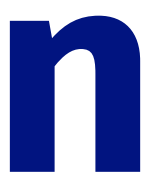
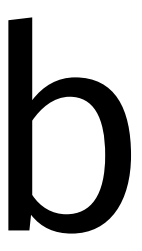
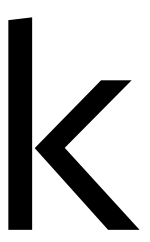
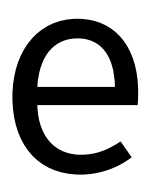
Suspend Functions

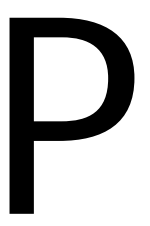
how can we make this code async?

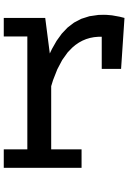


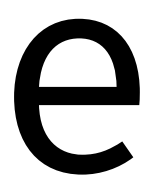

b

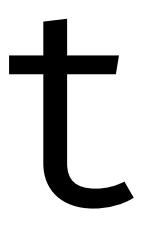




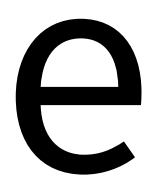




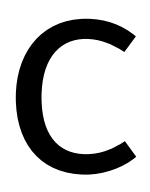


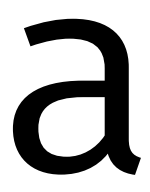


Z

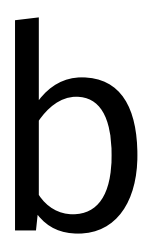


ɻ

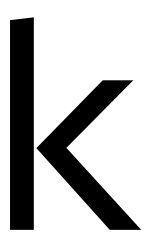

C

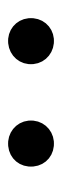


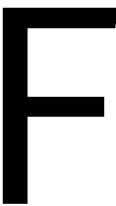
ɻ

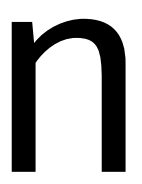


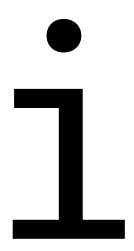
C

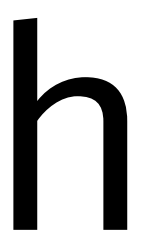


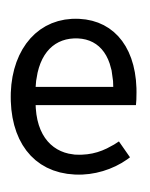


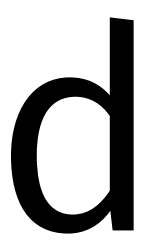


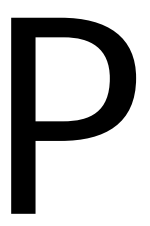


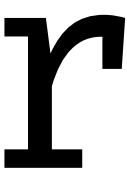


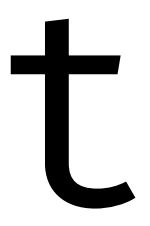


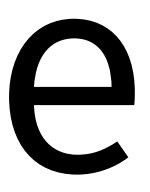



d

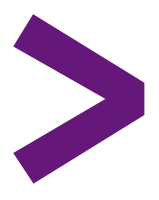


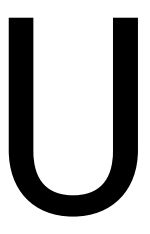


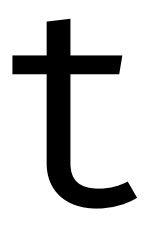




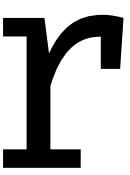
ɻ

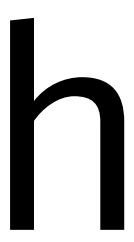


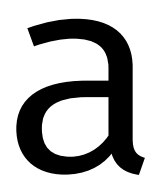


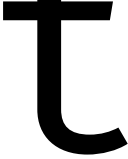


p

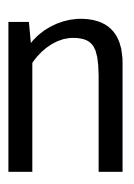


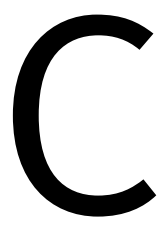


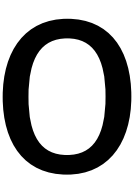




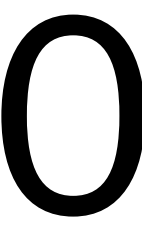
0

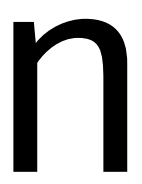


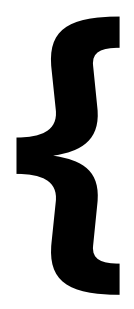


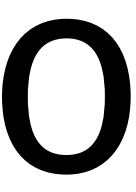


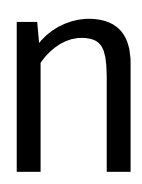
ɻ

