



O T U S

ОНЛАЙН-ОБРАЗОВАНИЕ

Онлайн-образование

СКАЧАНО С WWW.SW.HELP - ПРИСОЕДИНЯЙСЯ!

Проверить, идет ли запись!





Меня хорошо видно && слышно?

Ставьте , если все хорошо
Напишите в чат, если есть проблемы

Цели вебинара

1

Научиться обрабатывать жесты.

2

Научиться работать с Checkbox, Switch, Radio виджетами

3

Научиться работать с виджетами ввода текста.

The image features a central horizontal band with a blue-to-purple gradient. Overlaid on this band is a white network pattern of interconnected lines and nodes. The background of the entire image is an aerial view of a city skyline, with the top and bottom portions showing a dense cluster of skyscrapers. The text "Gestures (Жесты)" is centered in the blue band in a white, bold, sans-serif font.

Gestures (Жесты)

Gestures

GestureArenaMember

Recognizers

GestureRecognizer

MultiDragGestureRecognizer

DoubleTapGestureRecognizer

MultiTapGestureRecognizer

OneSequenceGestureRecognizer

Widgets

Listener

RawGestureDetector

GestureDetector

Gestures

GestureArenaMember

Recognizers

GestureRecognizer

MultiDragGestureRecognizer

DoubleTapGestureRecognizer

MultiTapGestureRecognizer

OneSequenceGestureRecognizer

Widgets

Listener

RawGestureDetector

GestureDetector

GestureDetector

Виджет который позволит нам считывать жесты пользователя, в зоне дочернего виджета [child]

Нажатия

Tap

Double tap

Long press

Перетаскивания

Vertical drag

Horizontal drag

Pan

Прочие

Scale

GestureDetector. Lifecycle

Во время касания вызывается **[TapGestureRecognizer]**.

onTapDown > onTapUp > onTap

Во время длительного нажатия вызывает **[LongPressGestureRecognizer]**

onLongPressStart > onLongPress > onLongPressEnd > onLongPressUp

Во время семантического горизонтального перетаскивания он вызывает **[HorizontalDragGestureRecognizer]**

onDown > onStart > onUpdate > onEnd

Затем **[PanGestureRecognizer]**

onDown > onStart > onUpdate > onEnd

Во время вертикального перетаскивания он вызывает **[VerticalDragGestureRecognizer]**

onDown > onStart > onUpdate > onEnd

затем **[PanGestureRecognizer]**

onDown > onStart > onUpdate > onEnd

Нажатия

<code>onTapDown</code>	Запускается каждый раз, когда пользователь коснулся экрана.
<code>onTapUp</code>	Запускается каждый раз, когда пользователь перестал касаться экрана.
<code>onTap</code>	Запускается если <code>onTapDown</code> и <code>onTapUp</code> произошли в одном месте.
<code>onTapCancel</code>	Запускается если <code>onTapDown</code> и <code>onTapUp</code> произошли в разных местах.
<code>onDoubleTap</code>	Вызывается при быстром двойном касании экрана.
<code>onLongPress</code>	Запускается если пользователь касается экрана дольше 500 миллисекунд

Перетаскивания Vertical

`onVerticalDragDown`

Вызывается когда указатель соприкасается с экраном и начинает двигаться в вертикальном направлении

`onVerticalDragStart`

Вызывается, когда указатель начинает двигаться в вертикальном направлении.

`onVerticalDragUpdate`

Это вызывается каждый раз, когда изменяется положение указателя на экране.

`onVerticalDragEnd`

Когда пользователь прекращает движение, перетаскивание считается завершенным и вызывается это событие.

`onVerticalDragCancel`

Вызывается, когда пользователь внезапно прекращает перетаскивание.

Перетаскивания Horizontal

`onHorizontalDragDown`

Вызывается когда указатель соприкасается с экраном и начинает двигаться в горизонтальном направлении

`onHorizontalDragStart`

Вызывается, когда указатель начинает двигаться в горизонтальном направлении.

`onHorizontalDragUpdate`

Это вызывается каждый раз, когда изменяется положение указателя на экране.

