



Текстовые поля

Последнее обновление: 12.06.2016



Ночью ударят морозы?

yandex.by/Погода

Смотрите прогноз погоды в Яндексе на сегодня и 10 дней вперёд.



В Xamarin Forms текстовые поля представлены несколькими классами:

- **Label**: текстовая метка для вывода текста
- **Entry**: однострочное текстовое поле
- **Editor**: многострочное текстовое поле

Label

Label представляет обычную текстовую метку, которая выводит информацию с помощью свойства Text. Label удобен для создания заголовков и меток к элементам ввода.

Label довольно простой элемент, однако даже с ним могут возникать некоторые сложности. Например, кнопки определяют событие нажатия, которое мы можем обработать. У Label подобного события нет. Но что делать, если мы вдруг хотим обрабатывать нажатие на заголовок? В этом случае мы можем воспользоваться специальным классом

TapGestureRecognizer, который позволяет обрабатывать нажатия:

```
public class MainPage : ContentPage
{
    public MainPage()
    {
        Label label = new Label
        {
            VerticalTextAlignment = TextAlignment.Center,
            HorizontalTextAlignment = TextAlignment.Center,
            Text = "Welcome to Xamarin Forms!",
            FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label))
        };

        TapGestureRecognizer tapGesture = new TapGestureRecognizer
        {
            NumberOfTapsRequired = 2
        };

        int count = 0; // счетчик нажатий
        tapGesture.Tapped += (s, e) =>
        {
            count++;
            if (count % 2 == 0)
            {
                label.BackgroundColor = Color.Black;
                label.TextColor = Color.White;
            }
            else
            {
                label.TextColor = Color.Black;
                label.BackgroundColor = Color.White;
            }
            label.Text = string.Format("Вы нажали {0} раз", count);
        };
        label.GestureRecognizers.Add(tapGesture);

        Content = label;
    }
}
```

}

TapGestureRecognizer представляет специальный класс, который позволяет распознать нажатия. С помощью свойства NumberOfTapsRequired мы можем установить, сколько нажатий необходимо. В данном случае мы устанавливаем двойное нажатие. И именно по двойному нажатию будет генерироваться событие Tapped, в обработчике которого мы будем переключать цвет текста и фона метки.

И чтобы связать объект TapGestureRecognizer с меткой, необходимо добавить этот объект в коллекцию GestureRecognizers:

```
label.GestureRecognizers.Add(tapGesture);
```



Текстовое поле Entry

Entry представляет текстовое поле для ввода однострочной информации:

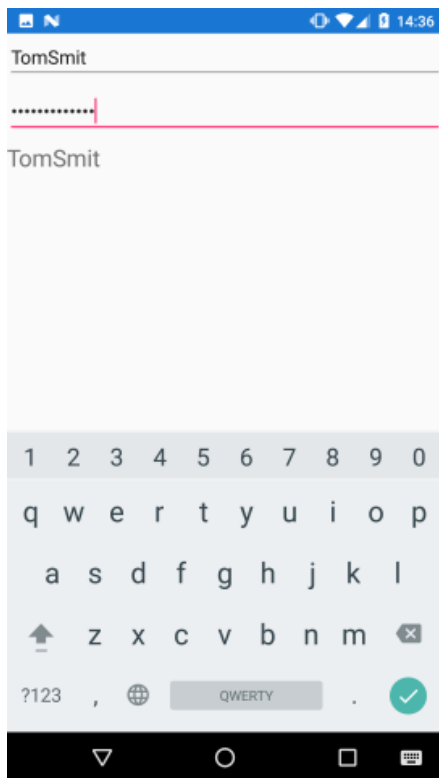
```
class MainPage : ContentPage
{
    Label textLabel;
    Entry loginEntry, passwordEntry;
    public MainPage()
    {
        StackLayout stackLayout = new StackLayout();

        loginEntry = new Entry { Placeholder = "Login" };
        loginEntry.TextChanged+=loginEntry_TextChanged;

        passwordEntry = new Entry
        {
            Placeholder = "Password",
            IsPassword = true
        };
        textLabel = new Label { FontSize = Device.GetNamedSize(NamedSize.Large, typeof(Label)) };

        stackLayout.Children.Add(loginEntry);
        stackLayout.Children.Add(passwordEntry);
        stackLayout.Children.Add(textLabel);
        this.Content = stackLayout;
    }

    private void loginEntry_TextChanged(object sender, TextChangedEventArgs e)
    {
        textLabel.Text = loginEntry.Text;
    }
}
```



