->Within Spring applications, you can use these @Enable<Technology> annotations to

follow the *convention over configuration* pattern, making your apps easier to develop and

maintain, and without worrying too much about its configuration.

->@EnableAutoConfiguration annotation -> AutoConfigurationImportSelector.class -> getCandidateConfigurations() -> loads spring.factories from META-INF folder.

->spring.factories defines all the auto-configuration

classes that are used to set up any configuration that your application needs for running.

->can check in nay technology auto configuration class, there will be @conditional annotations, import is applied only when conditions mentioned in @conditional are satisfied.

->auto configuration uses classpath to figure out what is needed for our app.

* @SpringBootApplication. This annotation is actually the @ComponentScan, @Configuration, and @EnableAutoConfiguration annotations. You already know everything about the @EnableAutoConfiguration from the previous sections.
* SpringApplication. This class provides the bootstrap for the Spring Boot application that is executed in the main method. You need to pass the class that is executed.

To disable banner, we can use spring.main.banner-mode = off