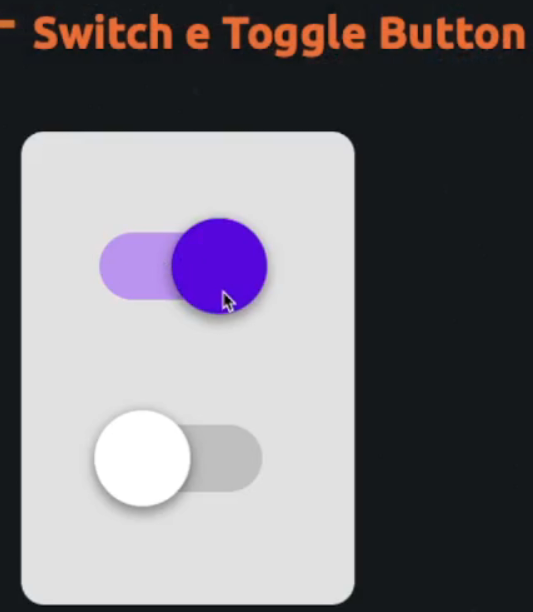
Switch



Toggle Button



Voce pode ativar e desativar

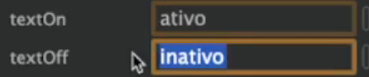
Atributos Switch

**Thumb** = podemos no local da bolina adicionar um ícone

**ThumbTint** = muda a cor da bolina

**Track** = onde ta cinza podemos colocar uma outra cor, ou ate mesmo um outro ícone 

**TrackTint** = ali podemos alterar a cor 



Esses são textos que podemos capturar conforme o usuário ativa e desativa o switch, imagina que queremos um tema preto, então podemos fazer um if, para quando o usuário ativar esse switch.

**Checked** = você pode ativar e ele vai ficar marcado como padrão

**showText** = podemos exibir o texto do textOn ou textoOff 

# ToggleButton



Também temos esse botão, ali já é um pouco diferente, ali conseguimos exibir o off ou on.

*/\*\* FORMA 1 DE FAZER  
 \* Aqui vamos pegar o objeto Switch e vamos fazer um evento de clique, que é quando ele for ativo  
 \* Ele verifica se o item foi ativo  
 \* Se foi ativo ele vai mudar o tema.  
 \* Aqui ele vai colocar no tema escuro ou nao  
 \*/*swTema.setOnCheckedChangeListener **{** \_, isChecked **->** // \_ pq nao vamos usar, então nao queremos que ocupe espaço na memoria  
 if (isChecked) { // Se estiver ativo  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.*MODE\_NIGHT\_YES*)  
 } else {  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.*MODE\_NIGHT\_NO*)  
 }  
**}**

Agora temos também a opção 2 fazendo um método

class MainActivity : AppCompatActivity() {  
  
 private val binding by *lazy* **{** ActivityMainBinding.inflate(*layoutInflater*)  
 **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *enableEdgeToEdge*()  
 setContentView(binding.*root*)  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->** val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)  
 insets  
 **}** *with*(binding) **{** */\*\* FORMA 1 DE FAZER  
 \* Aqui vamos pegar o objeto Switch e vamos fazer um evento de clique, que é quando ele for ativo  
 \* Ele verifica se o item foi ativo  
 \* Se foi ativo ele vai mudar o tema.  
 \*  
 \*/* swTema.setOnCheckedChangeListener **{** \_, isChecked **->** // \_ pq nao vamos usar, então nao queremos que ocupe espaço na memoria  
 if (isChecked) { // Se estiver ativo  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.*MODE\_NIGHT\_YES*)  
 } else {  
 AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.*MODE\_NIGHT\_NO*)  
 }  
 **}** */\*\* OPCAO 2  
 \* Aqui vamos exibir quando clicar no botao  
 \* Vamos exibir SWITCH marcado ou não  
 \*/* btnTema.setOnClickListener **{** switchTema()  
 **}  
  
  
 }** }  
  
 private fun switchTema() {

// Nessas variáveis, vai retornar um true ou false

val switchMarcado = binding.swTema.*isChecked* // verifcar se esta marcado  
 val toggleMarcado = binding.toggleAtivo.*isChecked* // verifcar se esta marcado  
  
  
 binding.textResultado.*text* = "Switch $switchMarcado \nToggle: $toggleMarcado"  
 }  
}