Обработать ввод пользователя мы можем с помощью обработки события `TextChanged`

Аналогичный пример в xaml:

```
<StackLayout>
  <Entry x:Name="loginEntry" TextChanged="loginEntry_TextChanged" />
  <Entry x:Name="passwordEntry" Placeholder = "Password" IsPassword = "True" />
  <Label x:Name="textLabel" FontSize="Large" />
</StackLayout>
```

Основные свойства элемента `Entry`, которые мы можем использовать:

- **Text:** введенный в поле текст
- **TextColor:** цвет вводимого текста
- **Placeholder:** текст-заменитель, который отображается в поле до ввода и исчезает при начале печати
- **IsPassword:** при значении `true` указывает, то поле будет предназначено для ввода пароля, и поэтому все вводимые символы будут заменяться звездочкой

Также класс `Entry` определяет два события:

- **TextChanged:** возникает при вводе символов в поле
- **Completed:** возникает при завершении ввода

Формат клавиатуры

Среди свойств текстового поля ввода наиболее интересным является свойство **Keyboard**. Оно позволяет задать формат клавиатуры для ввода символов. Например, если необходимо ввести в поле какую-то числовую информацию, то гораздо было бы проще с точки зрения юзабилити предоставить пользователю сразу числовую раскладку клавиатуры. Данное свойство принимает одно из значений перечисления `Keyboard`:

- Default
- Text
- Chat
- Url
- Email
- Telephone

- Numeric

Каждое значение предоставляет свою раскладку клавиатуры, предназначенную для ввода определенной информации. Например, `entry.Keyboard=Keyboard.Telephone` - при вводе нам автоматически представляется раскладка, содержащая только те знаки, которые используются при наборе телефонного номера.

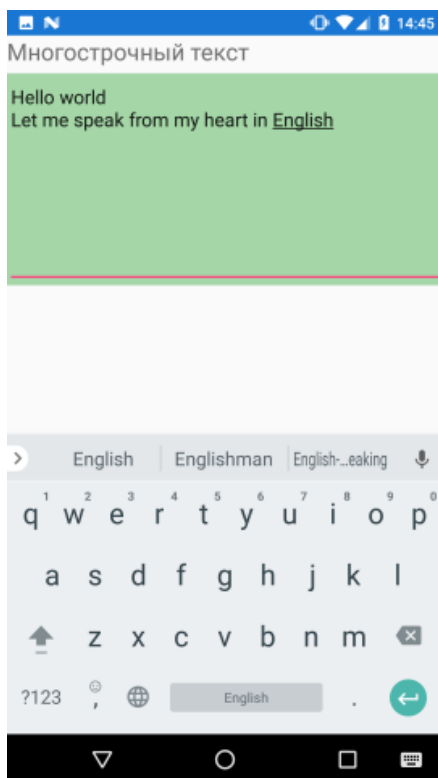
Editor

В отличие от Entry **Editor** представляет собой многострочное поле ввода, но принципы его работы аналогичны. Он также имеет все те же свойства и события. Например:

```
public MainPage()
{
    StackLayout stackLayout = new StackLayout();

    Editor textEditor = new Editor { HeightRequest=200, BackgroundColor=Color.Teal};
    Label textLabel = new Label
    {
        Text="Многострочный текст",
        FontSize= Device.GetNamedSize(NamedSize.Large, typeof(Label))
    };

    stackLayout.Children.Add(textLabel);
    stackLayout.Children.Add(textEditor);
    this.Content = stackLayout;
}
```



