



# Pie Chart

## *Compose*

Today I call everybody  
to learn **how** to **draw** a  
**pie chart** using Canvas  
with Jetpack Compose.



Scooby, today we  
will eat a lot  
of pies as many  
as we can!



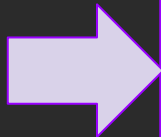
No!



*First we need to know  
why is a Pie Chart!*

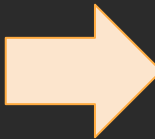
*Pie Chart is a circular  
diagram used to analyze the  
proportions of different  
categories of items in the  
same data class.*

Let's create a data class to represent each data set entry



```
data class PieDataItem(  
    val color: Color,  
    val label: String,  
    val value: Float  
)
```

I did a small **interview** with people of my class asking about their **favorite color**. I got the following **results**:



```
listOf(  
    PieDataItem(value = 0.15f, /*15%*/ color = Color.Red, label = "Red"),  
    PieDataItem(value = 0.05f, /*5%*/ color = Color.Blue, label = "Blue"),  
    PieDataItem(value = 0.3f, /*30%*/ color = Color.Green, label = "Green"),  
    PieDataItem(value = 0.25f, /*25%*/ color = Color.Magenta, label = "Magenta"),  
    PieDataItem(value = 0.25f, /*25%*/ color = Color.Gray, label = "Gray")  
)
```

Time to draw the data  
using Canvas:

```
@Composable
private fun PieChart(items: List<PieDataItem>) {
    // 1. sort the set according to percentage values
    val sortedItems = items.sortedBy { it.value }

    // 2. start drawing in the beginning
    var startAngle = 0f

    Box(
        modifier = Modifier.size(200.dp)
    ) { this: BoxScope
        sortedItems.forEach { ... }
    }
}
```

@Composable

private fun PieChart(items: List<PieDataItem>) {

// 1. sort the set according to percentage value  
val sortedItems = items.sortedBy { it.value }

// 2. start drawing in the beginning

var startAngle = 0f

Box(

modifier = Modifier.size(200.dp)

) { this: BoxScope

sortedItems.forEach { ... }

}

}

sortedItems.forEach { dataItem →

// 3. According to the value we need to know how degrees the arc should sweep

val sweepValue = dataItem.value \* 360f

Spacer(

modifier = Modifier

.size(200.dp)

// 4. drawBehind allow us to draw using DrawScope in any composable function

.drawBehind { this: DrawScope

// 5. Draw the arc

drawArc(

color = dataItem.color,

startAngle = startAngle,

sweepAngle = sweepValue,

useCenter = true,

)

// 6. Increment the startAngle variable to draw the next arc

startAngle += sweepValue

}

.align(Alignment.Center)

)

}





*What do you think  
friends?*





*Real pies are  
better!*

