


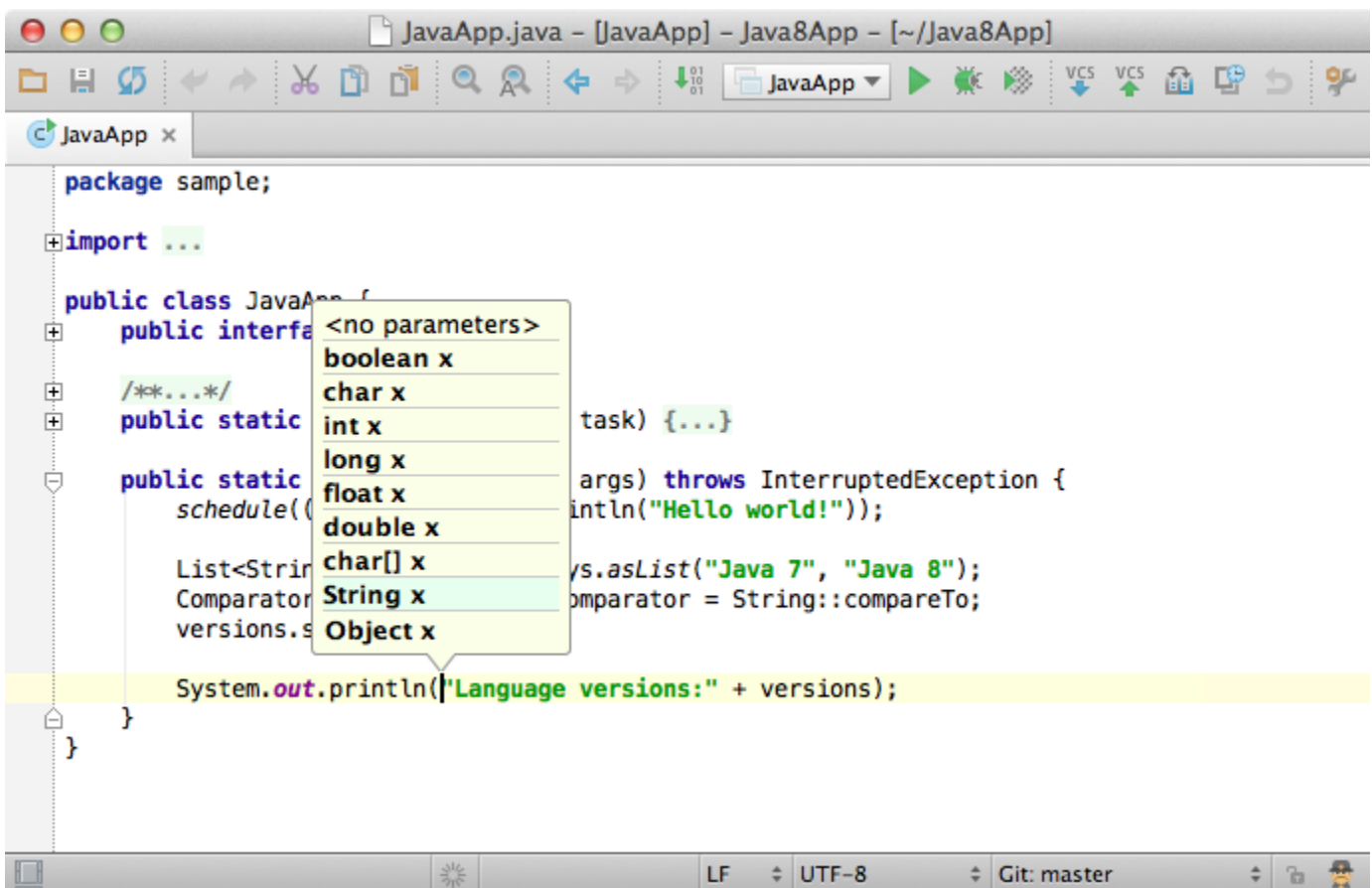
Quick Popups

 **Redirection Notice**
This page will redirect to <https://www.jetbrains.com/idea/help/discover-intellij-idea.html#QuickPopups>.