Реализация в xaml:

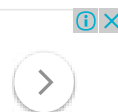
```
<StackLayout>
  <Label Font="26" Text="Многострочный ввод" />
  <Editor BackgroundColor="#a5d6a7" HeightRequest="200" />
</StackLayout>
```



Antminer S9 14TH/s

Bitmain-Partner

World's Most Powerful Bitcoin Miner, with Inexpensive Prices And Fast Shipping

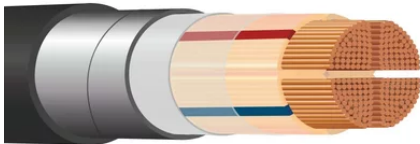


[Назад](#) [Содержание](#) [Вперед](#)



Кабель АВБШв. Склад. Доставка! ▾

inkabel.ru



Яндекс.Директ



9 Комментариев metanit.com

Войти ▾

Рекомендовать Поделиться

Лучшее в начале ▾



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ

ИЛИ ЧЕРЕЗ DISQUS

Имя



Катя • месяц назад

Как сделать особое поле для ввода телефона, со скобками, - и т д? что-то похожее на: (____)____-__

^ | ▾ • Ответить • Поделиться ▸



Hanna • 7 месяцев назад

Возник вопрос - существует ли для Entry подобное свойству MaxLength для TextBox в WPF?

^ | ▾ • Ответить • Поделиться ▸



Balalamb • 9 месяцев назад

Такой вопрос, у меня в элементе Entry при вводе курсор и полоса под текстом имеют розовый цвет, как это можно изменить?

^ | ▾ • Ответить • Поделиться ▸



Артем → **Balalamb** • 8 месяцев назад

нужно изменить значения цветов в проекте андроида Resources->values->styles.xml на те, которые нужны или переопределить рендер

^ | ▾ • Ответить • Поделиться ▸



Balalamb → **Артем** • 8 месяцев назад

спасибо

^ | ▾ • Ответить • Поделиться ▸



Илья → **Balalamb** • 9 месяцев назад

Возник такой же вопрос, Вы не нашли ответа?

^ | ▾ • Ответить • Поделиться ▸



Balalamb → **Илья** • 7 месяцев назад

вот, мне ответили - нужно изменить значения цветов в проекте андроида Resources->values->styles.xml на те, которые нужны или переопределить рендер

^ | ▾ • Ответить • Поделиться ▸

**Makarkin & Partners / Макаркин** • год назад

что-то на ANDroid эмуляторе не вылазит клавиатура

^ | ▾ • Ответить • Поделиться ›

**Майк Элеван** → Makarkin & Partners / Макаркин • 10 месяцев назад

А вы на чем набираете код ? Вот на нем и печатается текст в соотв поле.

^ | ▾ • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

Angular в ASP.NET Core | Первый проект

33 комментариев • 2 месяца назад

**Zaur YakubOff** — Проблема вот в чем оказалось. Прошу прощения за занудство
`app.UseWebpackDevMiddleware(new ...`**Vue.js | Компоненты в отдельных файлах и Webpack**

2 комментариев • 3 месяца назад

**Metanit** — `package.json`"scripts": { "dev": "webpack-dev-server --hot --open", "build": "webpack" },**C# и .NET | Раннее и позднее связывание**

3 комментариев • 2 месяца назад

**Metanit** — не только, это, к примеру, могут быть также абстрактные методы, которые переопределяются в классах-наследниках**Angular в ASP.NET Core | CRUD и маршрутизация. Часть 2**

5 комментариев • месяц назад

**Igor Teterkin** — Поправка - это в IE11. В google chrome всё

Подписаться Добавить Disqus на свой сайт Добавить Disqus Добавить Конфиденциальность

[Вконтакте](#) | [Twitter](#) | [Google+](#) | [Канал сайта на youtube](#) | [Помощь сайту](#)

Контакты для связи: metanit22@mail.ru

Copyright © metanit.com, 2012-2017. Все права защищены.