# Introduction to Gradle

Benedikt Ritter, 16.04.2019, JUG Essen

<https://speakerdeck.com/britter/introduction-to-gradle-at-jug-essen>

<https://github.com/britter/introduction-to-gradle>

<https://autoweird.fm/>

## Features

* Declarative DSL in Groovy and Kotlin
* Build description, with opt-in scriptability & plugin system
  + Inline scripting
  + Pre-compiled script plugins
  + Plugin portal for reusable logic
* Flexible dependency management
* Focus on performance
  + Build in parallelism
  + Gradle daemon
  + UP-TO-DATE checking
  + Local cache
  + Remote cache (nur in Gradle Enterprise)
* Discoverable builds via build scans: <https://scans.gradle.com/>
  + Gradle
  + Maven

## Notizen

* Gradle nimmt Maven Central nicht automatisch 🡪 explizite Angabe nötig
* Gradle daemon: JVM läuft bereits im Hintergrund (keine warm-up Zeiten)
* Kein „clean“ nötig: Gradle nutzt up-to-date checking
* gradle tasks [--all] zeigt alle Tasks
* sourceSet: z.B. src/main und src/test – weitere frei definierbar (z.B. src/integrationtest)
* Skripte können ausgelagert werden, eingebunden werden (apply()) und präkompiliert werden

## Gradle Enterprise

* Remote caching
* Build Scans auf Unternehmensserver (freie Version: Build Scans werden zu gradle.com propagiert)