 Quick popups are helpful for checking additional information related to the symbol at the caret. Here is a list of popups you should know if you want to be more productive.

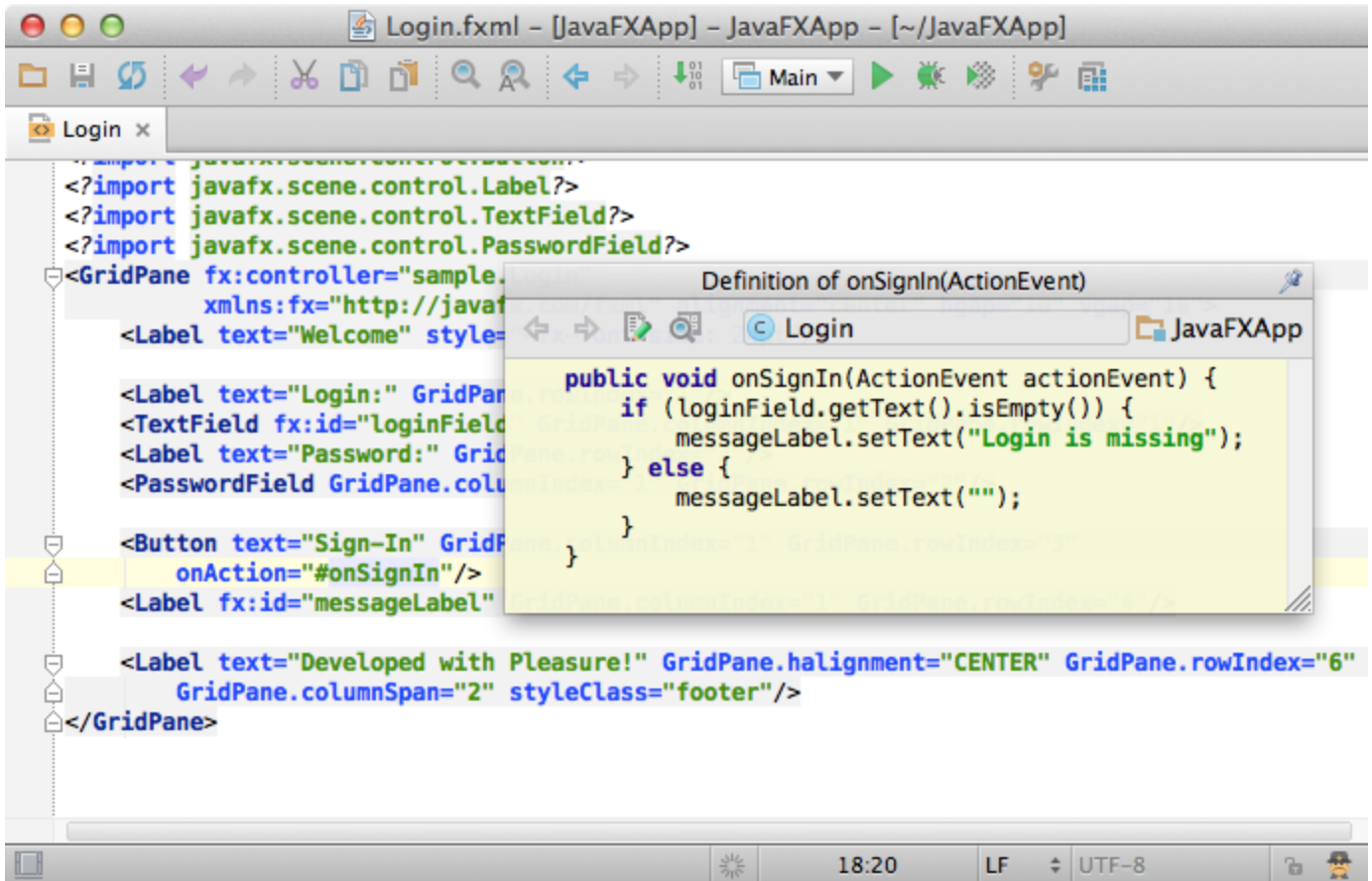
1. Parameter info

The information on parameters required for a particular method is available via `Ctrl+P` (`Cmd+P`) shortcut. The popup shows the matched overloaded method, highlighting the current parameter.



