

# Модификаторы и визуальный интерфейс

## Что такое модификаторы

Для настройки внешнего вида большинства встроенных компонентов в Jetpack Compose применяются так называемые модификаторы. Модификаторы представляют функции, которые задают какой-то отдельный аспект для компонентов (или иными словами "модифицируют" внешний вид компонента), например, установка размеров компонента или его фонового цвета и т.д.

Большинство встроенных компонентов поддерживают применение модификаторов через параметр `modifier`. Например, возьмем ранее встречавшийся простой компонент `Text`, который выводит некоторый текст. Данный компонент, как и другие компоненты в Jetpack Compose, определен в виде функции с аннотацией `@Composable`, которая имеет ряд параметров:

```
@Composable
fun Text(
    value: String,
    modifier: Modifier = Modifier,

    //..... остальные параметры

): @Composable Unit
```

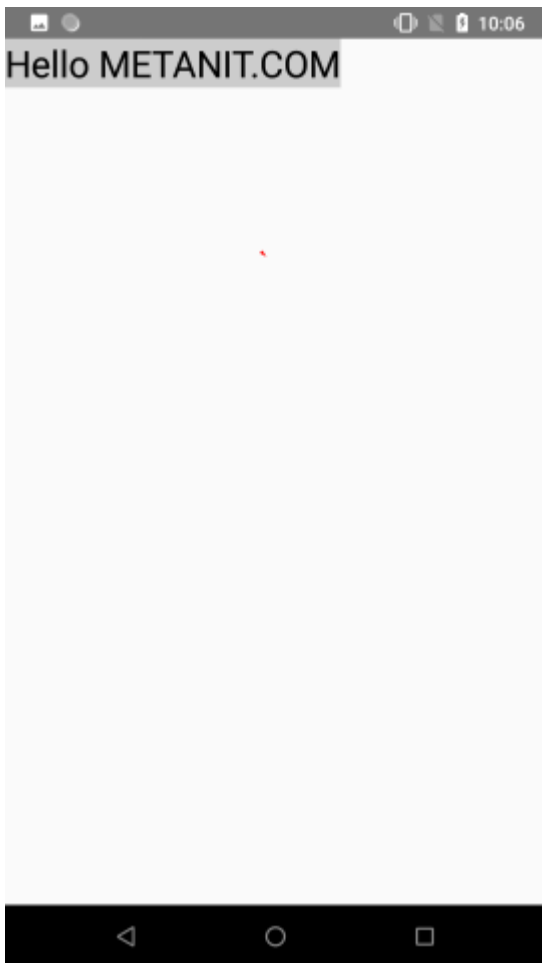
Из определения функции компонента мы видим, что она имеет параметр `modifier`, который представляет тип (а если точнее интерфейс) `Modifier`. И этот тип предоставляет огромное количество встроенных функций-модификаторов, изменяющих отдельные аспекты компонентов. Далее отдельно рассмотрим отдельные группы этих функций. А пока посмотрим на примере компонента `Text`, как применять модификаторы.

```
package com.example.helloapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.material.Text
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Text(
                "Hello METANIT.COM",
                fontSize = 22.sp,
```

```
        modifier = Modifier.background(Color.Yellow)
    )
}
}
```



Здесь для компонента Text задается три параметра. Первый параметр представляет отображаемый текст. Второй параметр - `fontSize` задает размер шрифта - 22 единицы. Третий параметр задает модификаторы, которые применяются к компоненту:

```
modifier = Modifier.background(Color.Yellow)
```

В данном случае применяется только один модификатор - функция `background()`, которая устанавливает фоновый цвет. В нее передается определение цвета - здесь желтый цвет, предоставленный встроенным значением `Color.Yellow`.

