

Calling Blocking Code

3

6

































































































































































































































































































never use `Main`
dispatcher for blocking
code: you don't want to
block the UI

```
val job = launch(Dispatchers.IO) {  
    Thread.sleep(100) //actual blocking code  
}
```

```
val result: Deferred<String> = async(Dispatchers.IO) {  
    Thread.sleep(100) //actual blocking code  
    "finished"  
}
```

Calling Blocking Code

```
val job = launch(Dispatchers.IO) {  
    Thread.sleep(100) //actual blocking code  
}
```

```
val result: Deferred<String> = async(Dispatchers.IO) {  
    Thread.sleep(100) //actual blocking code  
    "finished"  
}
```

never use Main
dispatcher for blocking
code: you don't want to
block the UI

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 { bake(oven.await(), uncookedPretzels.awaitAll()) }  
    val topping = async { prepareTopping() }  
    bakedPretzels.await().map { finishPretzel(it, topping.await()) }  
}
```