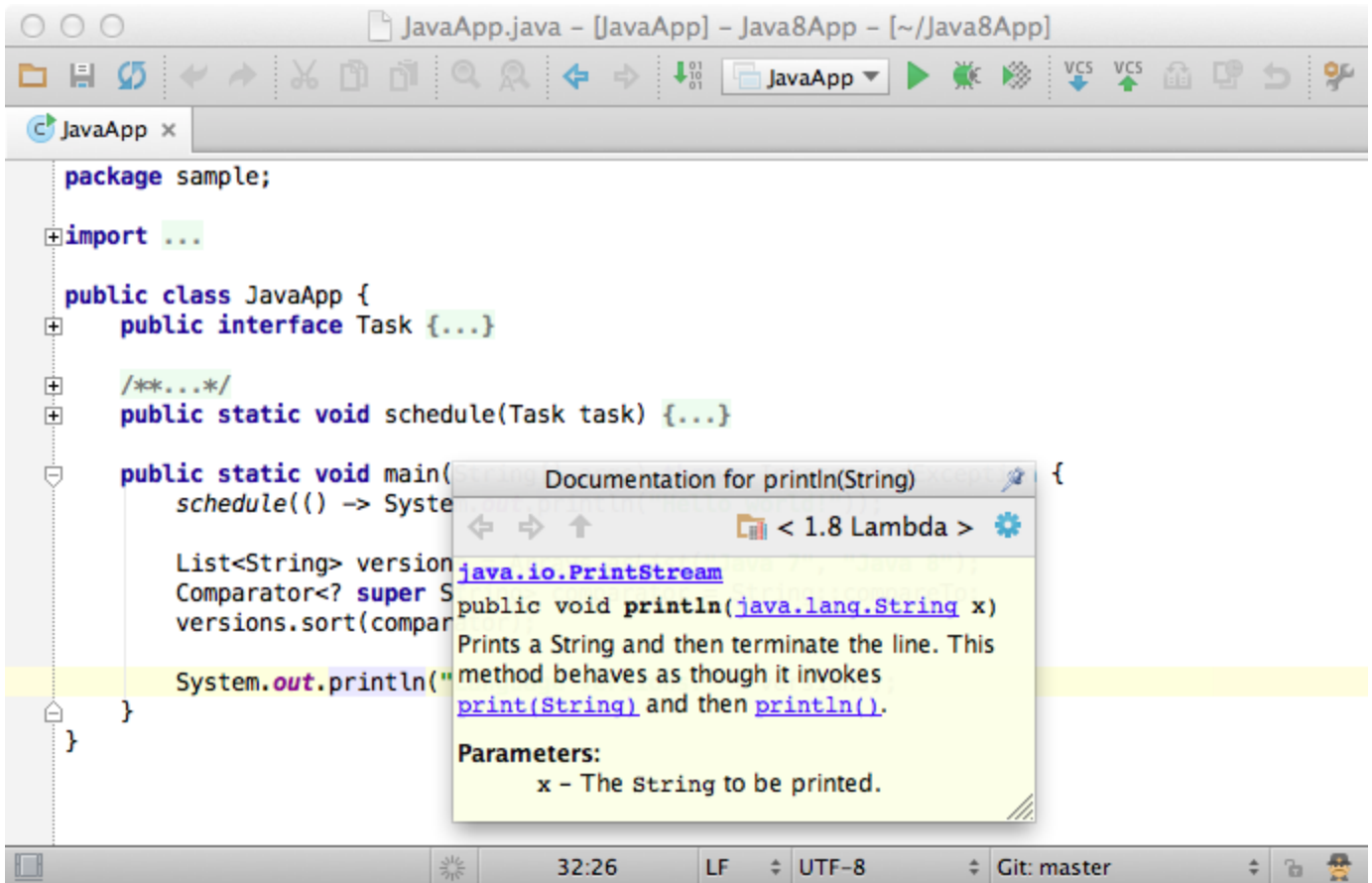
2. Quick Definition

If you want to quickly look at the implementation of a symbol at the caret, you can call the Quick Definition by pressing `Shift+Ctrl+I` (`Shift+Cmd+I`).



