

□□□□□ □□□□□□□□□□ □□□□□□ का अनुवाद

क्या आप जानते हैं कि...

[illegible]

□□□□ □□□□□□□□ □□□□□ □□ □□□ □□□□□?

□□□□□ □□□□□□□□□□ □□□□□□ □□ □□□□ □□□□□ □□ □□□□ □□□□□□□□ □□□□□ □□□□:

- [illegible]

□□□□ □□□□□□□□ □□□□ □□ □□□□ □□□□ □□□□

000 000 000000 00 00000 000000 00000000 0000 00000 00000 00 000000:

1. pom.xml 文件

00000000 00 <build><plugins> 00000000 0000 00000000 00000000 pom.xml 0000. 0000 00 00000000 00 000000
 spring-boot-starter-parent 000 000 0000 0000, 00000000 0000000 00 00000 000; 0000 00 000000000000 00000000
 00000000.

```
<build>
    <plugins>
        <plugin>
            <groupId>org.apache.maven.plugins</groupId>
            <artifactId>maven-checkstyle-plugin</artifactId>
            <version>3.1.2</version> <!-- Latest version se replace karein -->
        </plugin>
    </plugins>
</build>
```

2. Maven을 사용하여 checkstyle.xml을 검사하는 방법

이 예제를 빌드할 때 checkstyle.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 checkstyle.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 checkstyle.xml을 사용하여 코드를 검사하는 방법을 보여줍니다.

이 예제를 빌드할 때 checkstyle.xml을 사용하여 코드를 검사하는 방법을 보여줍니다.

```
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-checkstyle-plugin</artifactId>
      <version>3.1.2</version>
      <configuration>
        <configLocation>checkstyle.xml</configLocation>
      </configuration>
    </plugin>
  </plugins>
</build>
```

3. Maven을 사용하여 google_checks.xml을 검사하는 방법

이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다.

```
<configLocation>google_checks.xml</configLocation>
```

이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다.

```
<dependencies>
  <dependency>
    <groupId>com.puppcrawl.tools</groupId>
    <artifactId>checkstyle</artifactId>
    <version>8.44</version>
  </dependency>
</dependencies>
```

이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다.

이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다. 이 예제를 빌드할 때 google_checks.xml을 사용하여 코드를 검사하는 방법을 보여줍니다.

□ 00000000000 000 00000000000: 00000000000 00 000000 000000 000 0000000000000 000000 00 00000 000000 00 00000:

```
mvn checkstyle:check
```

□ 000000 00 0000000 000 0000000: 00000000 00000, 0000000 verify 000000 00 000000 00000 00000 0000. 0000 0000000:

```
mvn verify
```

00000000 00000000000 0000000 00 00000 0000000 00 00000 00 0000000 00 00000:

```
mvn checkstyle:checkstyle
```

000000000 00000000000 target/site/checkstyle.html 00000 00000000000 00000 00000.

00000000 00 000000000000 0000000 00000000

0000 00000000 00 00000000000 00 <configuration> 000000000 00000 00000000 0000 0000000 00000 0000000 pom.xml 00000:

□ 00000 00 00000000000: 000000000 00000, 0000000 00000 00000 0000 00000 0000000000000 000000 00000. 0000 000000000 0000000 00 00000:

```
<configuration>
  <failOnViolation>>false</failOnViolation>
</configuration>
```

□ 000000 00 000000000 00 000000000 0000000: 000000000 00000000 00000-00 0000000 0000000 00000 00000:

```
<configuration>
  <includes>**/*.java</includes>
  <excludes>**/generated/**/*.java</excludes>
</configuration>
```

□ 000000000000 0000000000 0000 00000000: 00000000 00000000 00000000000 0000000 00 0000000 000000000 000000000 0000000 000000:

```
<configuration>
  <violationSeverity>warning</violationSeverity>
</configuration>
```

checkstyle.xml

checkstyle.xml 是检查代码格式的工具，它可以根据你定义的规则来检查代码。你可以通过配置 checkstyle.xml 来定义这些规则。下面是一个简单的配置示例：

```
<?xml version="1.0"?>
<!DOCTYPE module PUBLIC
    "-//Checkstyle//DTD Checkstyle Configuration 1.3//EN"
    "https://checkstyle.org/dtds/configuration_1_3.dtd">

<module name="Checker">
    <module name="TreeWalker">
        <module name="JavadocMethod"/>
        <module name="MethodName"/>
        <module name="ConstantName"/>
    </module>
</module>
```

checkstyle 使用

checkstyle 可以通过以下几种方式使用：

- 1. 通过命令行使用：在命令行中运行 `mvn checkstyle:check` 来检查代码格式。
- 2. 通过 IDE 使用：大多数 IDE 都支持 checkstyle 插件，可以在 IDE 中配置 checkstyle.xml 来检查代码格式。
- 3. 通过 CI 工具使用：可以在 CI 工具中配置 checkstyle 来检查代码格式。

checkstyle 配置

- 1. 通过命令行使用：在命令行中运行 `mvn checkstyle:check` 来检查代码格式。
- 2. 通过 IDE 使用：大多数 IDE 都支持 checkstyle 插件，可以在 IDE 中配置 checkstyle.xml 来检查代码格式。
- 3. 通过 CI 工具使用：可以在 CI 工具中配置 checkstyle 来检查代码格式。

checkstyle 规则

checkstyle 提供了许多规则来检查代码格式，以下是一些常见的规则：

1. `FileTabCharacter`：检查文件中的制表符。
2. `Header`：检查文件头。
3. `Indentation`：检查代码缩进。
4. `LineLength`：检查代码行长度。

0000 00000000 00000-00000000, 000000000000 0000 00 000000000 000000 00000 000000 000000 0000 00000000 00000000 00 000000!