


```
fun main() {  
    val time = measureTimeMillis {  
        thread {  
            val pretzels = bakePretzels()  
            println("Baked ${pretzels.size} pretzels")  
        }.join()  
    }  
    println("finished baking in $time ms")  
}
```

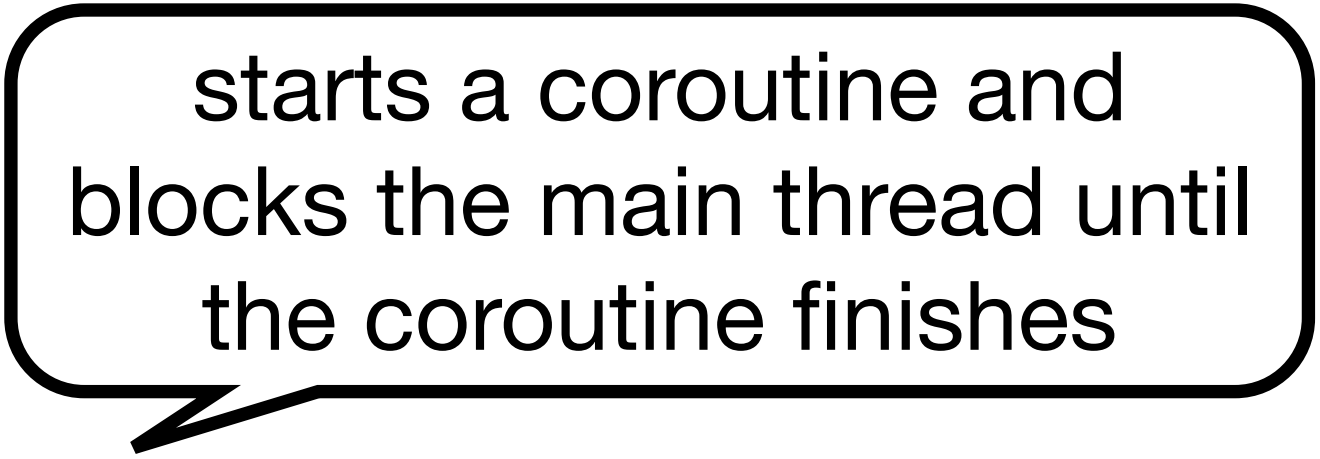
Concurrency

with routines

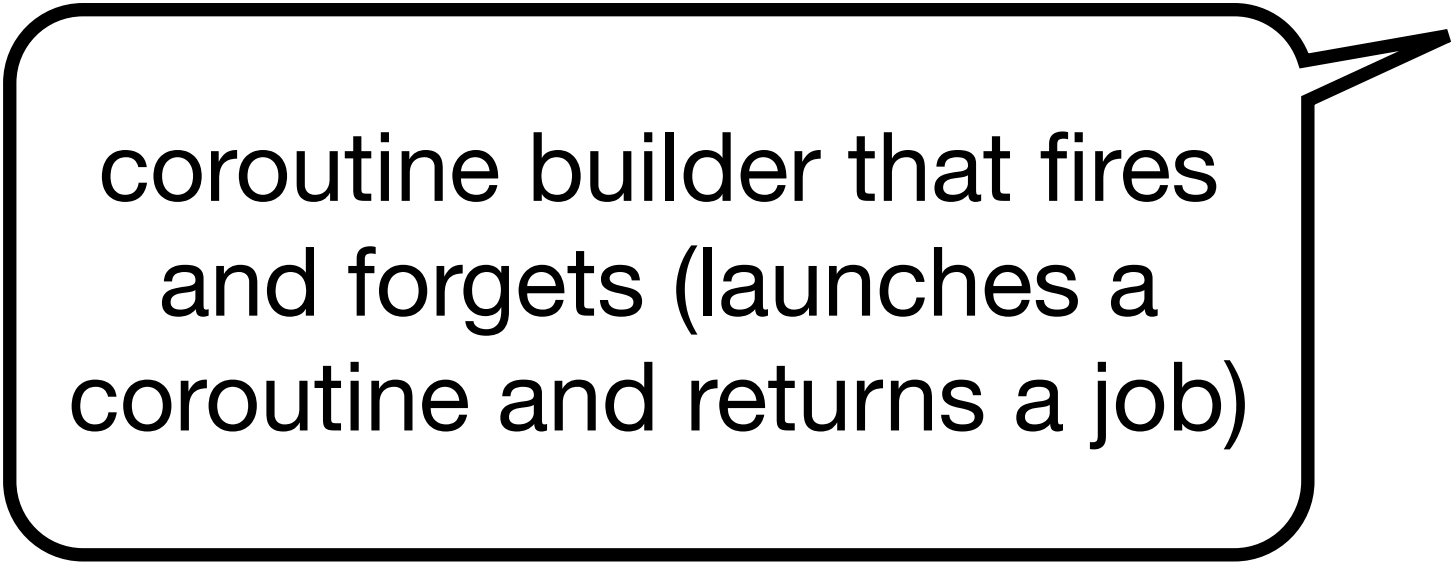


Baked 5 pretzels
finished baking in 812 ms

```
fun main() = runBlocking {  
    val time = measureTimeMillis {  
        launch {  
            val pretzels = bakePretzels()  
            println("Baked ${pretzels.size} pretzels")  
        }.join()  
    }  
    println("finished baking in $time ms")  
}
```



starts a coroutine and
blocks the main thread until
the coroutine finishes



coroutine builder that fires
and forgets (launches a
coroutine and returns a job)

alternative:

suspend fun main

Concurrency with coroutines

alternative:

suspend fun main

fun main() = **runBlocking** {

val time = measureTimeMillis {

launch {

val pretzels = bakePretzels()

 println(**"Baked \${pretzels.size} pretzels"**)

 }.join()

 println(**"finished baking in \$time ms"**)

}

starts a coroutine and
blocks the main thread until
the coroutine finishes

coroutine builder that fires
and forgets (launches a
coroutine and returns a job)

Baked 5 pretzels
finished baking in 812 ms

Concurrency

with coroutines