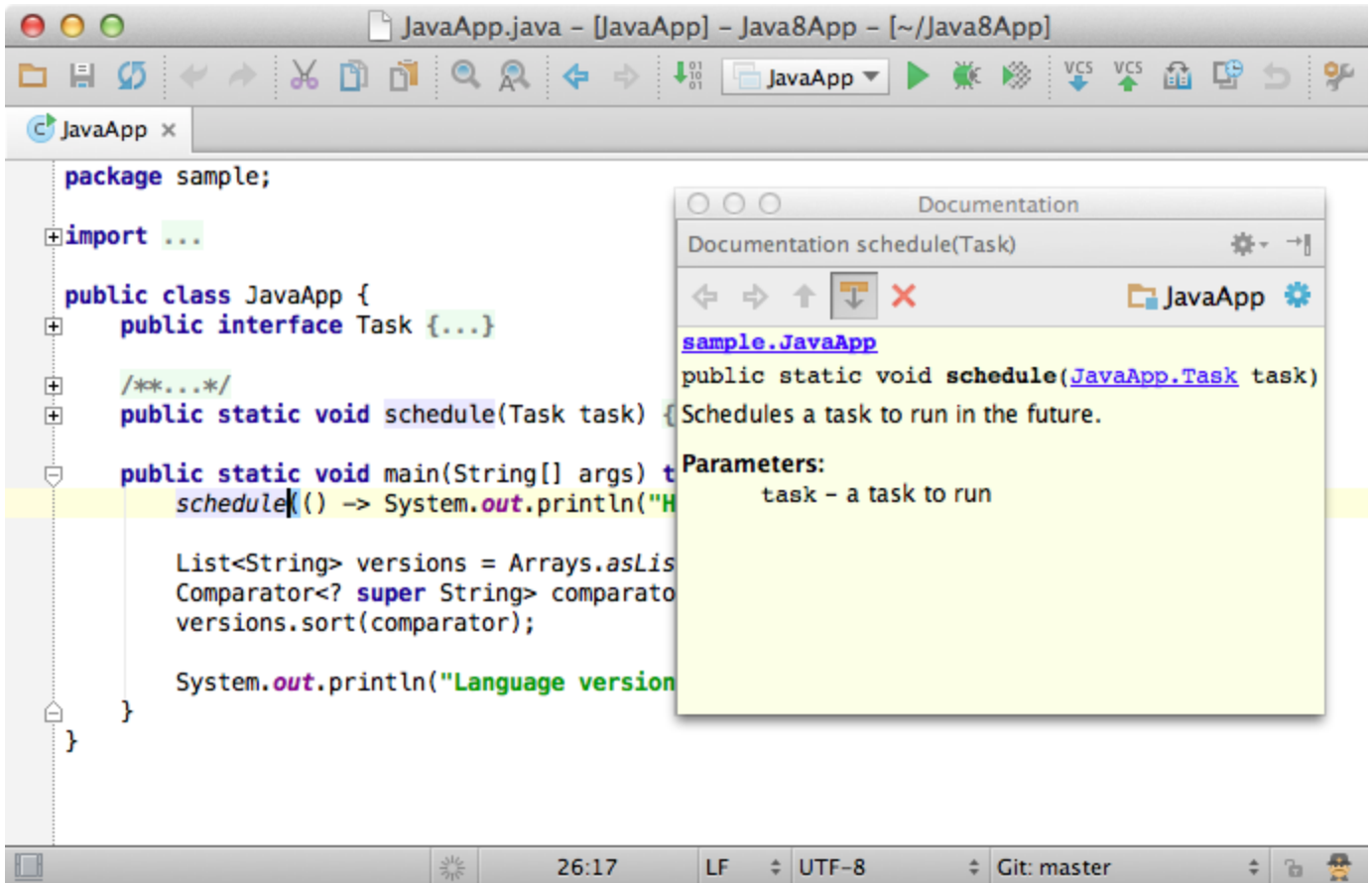
3. Documentation

The documentation popup is one of the most used editor features. To see the documentation, just press Ctrl+Q (Ctrl+J for Mac).



4. Pin popups

If you want to keep the documentation open, click the Pin button in the top right corner. It will show the documentation for the current symbol.



5. Dock popup

If you don't like float popup windows, you can dock them as a tool window by using the button in the top right-hand corner.

