

CUSTOMIZAR UN COMPONENTE

CUSTOMIZAR UN COMPONENTE

- El objetivo de esta lección es hacer que el componente **GreetingText** tenga una apariencia customizada.
- Para ello vamos a explorar el atributo **style** a través de la función **TextStyle**.

CUSTOMIZAR UN COMPONENTE

- TextStyle
 - Color del texto
 - Tamaño
 - Tipografía
 - Espacio entre letras
 - Indentación
 - ...

```
androidx.compose.ui.text.TextStyle
public constructor TextStyle(
    color: Color,
    fontSize: TextUnit,
    fontWeight: FontWeight?,
    fontStyle: FontStyle?,
    fontSynthesis: FontSynthesis?,
    fontFamily: FontFamily?,
    fontFeatureSettings: String?,
    letterSpacing: TextUnit,
    baselineShift: BaselineShift?,
    textGeometricTransform: TextGeometricTransform?,
    localeList: LocaleList?,
    background: Color,
    textDecoration: TextDecoration?,
    shadow: Shadow?,
    textAlign: TextAlign?,
    textDirection: TextDirection?,
    lineHeight: TextUnit,
    textIndent: TextIndent?
)
```

CUSTOMIZAR UN COMPONENTE

- TextStyle:
 - color: Color
 - fontWeight: FontWeight
 - fontSize: sp

```
@Composable
fun GreetingText(name: String) {
    Text(text = "Hello $name!",
        style = TextStyle(
            color = Color.Red,
            fontWeight = FontWeight.Bold,
            fontSize = 18.sp
        )
    )
}
```

DefaultPreview

Hello OpenWebinars!

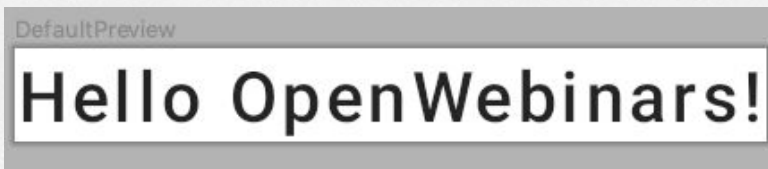
CUSTOMIZAR UN COMPONENTE

- Material Themes:
 - Google recomienda el uso de su sistema de diseño [Material Design](#).
 - Jetpack Compose provee estilos de **Material Design** ya predefinidos para poder usar en nuestros componentes:
 - h1, h2, button, caption, body...

CUSTOMIZAR UN COMPONENTE

- Material Button:
 - MaterialTheme
 - Aplica:
 - fontFamily: Roboto
 - fontSize: 14sp
 - letterSpacing: 1.25sp

```
@Composable
fun GreetingText(name: String) {
    Text(text = "Hello $name!",
        style = MaterialTheme.typography.button
    )
}
```



CUSTOMIZAR UN COMPONENTE

- Sobrescritura de valores:
 - Uso `MaterialTheme.typography.h5`
 - Sobrescritura del parámetro `fontWeight` por `SemiBold`.

```
@Composable
fun GreetingText(name: String) {
    Text(text = "Hello $name!",
        style = MaterialTheme.typography.h5,
        fontWeight = FontWeight.SemiBold
    )
}
```

DefaultPreview

Hello OpenWebinars!

CUSTOMIZAR UN COMPONENTE

- Clases **MaterialTheme** y **FontWeight**:
 - Son un **companion object** de Kotlin.
 - Todos sus campos son **estáticos** y, por tanto, accesibles desde nuestros componentes.