Стоит учитывать, что функции модификаторы, как и используемые ими значения, необходимо импортировать:

```
import androidx.compose.foundation.background
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
```

# Применение нескольких модификаторов

К компоненту можно применять множество модификаторов по цепочке:

```
Modifier.модификатор1().модификатор2().модификатор3()...модификаторN()
```

Например, применим к компоненту Text несколько модификаторов:

```
package com.example.helloapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.Text
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Text(
                "Hello METANIT.COM",
                fontSize=22.sp,
                modifier = Modifier.background(Color.Yellow)
                    .fillMaxSize()
                    .padding(30.dp)
            )
        }
    }
}
```

Здесь к компоненту Text применяются три функции-модификатора. Первая задает фоновый цвет:

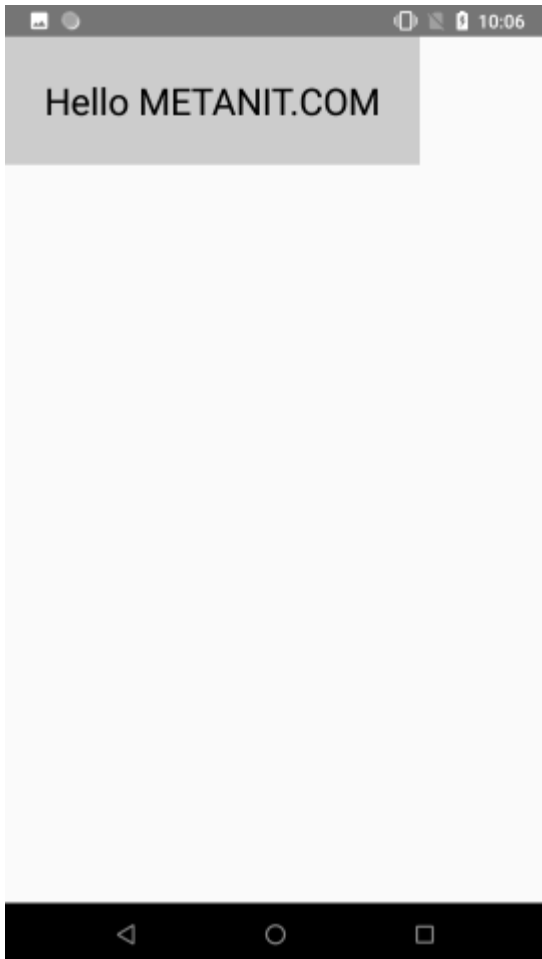
```
Modifier.background(Color.Yellow)
```

Второй модификатор устанавливает размер, а точнее растягивает компонент по всей длине контейнера:

```
fillMaxSize()
```

Третий модификатор устанавливает отступы (30 единиц) от границ контейнера:

```
padding(30.dp)
```



Здесь для компонента Text задается три параметра. Первый параметр представляет отображаемый текст. Второй параметр - `fontSize` задает размер шрифта - 22 единицы. Третий параметр задает модификаторы, которые применяются к компоненту:

```
modifier = Modifier.background(Color.Yellow)
```

В данном случае применяется только один модификатор - функция `background()`, которая устанавливает фоновый цвет. В нее передается определение цвета - здесь желтый цвет, предоставленный встроенным значением `Color.Yellow`.

Стоит учитывать, что функции модификаторы, как и используемые ими значения, необходимо импортировать:

```
import androidx.compose.foundation.background
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
```

## Применение нескольких модификаторов

К компоненту можно применять множество модификаторов по цепочке:

```
Modifier.модификатор1().модификатор2().модификатор3()...модификаторN()
```

Например, применим к компоненту Text несколько модификаторов:

```
package com.example.helloapp

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.Text
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Text(
                "Hello METANIT.COM",
                fontSize=22.sp,
                modifier = Modifier.background(Color.Yellow)
                    .fillMaxSize()
                    .padding(30.dp)
            )
        }
    }
}
```

Здесь к компоненту Text применяются три функции-модификатора. Первая задает фоновый цвет:

```
Modifier.background(Color.Yellow)
```

Второй модификатор устанавливает размер, а точнее растягивает компонент по всей длине контейнера:

```
fillMaxSize()
```

Третий модификатор устанавливает отступы (30 единиц) от границ контейнера:

```
padding(30.dp)
```

