Creating Documentation Comments

In this section:

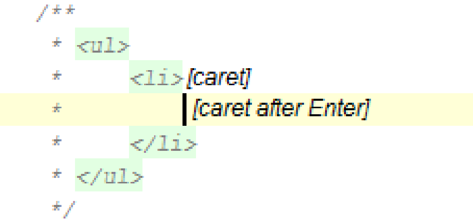
- Creating documentation comments for a method or function
- Creating tags in a documentation comment block

To create a documentation comment for a method or function

1. Place the caret *before* the declaration.
2. Type the opening block comment `/**`, and press **`Enter`**.
3. Add meaningful description of parameters and return values.

Please note the following:

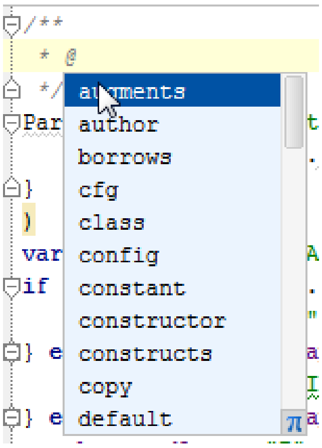
- IntelliJ IDEA checks syntax in the documentation comments and treats it according to the Error settings.
- If the entered text contains HTML tags, the closing tag will be automatically added after typing `>`, provided that this behavior is enabled in the editor settings.
- When typing in a documentation comment, the caret automatically moves to an expected position. For example:



```
/**
 * <ul>
 *     <li>[caret]
 *         [caret after Enter]
 *     </li>
 * </ul>
 */
```

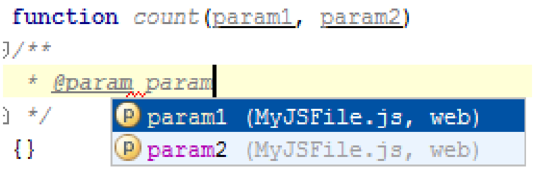
To create tags in a documentation comment block

1. In a comment block, select the desired empty line and type `@` character.
2. Press **`Ctrl+Space`**, or just wait for Code Completion to display the suggestion list:



```
/**
 * @
 */
Par
}
)
var
if
} e
} e
```

3. Select a tag from the suggestion list. For example, you can specify the parameters type, or return type.
4. If a certain tag has several values, press **`Ctrl+Space`** after the tag, and select the desired value from the suggestion list. For example, IntelliJ IDEA suggests to select the desired parameter name.



```
function count(param1, param2)
/**
 * @param param
 */
{}
```

Documentation comment can be created with the dedicated action **Fix Doc Comment**. It can be invoked by means of Find Action command.

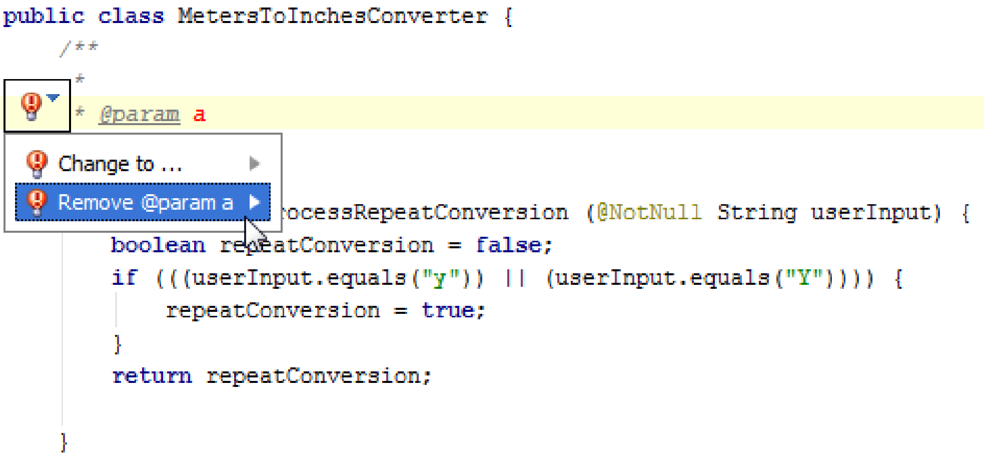
Press `Ctrl+Shift+A`, with the caret somewhere within a class, method, function, or field, which should be documented, and enter the action name **Fix Doc String**. The missing documentation stub with the corresponding tags is added. For example:

```
                /**
                *
                * @param userInput
                * @return
                */
               static boolean processRepeatConversion (@NotNull String userInput) {
     boolean repeatConversion = false;
     if (((userInput.equals("y")) || (userInput.equals("Y")))) {
     repeatConversion = true;
     }
     return repeatConversion;

 }
```

The next case lays with fixing problems in the existing documentation comments.

For example, if a method signature has been changed, IntelliJ IDEA highlights a tag that doesn't match the method signature, and suggests a quick fix.



```
public class MetersToInchesConverter {
    /**
     *
     * @param a

                     ocessRepeatConversion (@NotNull String userInput) {
        boolean repeatConversion = false;
        if (((userInput.equals("y")) || (userInput.equals("Y")))) {
            repeatConversion = true;
        }
        return repeatConversion;

    }
```

Note that the action has no keyboard shortcut bound with it. You can do it as described in the section Configuring Keyboard Shortcuts.

See Also

Procedures:

- Commenting and Uncommenting Blocks of Code
- Viewing Inline Documentation
- Creating JSDoc Comments

Reference:

- Inspections
- Editor. Smart Keys

Web Resources:

- Developer Community