

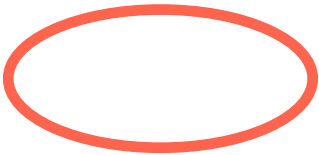
Exception Handling

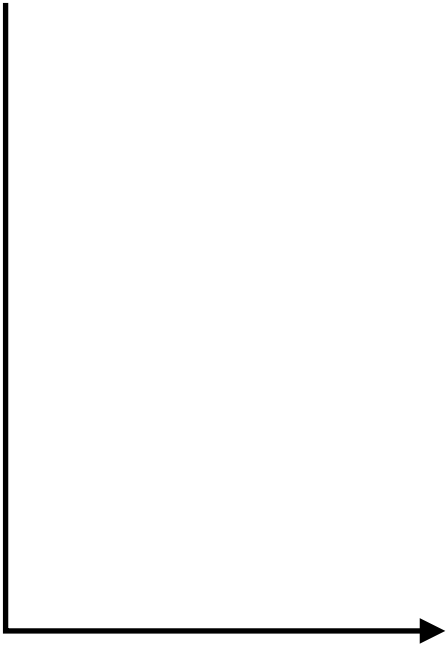
3

7

```
suspend fun bakePretzels(): List<FinishedPretzel> = coroutineScope {  
    val oven = async { preheatOven(ColdOven) }  
    val dough = async { prepareDough() }  
    val uncookedPretzels = List(5) { async { shapePretzel(dough.await()) } }  
    val bakedPretzels = async { bake(oven.await(), uncookedPretzels.awaitAll()) }  
    val topping = async { prepareTopping() }  
    bakedPretzels.await().map { finishPretzel(it, topping.await()) }  
}
```


```
throw IllegalStateException("Broken Over!");
```





Exception Handling

```
suspend fun bakePretzels(): List<FinishedPretzel> = coroutineScope {  
    val oven = async { preheatOven(ColdOven) }  
    val dough = async { prepareDough() }  
    val uncookedPretzels = List(5) { async { shapePretzel(dough.await()) } }  
    val bakedPretzels = async { bakeoven.await(), uncookedPretzels.awaitAll() }  
    val topping = async { prepareTopping() }  
    bakedPretzels.await().map { finishPretzel(it, topping.await()) }  
}
```

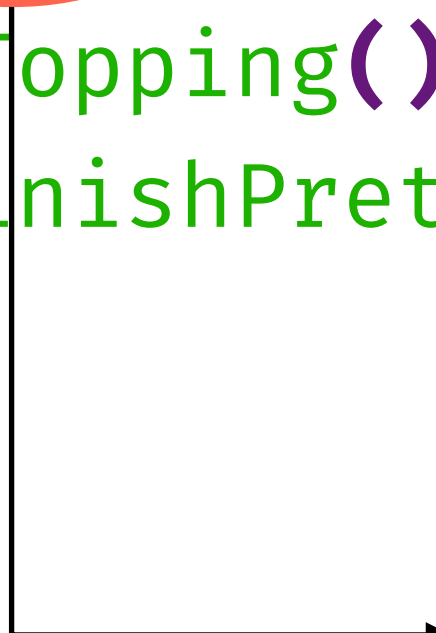


```
throw IllegalStateException("Broken Oven!")
```


Exception Handling

structured concurrency: no coroutine leaks

```
suspend fun bakePretzels(): List<FinishedPretzel> = coroutineScope {  
    val oven = async { preheatOven(ColdOven) }  
    val dough = async { prepareDough() }  
    val uncookedPretzels = List(5) { async { shapePretzel(dough.await()) } }  
    val bakedPretzels = async { bakeoven.await(), uncookedPretzels.awaitAll() }  
    val topping = async { prepareTopping() }  
    bakedPretzels.await().map { finishPretzel(it, topping.await()) }  
}
```



```
throw IllegalStateException("Broken Oven!")
```