

Unset

```
@Preview(showBackground = true, name = "Text preview")
@Composable
fun GreetingPreview() {
    BasicsCodeLabTheme {
        Greeting(name = "Android")
    }
}
```

GreetingPreview - Text preview

Hello Android!

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(color = MaterialTheme.colorScheme.primary) {
        Text(
            text = "Hello $name!",
            modifier = modifier
        )
    }
}
```

Hello Android!

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(color = MaterialTheme.colorScheme.primary) {
        Text(
            text = "Hello $name!",
            modifier = modifier.padding(24.dp)
        )
    }
}
```

GreetingPreview - Text preview



Hello Android!

```
// Don't copy over
Column {
    Text("First row")
    Text("Second row")
}
```

```

@Composable
fun MyApp(
    modifier: Modifier = Modifier,
    names: List<String> = listOf("World", "Compose")
) {
    Column(modifier) {
        for (name in names) {
            Greeting(name = name)
        }
    }
}

```

```

// Don't copy yet
Button(
    onClick = { } // ...
) {
    Text("Show less")
}

```

the `weight` modifier makes the element fill all available space, making it *flexible*, effectively pushing away the other elements that don't have a weight, which are called *inflexible*

```

fun Greeting(name: String, modifier: Modifier = Modifier) {
    Surface(
        color = MaterialTheme.colorScheme.primary,
        modifier = modifier.padding(vertical = 4.dp, horizontal = 8.dp)
    ) {
        Row(modifier = Modifier.padding(24.dp)) {
            Column(modifier = Modifier.weight(1f)) {
                Text(text = "Hello ")
                Text(text = name)
            }
            ElevatedButton(
                onClick = { /* TODO */ }
            ) {
                Text("Show more")
            }
        }
    }
}

```

```

ElevatedButton(
    onClick = { expanded.value = !expanded.value },
) {
    Text(if (expanded.value) "Show less" else "Show more")
}

```

```

@Composable
fun Greeting(name: String) {
    var expanded = false // Don't do this!

    Surface(
        color = MaterialTheme.colorScheme.primary,
        modifier = Modifier.padding(vertical = 4.dp, horizontal = 8.dp)
    ) {
        Row(modifier = Modifier.padding(24.dp)) {
            Column(modifier = Modifier.weight(1f)) {
                Text(text = "Hello, ")
                Text(text = name)
            }
            ElevatedButton(
                onClick = { expanded = !expanded }
            ) {
                Text(if (expanded) "Show less" else "Show more")
            }
        }
    }
}

```

```

import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
// ...

```

```

@Composable
fun Greeting(...) {
    val expanded = remember { mutableStateOf(false) }
    // ...
}

```