The screenshot shows an IDE window titled "JavaApp.java - [JavaApp] - Java8App - [~/Java8App]". The code editor displays the following Java code:

```
public static void main(String[] args) throws InterruptedException {
    schedule(() -> System.out.println("Hello world!"));

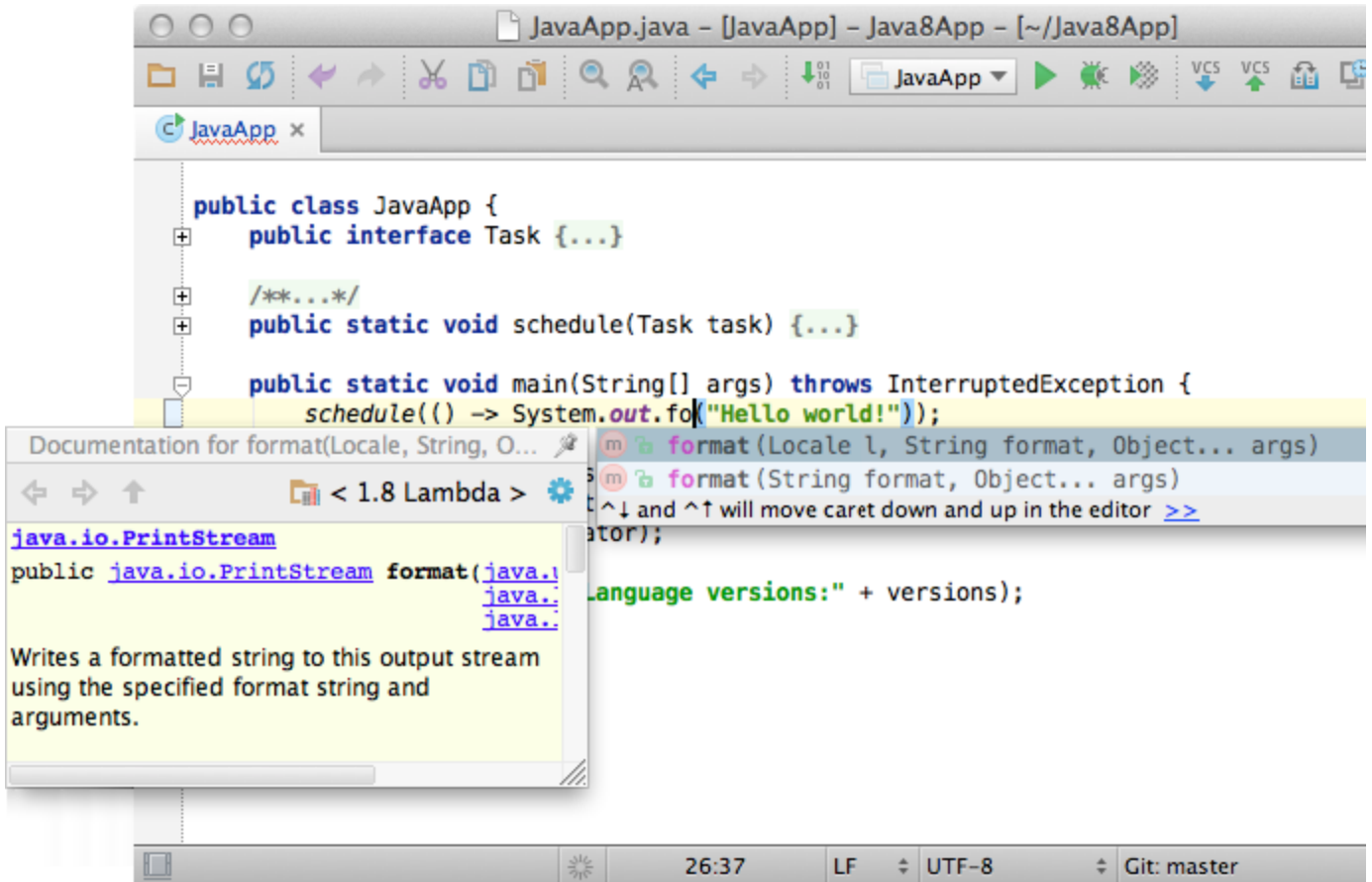
    List<String> versions = Arrays.asList("Java 7", "Java 8");
    Comparator<? super String> comparator = String::compareTo;
    versions.sort(comparator);

    System.out.println("Language versions:" + versions);
}
```

