3.3 Доступность

Доступность

Flet основан на Flutter, который включает в себя обширную поддержку доступности на уровне фреймворка, помимо той, что предоставляется базовой операционной системой.

**Средства чтения с экрана**

Для мобильных устройств, средства чтения с экрана (TalkBack, VoiceOver) позволяют пользователям с нарушениями зрения получать голосовую обратную связь о содержании экрана и взаимодействовать с пользовательским интерфейсом с помощью жестов на мобильных устройствах и горячих клавиш в десктопном варианте. Включите VoiceOver или TalkBack на своем мобильном устройстве и перемещайтесь с их помощью по вашему приложению .

Для веб-версии в настоящее время поддерживаются средства чтения с экрана: Мобильные браузеры:

* iOS - VoiceOver
* Android - TalkBack Десктопные версии:
* MacOS - VoiceOver
* Windows - JAWs & NVDA

Пользователи таких средств чтения с экрана на веб-страницах должны активировать опцию "Включить доступность" для построения семантического дерева.

## Текст

Используйте свойство элемента управления

|  |  |
| --- | --- |
| Text.semantics\_label | |
| Text | по умолчанию. |

## Кнопки

для переопределения семантики

Все кнопки с текстом на них генерируют правильную семантику.

Используйте свойство для добавления семантики экранного диктора для

tooltip

кнопок IconButton , FloatingActionButton и PopupMenuButton .

и

TextField

Dropdown

Используйте свойства

и

для добавления

семантики средств чтения с экрана к этим элементам управления.

TextField.label

Dropdown.label

## Кастомная (пользовательская) семантика

Для любых специфических требований используйте элемент управления Semantics .

## Отладка семантики

Установите в значение True , чтобы показать

page.show\_semantics\_debugger

наложение, которое показывает информацию о доступности, сообщаемую фреймворком.

Вы можете реализовать определенную комбинацию клавиш для удобного переключения отладчика семантики во время разработки приложения:

import flet as ft

def main(page: ft.Page):

page.title = "Пример счетчика Flet"

page.vertical\_alignment = ft.MainAxisAlignment.CENTER

page.horizontal\_alignment = ft.CrossAxisAlignment.CENTER

def on\_keyboard(e: ft.KeyboardEvent):

print(e)

if e.key == "S" and e.ctrl:

page.show\_semantics\_debugger = not

page.show\_semantics\_debugger

page.update()

page.on\_keyboard\_event = on\_keyboard

txt\_number = ft.Text("0", size=40

def button\_click(e):

txt\_number.value = str(int(txt\_number.value) + 1)

page.update()

page.add(

txt\_number,

ft.Text("Нажмите CTRL+S, чтобы переключить отладчик

семантики"),

ft.FloatingActionButton(

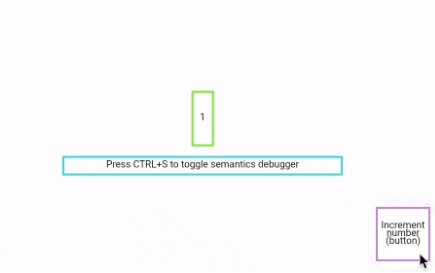
icon=ft.icons.ADD, tooltip="Увеличить число",

on\_click=button\_click

),

)

ft.app(target=main, view=ft.WEB\_BROWSER)

В этом примере кода, мы создали простое приложение с функцией увеличения числа по нажатию на кнопку. Кроме того, мы добавили возможность включения и выключения отладчика семантики при помощи комбинации клавиш CTRL+S. Это может быть полезно для проверки, как ваше приложение взаимодействует с семантическими дикторами и какие данные о доступности оно предоставляет.