`onHorizontalDragEnd`

Когда пользователь прекращает движение, перетаскивание считается завершенным и вызывается это событие.

`onHorizontalDragCancel`

Вызывается, когда пользователь внезапно прекращает перетаскивание.

Перетаскивания Pan

`onPanDown`

Вызывается, когда указатель касается экрана.

`onPanStart`

Запускается, когда событие указателя начинает перемещаться.

`onPanUpdate`

Вызывается каждый раз, когда указатель меняет местоположение.

`onPanEnd`

Вызывается когда взаимодействие закончено

onScale

`onScaleStart`

Указатели, соприкасающиеся с экраном, установили фокус и начальная шкала 1.0.

`onScaleUpdate`

Указатели, соприкасающиеся с экраном, указали на новую точку фокусировки и / или масштаб.

`onScaleEnd`

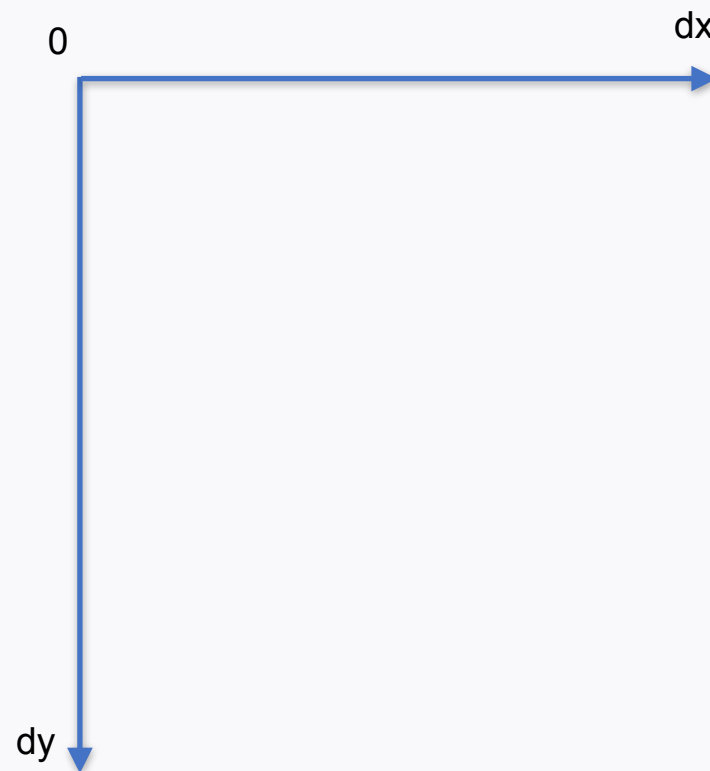
Указатели больше не соприкасаются с экраном

Offset

Смещение в логических пикселях относительно левого верхнего угла родительского поля. **Offset(dx,dy)**

localPosition - отсчет относительно родительского виджета

globalPosition - отсчет относительно экрана устройства



The image features a central horizontal band with a blue-to-green gradient. Overlaid on this band is a network of white lines connecting various points, resembling a data or neural network structure. The background of the entire image is an aerial view of a city skyline, with numerous skyscrapers and buildings, all rendered in a monochromatic blue and green color palette.

RawGestureDetector

RawGestureDetector

Позволяет добавлять кастомные Recognizers и обрабатывать их

```
Map<Type, GestureRecognizerFactory> gestures;
```

