

PySpark voor Big Data

Doelgroep Cursus PySpark voor Big Data

De cursus PySpark voor Big Data is bedoeld voor developers en aankomende Data Analisten die Apache Spark willen leren gebruiken vanuit Python.

Voorkennis training PySpark voor Big Data

Om aan deze cursus deel te nemen is kennis enige ervaring met programmeren bevorderlijk voor de begripsvorming. Voorafgaande kennis van Python of big data handling met Apache Spark is niet nodig.

Uitvoering cursus PySpark voor Big Data

De theorie wordt behandeld aan de hand van presentaties. Illustratieve demo's worden gebruikt om de behandelde concepten te verduidelijken. Er is voldoende gelegenheid om te oefenen en afwisseling van theorie en praktijk. De cursustijden zijn van 9.30 tot 16.30.

Certificering cursus PySpark voor Big Data

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat PySpark voor Big Data.

Duur: 3 dagen

Prijs: € 2450

[Open Rooster](#)

PySpark



Data Analysis with PySpark



PySpark

Inhoud PySpark voor Big Data

In de cursus PySpark voor Big Data leren de deelnemers Apache Spark vanuit Python te gebruiken. Apache Spark is een Framework voor parallelle processing van big data. Met PySpark wordt Apache Spark geïntegreerd met de Python taal.

Spark Architectuur

In de cursus PySpark voor Big Data komt aan de orde komt de architectuur van Spark, de Spark Cluster Manager en het verschil tussen Batch en Stream Processing.

Hadoop

Na een bespreking van het Hadoop Distributed File System wordt ingegaan op parallelle operaties and het werken met RDD's, Resilient Distributed Datasets. De configuratie van PySpark applicaties via SparkConf en SparkContext komt eveneens aan bod in de cursus PySpark voor Big Data.

MapReduce en SQL

Uitgebreid wordt ingegaan op de mogelijke operaties op RDD's waaronder map en reduce. Ook komt het gebruik van SQL in Spark aan de orde. De GraphX library wordt besproken en er wordt ingegaan op DataFrames. Verder komen iteratieve algorithmen aan de orde.

Mlib library

Tenslotte wordt in de cursus PySpark voor Big Data aandacht besteed aan machine learning met de Mlib library.

Modules PySpark voor 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