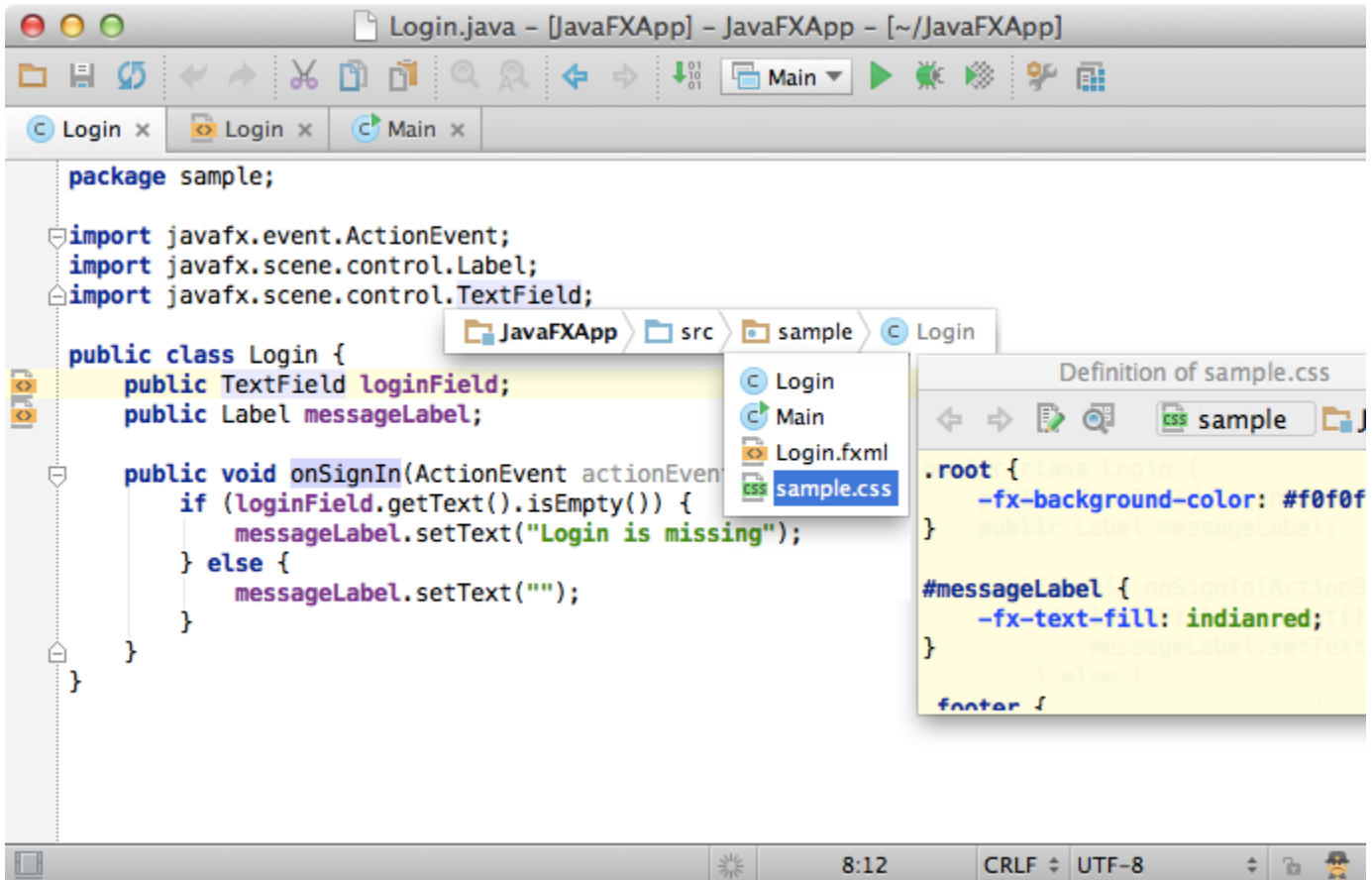
The documentation panel below the code shows the signature and description for `java.util.Arrays.asList(T... a)`. The signature is: `@java.lang.SafeVarargs public static <T> java.util.List<T> asList(T... a)`. The description states: "Returns a fixed-size list backed by the specified array. (Changes to the returned list 'write through' to the array.) This method acts as bridge between array-based and collection-based APIs, in combination with `Collection.toArray`. The returned list is serializable and implements `RandomAccess`." It also provides an example: `List<String> stooges = Arrays.asList("Larry", "Moe", "Curly");`

