



Позиционирование элементов на странице

Последнее обновление: 26.03.2017



Помогите им найти вашу
компанию в Google!

НАЧНИТЕ СЕЙЧАС

Компенсируем
до \$60

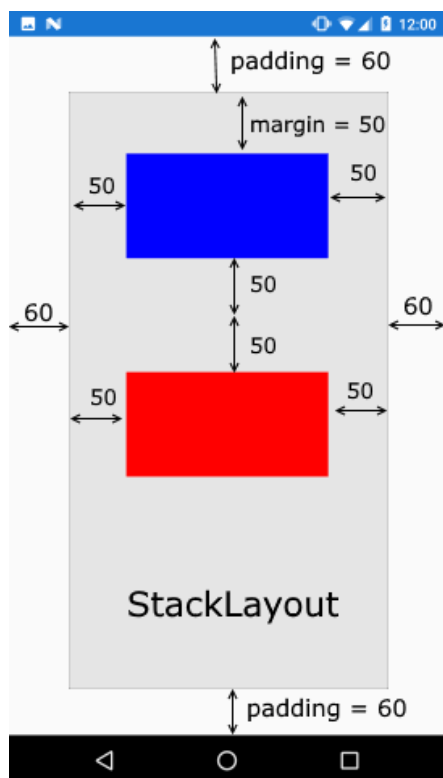
Google AdWords

Отступы

Для установки отступов у элементов применяются свойства **Margin** и **Padding**. Свойство **Margin** определяет внешний отступ элемента от других элементов или контейнера. А свойство **Padding** устанавливает внутренние отступы - от внутреннего содержимого элемента до его границ. Оба этих свойства представляют структуру **Thickness**.

Например, зададим оба отступа в XAML:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  x:Class="HelloApp.MainPage">
  <StackLayout Padding="60">
    <BoxView Color="Blue" Margin="50" HeightRequest="100" />
    <BoxView Color="Red" Margin="50" HeightRequest="100" />
  </StackLayout>
</ContentPage>
```



Установка в коде C#:

```
public class MainPage : ContentPage
{
    public MainPage()
    {
```

```

var stackLayout = new StackLayout
{
    Padding = new Thickness(60),
    Children = {
        new BoxView { Color = Color.Blue, Margin = new Thickness (50) },
        new BoxView { Color = Color.Red, Margin = new Thickness (50) }
    }
};
Content = stackLayout;
}
}

```

Кроме того, структура Thickness имеет свойства Left, Top, Right, Bottom, поэтому задать ее определение мы можем тремя способами:

- Указать одно значение для отступов со всех сторон
- Указать два значения для отступов по горизонтали (слева и справа) и вертикали (сверху и снизу)
- Указать четыре значения для отступов для каждой из сторон

Установка в коде:

```

public class MainPage : ContentPage
{
    public MainPage()
    {
        var stackLayout = new StackLayout
        {
            Padding = new Thickness(0, 20, 0, 0),
            Children = {
                new BoxView { Color = Color.Green, Margin = new Thickness (20) },
                new BoxView { Color = Color.Blue, Margin = new Thickness (10, 25) },
                new BoxView { Color = Color.Red, Margin = new Thickness (0, 20, 15, 5) }
            }
        };
        Content = stackLayout;
    }
}

```

Определение в XAML:

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    x:Class="HelloApp.MainPage">
    <StackLayout Padding="0,20,0,0">
        <BoxView Color="Green" Margin="20" />
        <BoxView Color="Blue" Margin="10, 15" />
        <BoxView Color="Red" Margin="0, 20, 15, 5" />
    </StackLayout>
</ContentPage>

```

В отношении отступ надо учитывать следующую вещь: так как на устройствах с iOS верхняя панель занимает 20 единиц, то для iOS желательно устанавливать отступ сверху размером как минимум в 20 единиц, чтобы текст метки не налезил на панель.

В этом случае мы можем глобально установить отступ сверху для всех платформ в 20 или более единиц. Либо можно задать отступ непосредственно только для iOS:

```

public class MainPage : ContentPage
{
    public MainPage()
    {
        Label header = new Label() { Text = "Привет из Xamarin Forms" };
        this.Content = header;
        Device.OnPlatform(iOS: () => { Padding = new Thickness(0, 20, 0, 0); });
    }
}

```

В данном случае применяется метод **Device.OnPlatform()**, Для задания кода только для iOS используются директивы препроцессора, которые позволяют задать код только для одной определенной платформы.

Выравнивание по горизонтали и вертикали

Все элементы, используемые при создании интерфейса, наследуются от класса View, который определяет два свойства **HorizontalOptions** и **VerticalOptions**. Они управляют выравнивание элемента соответственно по горизонтали и по вертикали.

В качестве значения они принимают структуру **LayoutOptions**. Данная структура имеет ряд свойств, которые хранят объекты опять же LayoutOptions:

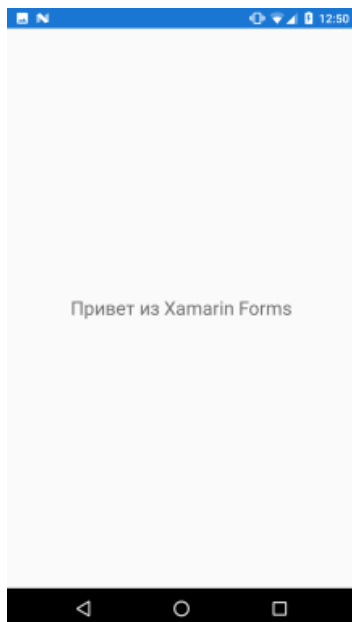
- **Start**: выравнивание по левому краю (выравнивание по горизонтали) или по верху (выравнивание по вертикали)
- **Center**: элемент выравнивается по центру
- **End**: выравнивание по правому краю (выравнивание по горизонтали) или по низу (выравнивание по вертикали)
- **Fill**: элемент заполняет все пространство контейнера
- **StartAndExpand**: аналогичен опции Start с применением растяжения
- **CenterAndExpand**: аналогичен опции Center с применением растяжения
- **EndAndExpand**: аналогичен опции End с применением растяжения
- **FillAndExpand**: аналогичен опции Fill с применением растяжения

