Difference between JPA, Hibernate and Spring Data JPA

1) **JPA (Java Persistence API):** JPA is a **specific** set of rules and interfaces for mapping Java objects entities)to relational database tables. It is part of **Jakarta EE formerly Java EE** and doesn't do anything by itself — it requires a **provider** like Hibernate.
   Example:

```
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
@Entity
public class Student {
@Id
private Long id;
private String name;
}
```

2) **Hibernate:** Hibernate is a framework (and the most popular implementation of JPA).It provides powerful ORM features like lazy loading, caching, automatic SQL generation, we can use Hibernate with or without JPA.
   Example:

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

 public class Main {
    public static void main(String[] args) {
        SessionFactory factory = new Configuration().configure().buildSessionFactory();
        Session session = factory.openSession();

        Student student = new Student();
        student.setId(1L);
        student.setName("Arun");
        session.beginTransaction();
        session.save(student);  // Hibernate saves to DB
        session.getTransaction().commit();
        session.close();
    }
}
```

   How it works: Developers must handle things like starting sessions, managing transactions, opening/closing database connections, and dealing with exceptions on their own.

3) **Spring Data JPA:** Spring Data JPA is a part of the Spring Framework that builds on top of JPA (and usually Hibernate) to reduce boilerplate code and make data access as simple as writing an interface.
   Example:

```
// Entity
@Entity
public class Student {
    @Id
    private Long id;
    private String name;
}
```

```
// Repository
public interface StudentRepository extends JpaRepository<Student, Long> {
}

// Controller
@RestController
public class StudentController {
  @Autowired
  StudentRepository repo;

  @PostMapping("/students")
  public Student save(@RequestBody Student s) {
    return repo.save(s);
  }
}
```

Conclusion

- JPA is a set of rules (specification), Hibernate is a tool that follows those rules (implementation), and Spring Data JPA is a Spring-based shortcut built on top of JPA.
- JPA doesn't provide ready-to-use code, Hibernate requires more manual coding, while Spring Data JPA automates most of the work with minimal code.
- JPA needs a third-party tool to work, hibernate itself handles database mapping and sessions, and Spring Data JPA internally uses Hibernate to simplify data access.