Creating custom annotations in Java involves defining the annotation using the @interface keyword and specifying its retention policy and target. Here's a step-by-step guide:

**Step-by-Step Guide to Creating Custom Annotations**

1. **Define the Annotation**:
   * Use the @interface keyword to define your annotation.
   * Example:
2. public @interface MyCustomAnnotation {
3. String value() default "default value";
4. int number() default 0;
5. }
6. **Specify Retention Policy**:
   * Use the @Retention meta-annotation to specify how long the annotation should be retained.
   * Example:
7. import java.lang.annotation.Retention;
8. import java.lang.annotation.RetentionPolicy;
9. @Retention(RetentionPolicy.RUNTIME)
10. public @interface MyCustomAnnotation {
11. String value() default "default value";
12. int number() default 0;
13. }
14. **Specify Target**:
    * Use the @Target meta-annotation to specify where the annotation can be applied (e.g., methods, fields, classes).
    * Example:
15. import java.lang.annotation.ElementType;
16. import java.lang.annotation.Target;
17. @Retention(RetentionPolicy.RUNTIME)
18. @Target({ElementType.METHOD, ElementType.FIELD})
19. public @interface MyCustomAnnotation {
20. String value() default "default value";
21. int number() default 0;
22. }

**Using Custom Annotations**

1. **Apply the Annotation**:
   * Apply your custom annotation to methods, fields, or classes as specified.
   * Example:
2. public class MyClass {
3. @MyCustomAnnotation(value = "Hello", number = 42)
4. public void myMethod() {
5. // Method implementation
6. }
7. }
8. **Process the Annotation**:
   * Use reflection to process the annotation at runtime.
   * Example:
9. import java.lang.reflect.Method;
10. public class AnnotationProcessor {
11. public static void main(String[] args) throws Exception {
12. Method method = MyClass.class.getMethod("myMethod");
13. MyCustomAnnotation annotation = method.getAnnotation(MyCustomAnnotation.class);
14. if (annotation != null) {
15. System.out.println("Value: " + annotation.value());
16. System.out.println("Number: " + annotation.number());
17. }
18. }
19. }

**Resources for Further Reading**

* [Creating a Custom Annotation in Java](https://www.baeldung.com/java-custom-annotation) [[1]](https://www.baeldung.com/java-custom-annotation)
* [How to Create Your Own Annotations in Java](https://www.geeksforgeeks.org/how-to-create-your-own-annotations-in-java/) [[2]](https://www.geeksforgeeks.org/how-to-create-your-own-annotations-in-java/)

Would you like more details on any of these steps or examples of more complex annotations?

**References**

[1] [Creating a Custom Annotation in Java - Baeldung](https://www.baeldung.com/java-custom-annotation)

[2] [How to Create Your Own Annotations in Java? - GeeksforGeeks](https://www.geeksforgeeks.org/how-to-create-your-own-annotations-in-java/)