







```
// Compose!
```

```
+Sketch(
```

```
    modifier ..
```

```
) { t → // t increments somewhere?!
```

```
    translate(size.width/2f, size.height/2f) {
```

```
        drawCircle(
```

```
            center = Offset(0f, sin(t/10) * 50f),
```

```
            ...
```

```
        )
```

```
    }
```

```
}
```



// Processing - Java

void draw() {

translate(width/2, height/2);

point(0, sin(t/10) \* 50);

t++;

}

Draw/Loops

Processing vs Compore







# Draw Loops

Processing vs Compose



```
// Processing - Java
void draw() {
    translate(width/2, height/2);
    point(0, sin(t/10) * 50);
    t++;
}
```

```
// Compose!
+Sketch(
    modifier ..
) { t → // t increments somewhere?!
    translate(size.width/2f, size.height/2f) {
        drawCircle(
            center = Offset(0f, sin(t/10) * 50f),
            ...
        )
    }
}
```

# Sketch

Get a quick draw-loop using  
`AnimationState`

```
val time = remember { AnimationState(0f) }
```

```
LaunchedEffect(Unit) {  
    while (isActive) {  
        time.animateTo(  
            targetValue = time.value + speed,  
            animationSpec = animationSpec,  
            sequentialAnimation = true  
        )  
    }  
}
```