6. Popups in other components

It is well-known that Quick popups are available for symbols in the editor. However, they are also available for items in any other list, via the same shortcuts:

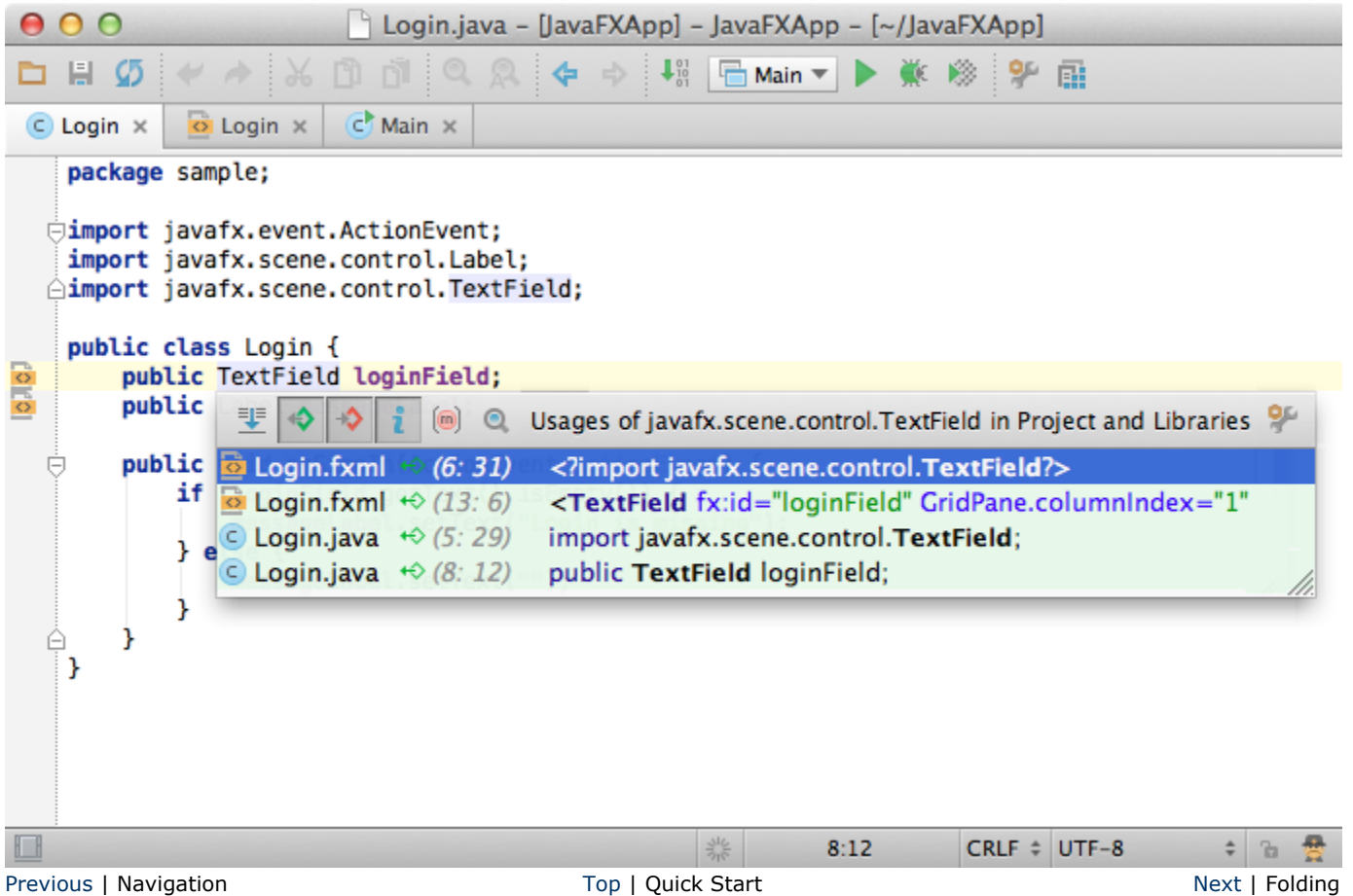


You can use quick popups even if you browse the elements from the Navigation bar via `Alt+Home` (`Alt+for` for Mac).



7. Show usages

Show usages is a very simple but powerful feature. Just press `Alt+Ctrl+F7` (`Alt+Cmd+F7` for Mac) and get a list of usages for any symbol at the caret:



The screenshot shows an IDE window titled "Login.java - [JavaFXApp] - JavaFXApp - [~/JavaFXApp]". The code in the editor is as follows:

```
package sample;

import javafx.event.ActionEvent;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;

public class Login {
    public TextField loginField;
    public
    public
    if
} e
}
}
```

A tooltip window titled "Usages of javafx.scene.control.TextField in Project and Libraries" is displayed over the code. It lists the following usages:

- Login.fxml (6: 31) <?import javafx.scene.control.TextField?>
- Login.fxml (13: 6) <TextField fx:id="loginField" GridPane.columnIndex="1"
- Login.java (5: 29) import javafx.scene.control.TextField;
- Login.java (8: 12) public TextField loginField;

The IDE interface includes a toolbar at the top with various icons, a tab bar with "Login x", "Login x", and "Main x", and a status bar at the bottom with "Previous | Navigation", "Top | Quick Start", "8:12", "CRLF", "UTF-8", and "Next | Folding".