

PySpark for Big Data

Audience PySpark for Big Data

The course PySpark for Big Data is intended for developers and upcoming Data Analysts who want to learn how to use Apache Spark from Python.

Prerequisites training PySpark for Big Data

To participate in this course, some experience with programming is beneficial for understanding. Prior knowledge of Python or big data handling with Apache Spark is not required.

Realization course PySpark for Big Data

The theory is treated on the basis of presentations. Illustrative demos are used to clarify the concepts discussed. There is ample opportunity to practice and alternate theory and practice. The course times are from 9.30 am to 4.30 pm.

Certification course PySpark for Big Data

Participants receive an official certificate PySpark for Big Data after successful completion of the course.

Duration: 3 days

Price: € 2450

[Open Schedule](#)

PySpark



Data Analysis with PySpark



PySpark

Content PySpark for Big Data

In the course PySpark for Big Data participants learn to use Apache Spark from Python. Apache Spark is a Framework for parallel processing of big data. With PySpark, Apache Spark is integrated with the Python language.

Spark Architecture

The course PySpark for Big Data discusses the architecture of Spark, the Spark Cluster Manager and the difference between Batch and Stream Processing.

Hadoop

After a discussion of the Hadoop Distributed File System, parallel operations and working with RDDs, Resilient Distributed Datasets are discussed in the course PySpark for Big Data. The configuration of PySpark applications via SparkConf and SparkContext is also explained.

MapReduce en SQL

Extensive consideration is given to the possible operations on RDDs, including map and reduce. The use of SQL in Spark is also discussed. The GraphX library is discussed and DataFrames is discussed. Iterative algorithms are also treated.

Mlib library

Finally the course PySpark for Big Data pays attention to machine learning with the Mlib library.

Modules PySpark for Big Data

Module 1 : Python Primer	Module 2 : Spark Intro	Module 3 : HDFS
Python Syntax Python Data Types List, Tuples, Dictionaries Python Control Flow Functions and Parameters Modules and Packages Comprehensions Iterators and Generators Python Classes Anaconda Environment Jupyter Notebooks	What is Apache Spark? Spark and Python PySpark Py4j Library Data Driven Documents RDD's Real Time Processing Apache Hadoop MapReduce Cluster Manager Batch versus Stream Processing PySpark Shell	Hadoop Environment Environment Setup Hadoop Stack Hadoop Yarn Hadoop Distributed File System HDFS Architecture Parallel Operations Working with Partitions RDD Partitions HDFS Data Locality DAG (Direct Acyclic Graph)
Module 4 : SparkConf	Module 5 : SparkContext	Module 6 : RDD's
SparkConf Object Setting Configuration Properties Uploading Files SparkContext.addFile Logging Configuration Storage Levels Serialize RDD Replicate RDD partitions DISK_ONLY MEMORY_AND_DISK MEMORY_ONLY	Main Entry Point Executor Worker Nodes LocalFS SparkContext Parameters Master RDD serializer batchSize Gateway JavaSparkContext instance Profiler	Resilient Distributed Datasets Key-Value pair RDDs Parallel Processing Immutability and Fault Tolerance Transformation Operations Filter, groupBy and Map Action Operations Caching and persistence PySpark RDD Class count, collect, foreach, filter map, reduce, join, cache
Module 7 : Spark Processing	Module 8 : Broadcast and Accumulator	Module 9 : Algorithms
SQL support in Spark Spark 2.0 Dataframes Defining tables Importing datasets Querying data frames using SQL Storage formats JSON / Parquet GraphX GraphX library overview GraphX APIs	Performance Tuning Serialization Network Traffic Disk Persistence MarshalSerializer Data Type Support Python's Pickle Serializer DStreams Sliding Window Operations Multi Batch and State Operations	Iterative Algorithms Graph Analysis Machine Learning API mllib.classification Random Forest Naive Bayes Decision Tree mllib.clustering mllib.linalg mllib.regression