Ink

Удобный виджет для создания эффекта чернил и обработки касаний

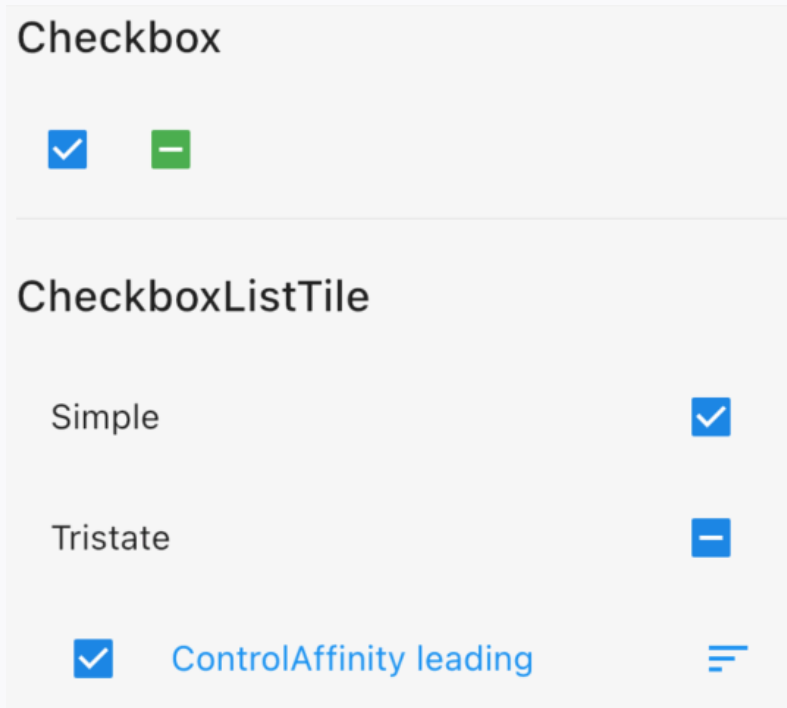
<https://api.flutter.dev/flutter/material/Ink-class.html>



The image features a central horizontal band with a blue-to-teal gradient. Overlaid on this band is a white network pattern of interconnected lines and nodes. The background of the entire image is an aerial view of a city skyline, with numerous skyscrapers and buildings, all rendered in a monochromatic blue color scheme. The text is centered within the blue band.

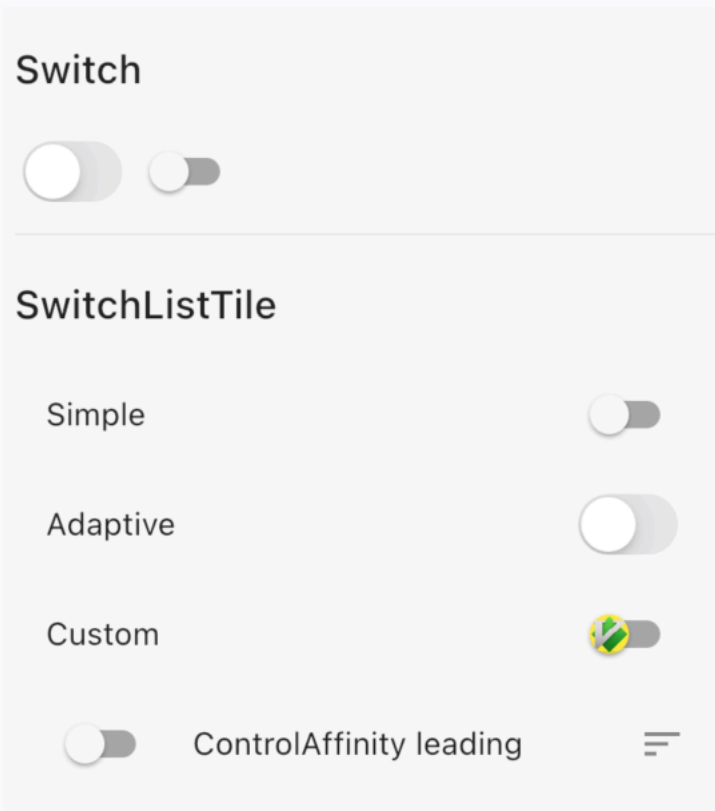
Виджеты ввода данных

Checkbox



```
Checkbox(  
  onChanged: (value){  
    // Тут необходимо поменять значение  
    // И обновить состояние виджета  
    // Например  
    // setState((value) => this.value = value)  
  },  
  // Позволяет иметь три состояния  
  // третье состояние - null  
  tristate: true,  
  // текущее значение  
  value: value,  
),
```

Switch



```
Switch(  
    onChanged: (value){  
        //аналогично checkbox  
    },  
    // текущее значение  
    value: value,  
),
```

```
//Создает CupertinoSwitch если используется iOS  
//Либо Material Switch если Android  
Switch.adaptive()
```

