APACHE Spark

# SparkSession vs SparkContext

## SparkSession

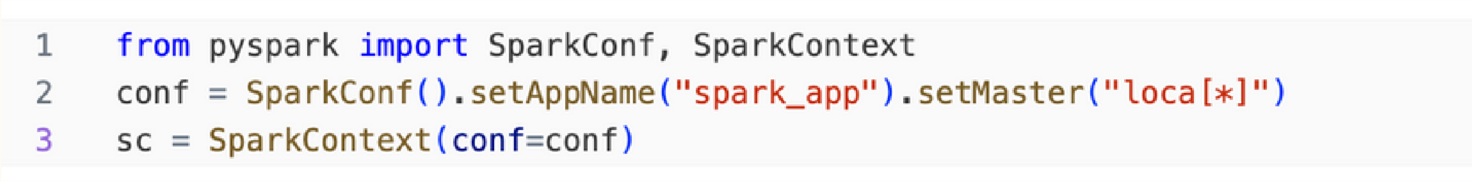
- Serves as the primary interface for interacting with data, creating DataFrames, and supersedes SQLContext and HiveContext.
- Allows multiple instances of SparkSession within an application.
- Preferred for working with high-level Spark data structures like DataFrames and Datasets.
- Facilitates access to stored data in Spark and supports various operations on it.

## SparkContext

- Serves as the initial access point to Spark's functionalities, managing task execution, data distribution, and job scheduling across a cluster.
- Restricts creation to a single instance within a Spark application.
- Primarily handles lower-level operations, such as RDDs, accumulators, and broadcast variables.
- Used to access the fundamental Spark environment and perform operations like configuring executor memory

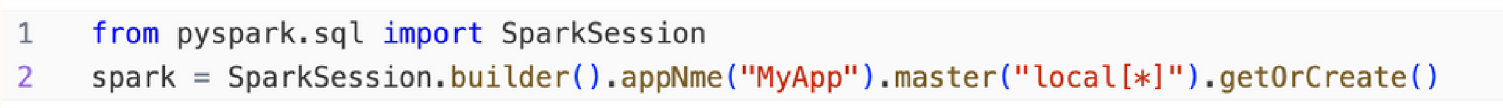
# SparkSession vs SparkContext

## Create a SparkContext



```python
from pyspark import SparkConf, SparkContext
conf = SparkConf().setAppName("spark_app").setMaster("loca[*]")
sc = SparkContext(conf=conf)
```

## Create a SparkSession



```python
from pyspark.sql import SparkSession
spark = SparkSession.builder().appNme("MyApp").master("local[*]").getOrCreate()
```