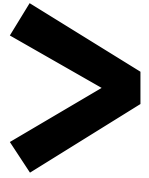


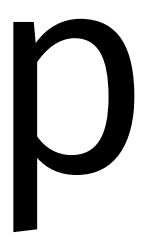


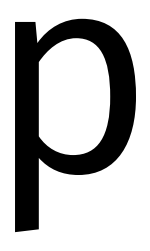


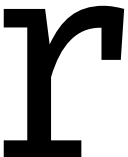


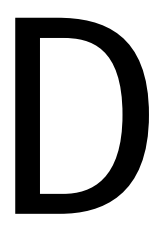


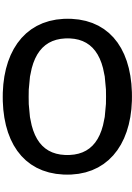


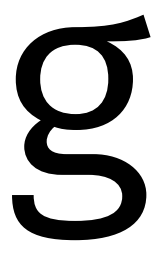


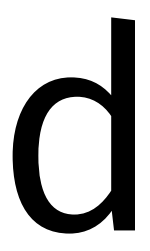


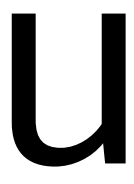


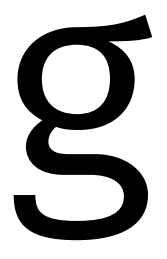

D

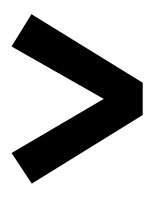


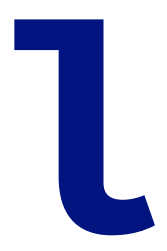


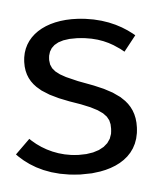

d

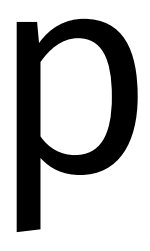




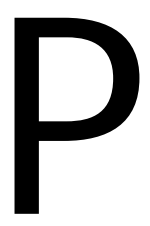


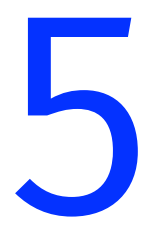


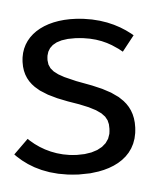


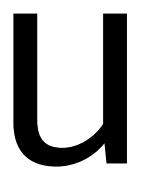


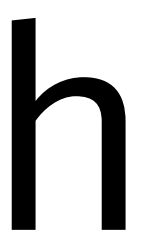
d

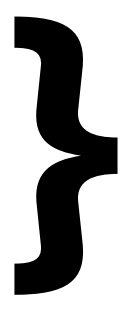


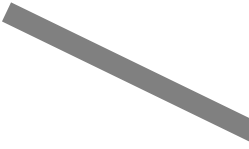


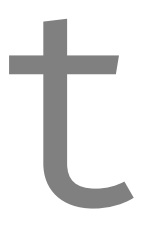




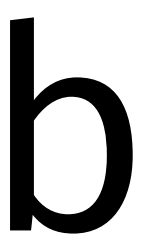


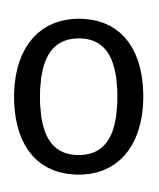


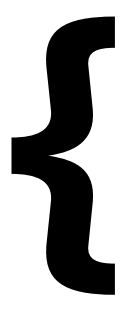


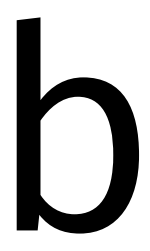


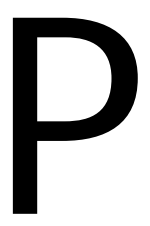
C

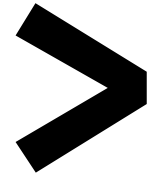


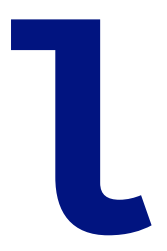


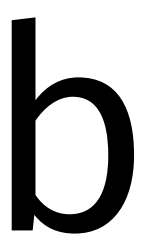


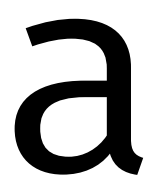

b

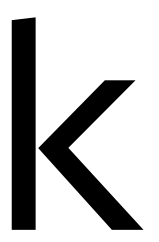


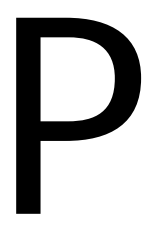


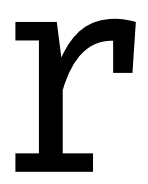


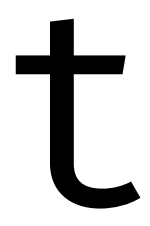

b


a





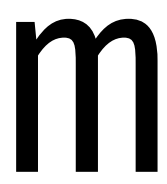


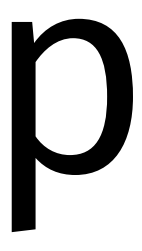


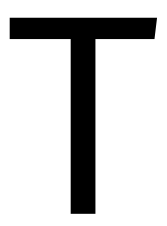
ɻ

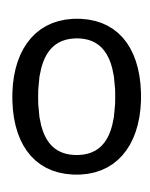
[redacted]

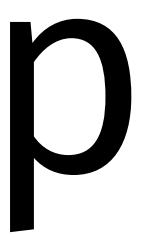
[redacted]