Radio

RadioListTile

- Item 0
- Item 1
- Item 2
- Item 3
- Item 4

```
Radio<T>(  
  onChanged: (T value){  
    //Здесь присваиваем значение для всей  
    группы  
  },  
  // Значение этого Radio  
  value: T,  
  // Текущее выбранное значение для группы  
  groupValue: T,  
  // Дает возможность отменить выбранное  
  значение  
  toggleable: true,  
);
```

`\${Name}ListTile

Checkbox, **Switch** и **Radio** имеют удобную обертку в виде **NameListTile**, где **Name** - вид виджета

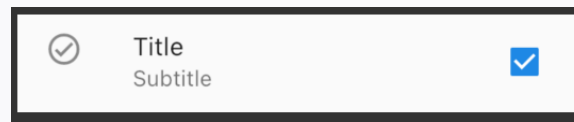
Пример **CheckboxListTile**

controlAffinity - отвечает за то, с какой стороны будет активный элемент

ListTileControlAffinity.**leading**



ListTileControlAffinity.**trailing**





TextField



TextFields

EditableText

TextField

CupertinoTextField

<https://material.io/components/text-fields>

<https://api.flutter.dev/flutter/cupertino/CupertinoTextField-class.html>

<https://developer.apple.com/documentation/uikit/uitextfield>

TextEditingController

Контроллер редактируемого текстового поля.

Каждый раз, когда пользователь изменяет текстовое поле со связанным [TextEditingController], текстовое поле обновляет [value] и контроллер уведомляет своих слушателей. Затем слушатели могут прочитать [text] и [selection] свойства, чтобы узнать, что набрал пользователь или как был сделан выбор.

<https://flutter.dev/docs/cookbook/forms/text-field-changes#2-use-a-texteditingcontroller>

FocusNode

Объект, который может использоваться виджетом с отслеживанием состояния для получения фокуса клавиатуры и обработки событий клавиатуры.

InputFormatters

TextInputFormatter

BlacklistingTextInputFormatter

WhitelistingTextInputFormatter

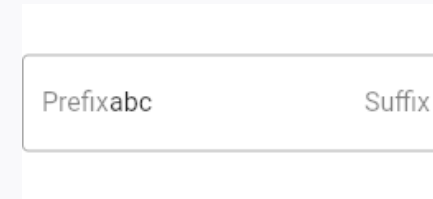
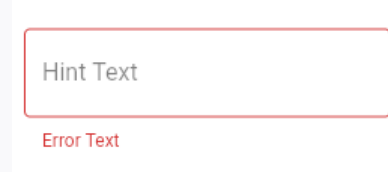
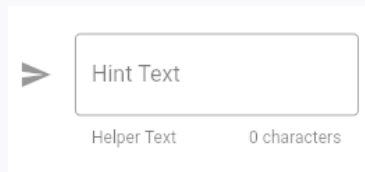
FilteringTextInputFormatter

LengthLimitingTextInputFormatter

InputDecoration

Позволяет настроить для TextField такие параметры как border, labels, icons, and styles

<https://api.flutter.dev/flutter/material/InputDecoration-class.html>



Form

Предоставляет нам возможность проверить значение внутри виджета.

Form

Виджет может получать состояния всех FormFields внутри себя

FormField

Виджет - который содержит в себе состояние об ошибках и валидации дочернего виджета

TextFormField

FormField которое содержит в себе TextField

FormField

initialValue: начальное значение для отображения в поле.

builder: обратный вызов, который отвечает за создание виджета внутри поля формы.

onSaved: обратный вызов, который вызывается всякий раз, когда мы отправляем поле (обычно путем вызова метода сохранения в форме).

validator: обратный вызов, который вызывается, чтобы узнать, имеет ли поле в настоящее время допустимое значение.

autovalidate: логическое значение, указывающее, хотим ли мы вызывать валидатор каждый раз при изменении поля или только при отправке поля.

Цели вебинара

1

Научиться обрабатывать жесты.

2

Научиться работать с Checkbox, Switch, Radio виджетами

3

Научиться работать с виджетами ввода текста.



Заполните, пожалуйста,
опрос о занятии по ссылке в чате