Зададим выравнивание в коде C#:

```
Label header = new Label() { Text = "Привет из Xamarin Forms" };  
header.VerticalOptions = LayoutOptions.Center;  
header.HorizontalOptions = LayoutOptions.Center;
```

Аналог в xaml:

```
<Label Text="Привет из Xamarin Forms" VerticalOptions="Center" HorizontalOptions="Center" />
```



Выравнивание текста внутри элемента

Выравнивание текста по горизонтали и вертикали задается с помощью свойств **HorizontalTextAlignment** и **VerticalTextAlignment** соответственно. В качестве значения эти свойства принимают одну из констант перечисления **TextAlignment**:

- Start: текст выравнивается по левому краю по горизонтали или по верху по вертикали
- Center: текст выравнивается по центру
- End: текст выравнивается по правому краю по горизонтали или по низу по вертикали

Пример на c#:

```
Label header = new Label() { Text = "Привет из Xamarin Forms" };  
header.HorizontalTextAlignment = TextAlignment.Center;  
header.VerticalTextAlignment = TextAlignment.Center;
```

В xaml:

```
<Label Text="Привет из Xamarin Forms" VerticalTextAlignment="Center" HorizontalTextAlignment="Center" />
```

	41,01 руб.	51,30 руб.	79 руб.	84 руб.	59,98 руб.	88 руб.	24 80 руб. ООО "Триовист"


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



Тачка DGM GT-1081 ^

21vek.by

Количество колес: 1. Материал: оцинкованная сталь. Купить за 41 Br.



**ОБРАЗОВАТЕЛЬНЫЙ
ЦЕНТР ПВТ**
**КУРС iOS
РАЗРАБОТЧИКА**
Высокая востребованность
на рынке IT!

Яндекс.Директ

7 Комментариев


metanit.com

1 Войти

Рекомендовать

Поделиться


Лучшее в начале



Присоединиться к обсуждению...

ВОЙТИ С ПОМОЩЬЮ


ИЛИ ЧЕРЕЗ DISQUS ?



Георгий • год назад

Разметка для XAML, для определения платформы, в студию:)


1 ^ | ▾ • Ответить • Поделиться ▸



C13r1k → Георгий • год назад

```
<contentpage.padding>
<onplatform x:typearguments="Thickness">
<onplatform.ios>20</onplatform.ios>
<onplatform.android>10</onplatform.android>
</onplatform>
</contentpage.padding>
```

^ | ▾ • Ответить • Поделиться ▸




Erazer • 5 месяцев назад

Подскажите где туплю, пытаюсь расположить кнопку используя свойства Padding и Margin но кнопка до середины доходит а потом начинается во что то упираться...

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/form
xmlns:x="http://schemas.microsoft.com/winfx
x:Class="HelloApp.MainPage">
  <RelativeLayout>

    <Image Source="StartFon.jpg" />
    <StackLayout Padding="1">
      <Button Text="Starwt"
        BackgroundColor="White"
```



```

<StackLayout
    BorderColor="Black"
    Margin="90"
    Clicked="OnButtonClicked"/>
</StackLayout>
</RelativeLayout>
</ContentPage>

```



показать больше

^ | v • Ответить • Поделиться ›



Александр Братцев → Egrazer • 2 месяца назад

Либо в демона Максвелла упирается, либо может Margin="90" со всех сторон отступ дает. Но могу быть не правым(

^ | v • Ответить • Поделиться ›



Майк Элеван • 10 месяцев назад

Разберите еще SearchBar

^ | v • Ответить • Поделиться ›



Игорь • год назад

"отступ в 50 пикселей". Везде вы укзывали о Еденицах, но не пикселях. В XAML-разметке тут используются поинты, а не пиксели. Формулу можно прогуглить.

^ | v • Ответить • Поделиться ›



Metanit Модератор → Игорь • год назад

исправил на единицы

^ | v • Ответить • Поделиться ›

ТАКЖЕ НА METANIT.COM

Go | Соответствие интерфейсу

1 комментарий • 11 дней назад



Пу — Еще давай. Не останавливайся, пожалуйста. Это гениальный язык. Он, конечно, не божественен, как шарп, но гениальный.

C# и .NET | Переменные

2 комментария • 2 месяца назад



Максим Кусков — Ну это было просто) Двигаемся дальше!)

Kotlin | Руководство

2 комментария • месяц назад



Артур Голояд — Просто шикарно, может подкинете материалов по работе с сетями на Kotlin?

Angular в ASP.NET Core | Введение

14 комментариев • 2 месяца назад



Igor Teterkin — Если у меня есть в контроллере UserController) метод Send (отправка почты сотруднику) [HttpPost("{id}")] public IActionResult Send(int id), то как ...

✉ Подписаться • ➕ Добавить Disqus на свой сайтДобавить DisqusДобавить • 🔒 Конфиденциальность



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

Контакты для связи: metanit22@mail.ru

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