцы (палитры) состоят из разных оттенков определенного цвета.

100

900

Большинство образцов имеют оттенки от

до

с шагом в сотню, плюс

цвет 50 . Чем меньше число, тем бледнее цвет. Чем больше число, тем темнее цвет. У

акцентных образцов (например, ) только

700

redAccent

значения 100 , 200 , и .

400

Кроме того, доступна серия черных и белых цветов с общими значениями

прозрачности. Например, - это чистый черный цвет с прозрачностью **54%**.

black54

Цвета палитры могут быть использованы для задания цвета отдельных элементов

управления или как цвет для генерации цветов темы.

seed

# Прозрачность цвета

Вы можете указать прозрачность для любого цвета (шестнадцатеричного значения

или именованного) с помощью метода . Значение прозрачности

with\_opacity

должно быть между (полностью прозрачный) и 1.0 (не прозрачный).

0.0

color = ft.colors.with\_opacity(0.5, ft.colors.PRIMARY)

color = ft.colors.with\_opacity(0.5, '#ff6666')

Еще один способ задать прозрачность для строкового значения:

color = "surface,0.5"

Для шестнадцатеричного значения можно указать канал aa со значениями

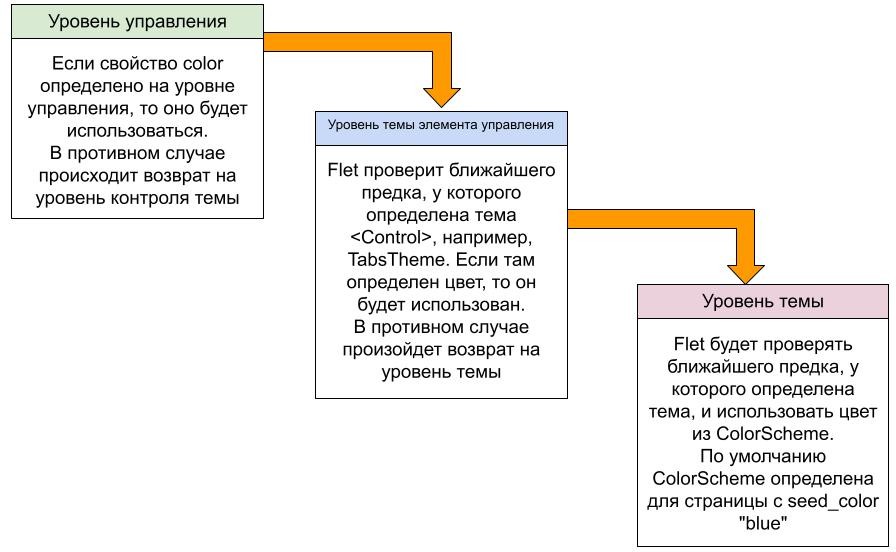
от до ff , например:

00

color = "#7fff6666"

# Определение цветов для элементов управления

**Flet**

Большинство элементов управления Flet имеют цвета по умолчанию, определенные color\_scheme, которые могут быть переопределены на разных уровнях.

# Уровень элемента управления

Если цвет определен на уровне элемента управления, он будет использован.

c = ft.Container(width=100, height=100, bgcolor=ft.colors.GREEN\_200)

Не каждый элемент управления Flet имеет свойство цвета, которое можно задать на

уровне элемента управления. Например, у определенный темой ( theme ) ближайшего предка .

FilledButton

всегда есть цвет "*primary*",

# Уровень темы элемента управления

Для (используется в прокручиваемых элементах

ScrollBar

управления: Page , View , Column , Row , и

ListView

и GridView ),

Tabs

Text

controls , Flet проверит, есть ли у ближайшей предка темы *ScrollBar Theme*, *Tabs theme* или *Text theme*.

*ПРИМЕЧАНИЕ*

*Если вам нужно изменить тему для определенного элемента управления ScrollBar, Text или Tabs, вы можете обернуть этот элемент управления в контейнер и*

*text\_theme*

*tabs\_theme*

*настроить scrollbar\_theme , контейнера.*

# Уровень темы

*или*

*для этой темы*

Flet проверит наличие темы ( theme ) у ближайшего предка и возьмет цвет

из . В примере ниже, ближайшим предком

ColorScheme

для FilledButton является Container , и основной цвет, который будет использоваться для кнопки, будет взят из темы контейнера.

import flet as ft

def main(page: ft.Page):

container = ft.Container(

width=200,

height=200,

border=ft.border.all(1, ft.colors.BLACK),

content=ft.FilledButton("Основной цвет"),

theme=ft.Theme(color\_scheme=ft.ColorScheme(primary=ft.colors.YELLOW)))

page.add(container)

ft.app(target=main)