

Basic Code Completion. Completing Names and Keywords

Basic code completion helps you complete the names of classes, methods, fields, and keywords within the visibility scope. When you invoke code completion, IntelliJ IDEA analyses the context and suggests the choices that are reachable from the current position of the caret.

If basic code completion is applied to a part of a field, parameter, or variable declaration, IntelliJ IDEA suggests a list of possible names with regard to the type of the item.

Invoking *Basic code completion* for the second time shows the names of classes available through the module dependencies.

When invoked once more (for the third time in a row), the suggestion list expands to the whole project, regardless of the dependencies.

This action completes the names of classes and interfaces, searching through the entire project. If the desired class is not yet imported, it will be imported automatically.

Code completion covers supported file types and custom file types. However, IntelliJ IDEA does not recognize the structure of custom file types and suggests completion options regardless of whether a specific type is appropriate in the current context.

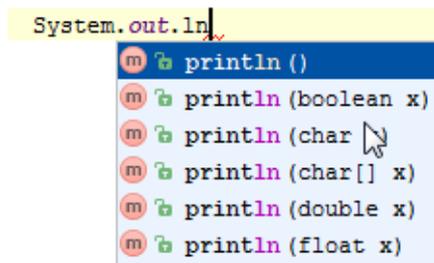
To automatically complete names in the current visibility scope, follow these general steps:

1. Type a name or a part of a name.

While typing, use the following techniques:

- Narrow down the [suggestion list](#) by typing any part of a word, even the characters somewhere in the middle, or invoking code completion after a dot separator. IntelliJ IDEA will show suggestion list that includes occurrences with all the possible positions of the entered characters.

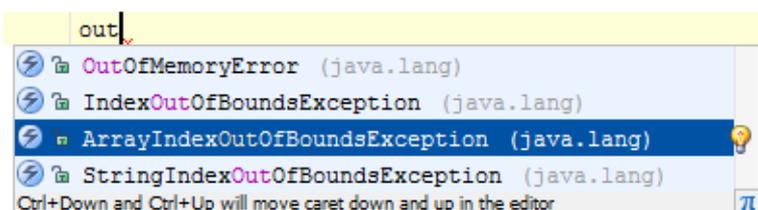
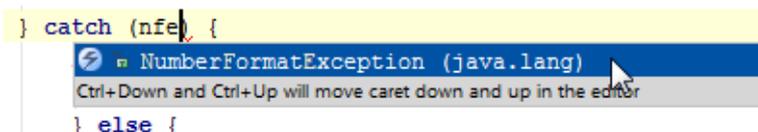
This makes the use of the asterisk wildcard unnecessary:



- In case of *CamelCase* or *snake_case* names, type the initial letters only. So doing, IntelliJ IDEA automatically recognizes and matches the initial letters.

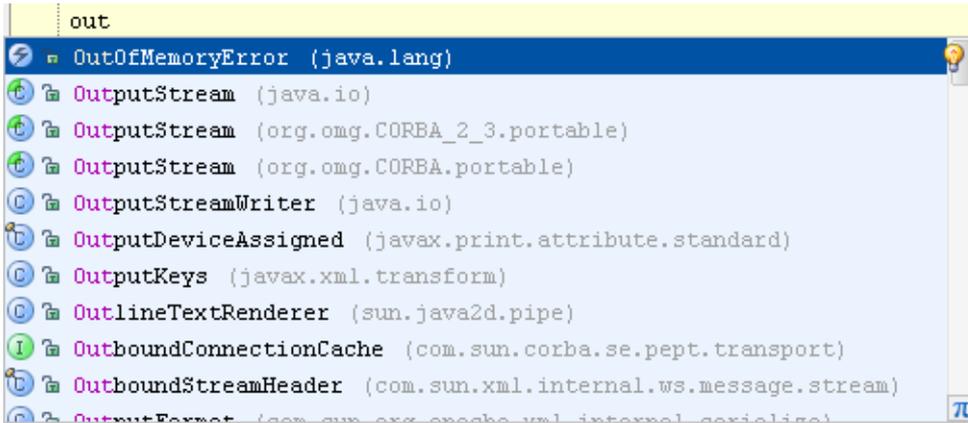
2. Do one of the following:

- On the main menu, choose **Code | Completion | Basic**.
- Press **Ctrl+Space**.



3. If necessary, press `Ctrl+Space` for the second time.

This shows all classes, available through the module dependencies. The second completion shows also static fields and methods.



This action produces the same effect as pressing `Ctrl+Alt+Space`.

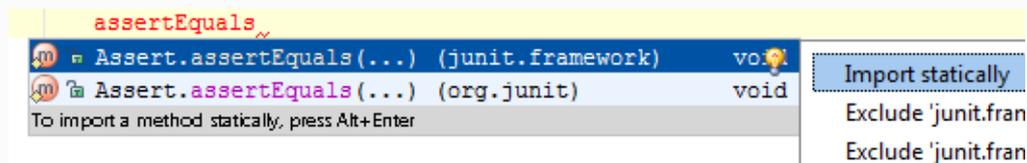
When invoked for the third time, basic code completion expands the suggestion list to all classes throughout the project, regardless of the dependencies.

4. Select the desired class from the suggestion list, and do one of the following:

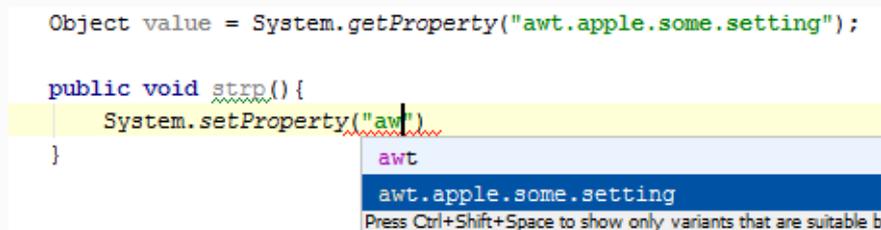
- Press `Enter` or double-click the desired choice .
- Press `Tab` .

- IntelliJ IDEA makes it possible to choose between adding `Class.methodName()` at the insertion point, and adding a static import statement. In the first variant, just [complete selection](#).

In the second variant, press `Alt+Enter` in the suggestion list, and choose *Import statically*.



- The option **Autocomplete common prefix (Settings | IDE Settings | Code Completion)** makes IntelliJ IDEA look for common prefixes and automatically complete them in the editor. This feature is useful for classes with numerous similar-named members.
- If a suggestion list shows *n variants* of a class, type a dot, and the next suggestion list after dot will show all the possible qualified names of the members. After selecting the desired one, import statement will be added automatically.
- Suggestion list includes literals, already defined in a file:



See Also

Procedures:

- [Using Suggestion List](#)
- [Navigating to Class, File or Symbol by Name](#)

Reference:

- [Editor. Code Completion](#)

External Links:

- [Chained expression completion](#) 

Web Resources:

- [Developer Community](#) 