

# Coroutines - Teaser



BNE Kotlin User Group

James Bassett  
23 Aug 2017

# Agenda

- Definition - what is a coroutine?
- Problems coroutines aim to solve
- How to get set up
- The basics
- How they work

# What's not covered

- [Coroutine libraries](#) (apart from `kotlinx.coroutines`)
  - Reactive Streams, RxJava, etc
  - Callback/future libraries: JDK8 `CompletableFuture`, Guava `ListenableFuture`, etc
- More advanced usage
  - We're just covering the basics

*Sounds like a good future meetup topic!!*



# What is a coroutine?

“A coroutine can be thought of as an instance of suspendable computation, i.e. the one that can suspend at some points and later resume execution possibly on another thread”

- [Coroutines Spec](#)

“threads are sometimes called lightweight processes”

- [Oracle Java Tutorials](#)

“coroutines are light-weight threads”

- [Kotlin Coroutines Guide](#)

# What is a coroutine?

**TLDR: Coroutines are *really* lightweight! (apparently)**

[Demo]

# Process vs Thread vs Coroutine (Theory time!)

- **Process**
  - Own memory space (OS feature)
  - Example: JVM runs as a single process
- **Thread**
  - Share resources within a process (OS feature)
  - Example: Java app may utilise many threads
- **Coroutine**
  - Suspendable functions (Language i.e. Kotlin feature)
  - Can run on any thread (and jump between them)

# Problems coroutines try to solve

- When you can't afford threads
- Async code and callback hell (take it away Roman!)
  - <https://vimeo.com/221264980> (8:21-15:52)
- Generators (yield)
  - [Demo]

# How to start using coroutines

- Project [setup](#)
  - Standard Kotlin setup
  - Add kotlinx-coroutines-core library
  - Disable experimental warning
  - Bonus points for using [Gradle Kotlin DSL](#) :)
- Take a test drive with Coroutines by Example
  - <https://github.com/Kotlin/kotlinx.coroutines/blob/master/coroutines-guide.md>
  - Let's go through the very basics together!



# How do coroutines work?

- Continuation passing style + state machine + Magic!
  - <https://vimeo.com/221264980> (15:58-19:48)
- Read spec for more details
  - <https://github.com/Kotlin/kotlin-coroutines/blob/master/kotlin-coroutines-informal.md#implementation-details>

# How do I find out more???

- Reading
  - [Informal specification](#)
- Watching
  - [Introduction to Kotlin coroutines - Roman Elizarov 2017](#)
- Doing!
  - [Kotlin coroutines tutorial](#)
  - [Coroutines by example guide](#)

# Questions?

