



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

I-Seminar (BTI-7311)

Prof. Dr. Olivier Biberstein

Maven

Apache Maven is a software project management. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information. Using maven we can build and manage any Java based project.

The goal of this study is explore the various aspects of Maven, the role of the POM and its plugins, dependencies, etc. How Maven can be used in other contexts than Java. What are the differences with existing building tools such as Ant, Gradle, and others.

- ▶ <https://maven.apache.org/>
- ▶ Mastering Apache Maven 3, Prabath Siriwardena

Gradle : just another "make" tool ?

Gradle is an open-source build automation system that builds upon the concepts of Apache Ant and Apache Maven and introduces a Groovy-based domain-specific language (DSL) instead of the XML form used by Apache Maven for declaring the project configuration[Wikipedia]

It is also the official building system used for Android applications. The goal of this presentation of to explore and explain the main ideas of gradle, explain some of the syntax of gradle build files and shows how in integrates with other existing build system (likes Maven or Ant)

- ▶ <https://en.wikipedia.org/wiki/Gradle>
- ▶ Gradle in Action, Benjamin Muschko, Manning

Haskell programming language

Haskell is often considered as the Rolls-Royce of the pure functional programming language.

The goal of this study is not to produce a tutorial of the language (there are plenty on the Internet) but explore briefly the syntax of the language and focus on the numerous functional aspects that are supported by Haskell. It is important to explain clearly each aspect that is addressed and then present some simple examples that illustrate the selected aspect.

- ▶ <https://www.haskell.org/>
- ▶ Haskell, the craft of functional programming, Simon Thompson, Pearson Education.

Scala vs Kotlin

Scala and Kotlin are two modern object-oriented programming languages that support many aspect of functional programming. They are both often considered as the possible successor of Java. This study aims at exploring the differences (advantages and drawbacks) between the two languages. The idea is not to compare the syntax but to focus on the various interesting features of both the languages. The comparison might also cover some aspects of Java.

- ▶ <https://www.scala-lang.org/>
- ▶ <https://kotlinlang.org/>

DevOps

DevOps ("development" and "operations") is a software engineering culture and practice that aims at unifying software development (Dev) and software operation (Ops). The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and infrastructure management. DevOps aims at shorter development cycles, increased deployment frequency, and more dependable releases, in close alignment with business objectives. The goal of this study is to explore the DevOps culture, the concepts and the tools used in this approach.

- ▶ <https://en.wikipedia.org/wiki/DevOps>
- ▶ The DevOps Handbook, Gene Kim, Patrick Debois, John Willis, and Jez Humble

Neural Networks

The term neural network was traditionally used to refer to a network or circuit of neurons. The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term may refer to either biological neural networks, made up of real biological neurons, or artificial neural networks, for solving artificial intelligence (AI) problems. The goal of this study is to explore the world of artificial neural networks, classify the various types of such system and address the numerous fields of application.

- ▶ https://en.wikipedia.org/wiki/Neural_network
- ▶ Neural Networks and Learning Machines, Third Edition, Simon Haykin

NoSQL: Document-oriented database and MongoDB

A document-oriented database, or document store, is a computer program designed for storing, retrieving and managing document-oriented information, also known as semi-structured data. Document-oriented databases are one of the main categories of NoSQL databases. The objective of this study is explore the world of Not Only SQL databases in particular document-oriented database that contrast strongly with traditional relational databases. A well known product called MongoDB is a typical document-oriented database that will be used to illustrate the concepts such databases in this study.

- ▶ <https://en.wikipedia.org/wiki/NoSQL>
- ▶ <https://www.mongodb.com/>
- ▶ https://en.wikipedia.org/wiki/Apache_CouchDB

NoSQL: Graph Database and Neo4j

A graph database is a database that uses graph structures for semantic queries with nodes, edges and properties to represent and store data. A key concept of the system is the graph (or edge or relationship), which directly relates data items in the store. The relationships allow data in the store to be linked together directly, and in many cases retrieved with one operation.

The objective of this study is explore the world of Not Only SQL databases in particular database that use graph structure to store information. A well known product called Neo4j is a typical graph database that will be used to illustrate the concept of graph database in this study.

- ▶ https://en.wikipedia.org/wiki/Graph_database
- ▶ <https://neo4j.com/>
- ▶ <https://franz.com/agraph/allegrograph/>

Docker vs VM

Virtual Machine are used to isolate some potentially dangerous or unstable application from their host. They allows also users to share a single physical machine without having to share the software configuration. On the other hand, containers like Docker provide a probably less secure isolation environment but allowing an much more easier and efficient way to run application into a controlled environment without having to configure all the needed tools.

In the project you have to compare these two ways to securely run applications. Performance, simplicity (or complexity) are some of the topics that should be analysed.

- ▶ [https://en.wikipedia.org/wiki/Docker_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))
- ▶ https://en.wikipedia.org/wiki/Virtual_machine