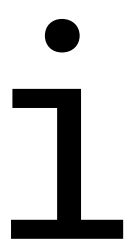


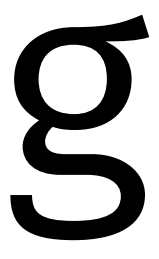


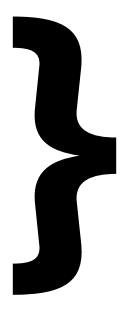


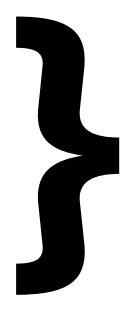


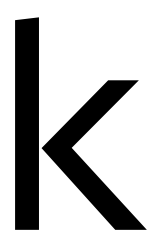






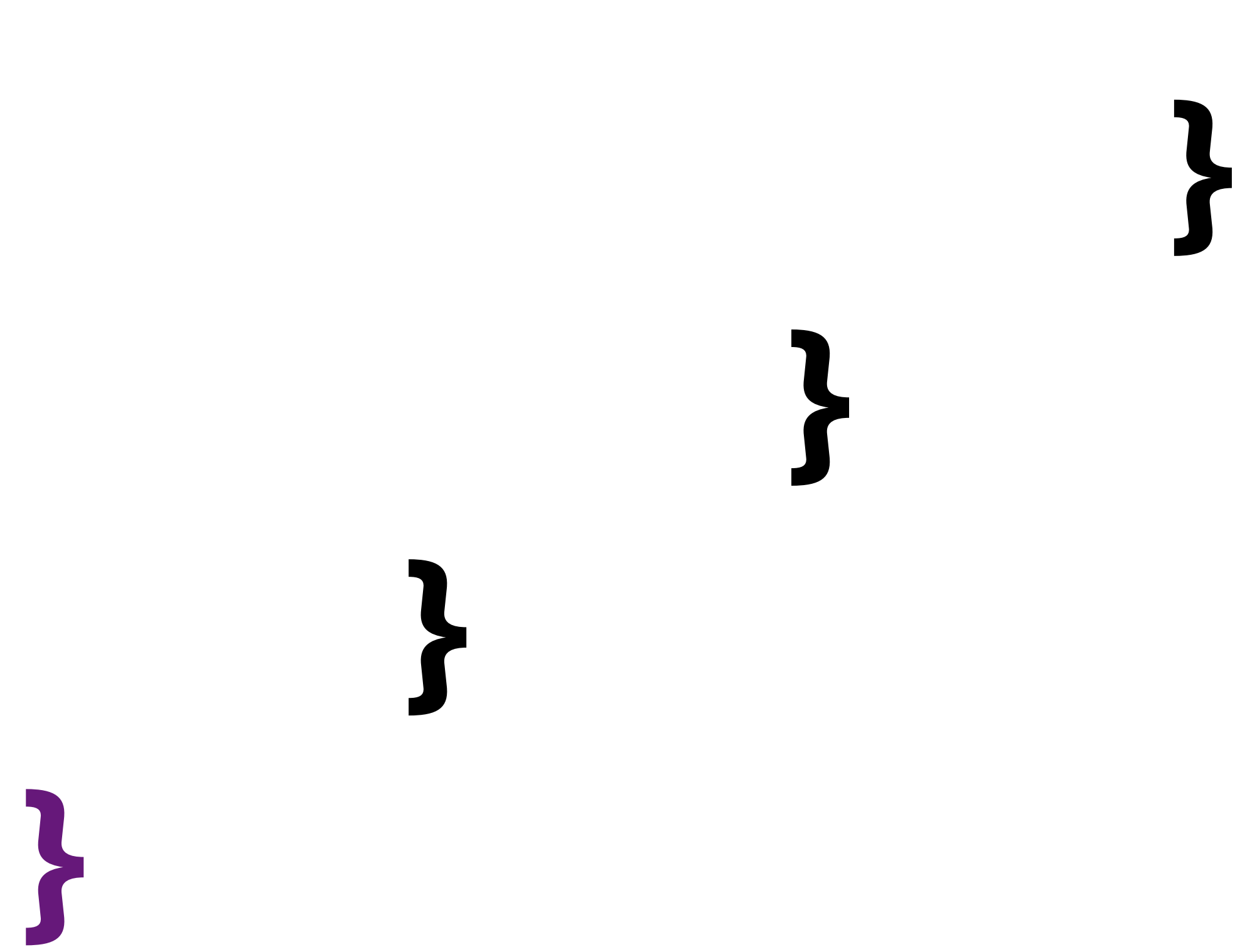






* pseudocode - it wouldn’t actually work because handling collections is a lot more complicated

Callback Hell



```
fun bakePretzels(callback: (List<FinishedPretzel>) -> Unit) {
```

```
preheatOven(ColdOven) { hotOven ->
```

```
prepareDough { dough ->
```

```
val shapedPretzels = List(5) { shapePretzel(dough) } //*this is tricky
```

```
bake(hotOven, shapedPretzels) { bakedPretzels ->
```

```
val bakedPretzel = bakePretzels.map { //*this is tricky
```

```
prepareTopping() { topping ->
```

```
addTopping(it, topping)
```

}

}

```
callback(bakedPretzels)
```

Suspend Functions

how can we make this code async?

```
fun bakePretzels(callback: (List<FinishedPretzel>) -> Unit) {
    preheatOven(ColdOven) { hotOven ->
        prepareDough { dough ->
            val shapedPretzels = List(5) { shapePretzel(dough) } //*this is tricky
            bake(hotOven, shapedPretzels) { bakedPretzels ->
                val bakedPretzels = bakePretzels.map { //*this is tricky
                    prepareTopping { topping ->
                        addTopping(it, topping)
                    }
                }
                callback(bakedPretzels)
            }
        }
    }
}
```

* pseudocode - it wouldn't actually work because handling collections is a lot more complicated

Suspend Functions

how can we make this code async?

```
fun bakePretzels(): List<FinishedPretzel> {
    val oven = preheatOven(ColdOven)
    val dough = prepareDough()
    val shapedPretzels: List<UncookedPretzel> = List(5) { shapePretzel(dough) }
    val bakedPretzels: List<CookedPretzel> = bake(oven, shapedPretzels)
    val topping: Topping = prepareTopping()
    return bakedPretzels.map { finishPretzel(it, topping) }
}
```