**Day 1: Spark & PySpark Fundamentals**

* **Understand Spark Architecture:** Master driver, executors, cluster manager, RDDs, and DataFrames.
* **Installation & Setup:** Install PySpark locally or use Databricks/Google Dataproc for hands-on practice.
* **First PySpark Script:** Run a simple word count or data transformation to verify your setup.

**Resources:**

* Official [PySpark documentation](https://spark.apache.org/docs/latest/api/python/" \t "_blank)
* Databricks Community Edition (for free cloud notebook)

**Day 2: DataFrames & Transformations**

* **DataFrame API:** Learn to create, inspect, and manipulate DataFrames (read/write CSV, Parquet, JSON).
* **Transformations & Actions:** Practice select, filter, groupBy, agg, and withColumn.
* **Comparison with Dataflow:** Note similarities/differences in lazy evaluation, pipeline construction, and schema handling[1](https://www.projectpro.io/compare/google-cloud-dataflow-vs-apache-spark)[2](https://stackoverflow.com/questions/33518104/google-dataflow-vs-apache-spark).

**Exercise:**

* Load a dataset, perform basic ETL (cleaning, filtering, aggregation).

**Day 3: Advanced Data Processing**

* **Joins & Window Functions:** Practice join, window, rank, lead/lag.
* **UDFs:** Write and apply User Defined Functions for custom logic.
* **Partitioning & Caching:** Learn how to optimize performance with repartition, cache, and persist.

**Exercise:**

* Build a multi-step ETL pipeline using joins and window functions.

**Day 4: Spark SQL & Integration**

* **Spark SQL:** Use SQL queries on DataFrames, register temp views.
* **Integration:** Read/write from/to external sources (HDFS, S3, GCS, JDBC).
* **Comparison:** Relate to Dataflow’s BigQuery/Cloud Storage connectors[1](https://www.projectpro.io/compare/google-cloud-dataflow-vs-apache-spark)[2](https://stackoverflow.com/questions/33518104/google-dataflow-vs-apache-spark)[4](https://dev.to/stack-labs/serverless-spark-on-gcp-how-does-it-compare-with-dataflow--2o8n).

**Exercise:**

* Query data using both DataFrame API and Spark SQL.

**Day 5: Spark Streaming & Structured Streaming**

* **Streaming Basics:** Understand DStreams vs. Structured Streaming.
* **Source & Sink:** Read from Kafka, files, or sockets; write to files or databases.
* **Windowed Operations:** Implement sliding/tumbling windows, watermarking.

**Exercise:**

* Build a simple streaming pipeline (e.g., word count from a socket or file stream).

**Day 6: Performance Optimization & Monitoring**

* **Tuning:** Learn about partition sizing, broadcast joins, and memory management.
* **Monitoring:** Use Spark UI to inspect jobs, stages, and executors[1](https://www.projectpro.io/compare/google-cloud-dataflow-vs-apache-spark)[6](https://www.projectpro.io/compare/google-cloud-dataflow-vs-apache-spark-streaming).
* **Error Handling:** Debugging and handling failed jobs.

**Exercise:**

* Tune a slow pipeline and observe improvements via Spark UI.

**Day 7: Project & Review**

* **Mini Project:**
  + Choose a real-world dataset (e.g., NYC taxi data).
  + Implement an end-to-end ETL or streaming pipeline.
  + Optimize and document your work.
* **Review:**
  + Summarize key differences with Dataflow (e.g., cluster management, pipeline design, monitoring)[1](https://www.projectpro.io/compare/google-cloud-dataflow-vs-apache-spark)[2](https://stackoverflow.com/questions/33518104/google-dataflow-vs-apache-spark)[4](https://dev.to/stack-labs/serverless-spark-on-gcp-how-does-it-compare-with-dataflow--2o8n)[6](https://www.projectpro.io/compare/google-cloud-dataflow-vs-apache-spark-streaming).
  + List remaining questions or advanced topics for future study (e.g., MLlib, GraphX).

**Tips**

* Leverage your GCP experience: Try running PySpark on Google Dataproc or integrate with GCS/BigQuery.
* Use notebooks (Jupyter, Databricks) for interactive exploration.
* Focus on distributed data processing patterns and Spark’s in-memory advantages.