Top 30 Eclipse Keyboard Shortcuts for Java Programmers

Want to learn more about keyboard shortcuts in Java. Check out this tutorial of 30 different shortcuts you can use with Eclipse IDE.

Javin Paul
·

This tutorial is about **30 Eclipse keyboard shortcuts**, this list is by no means complete and I will suggest you guys share eclipse shortcuts listed other than here to make this more useful. Eclipse is the most used Java development IDE and knowing *Eclipse shortcuts* not only improve your productivity but also makes you more efficient. You will have more time for things you like to do. Using keyboard shortcuts also helps to impress colleagues and shows that you have a good hold on tools you used for Java Development. If you are a Java developer, who use Eclipse as your main IDE for all-purpose e.g. coding, development, and debugging, then I also suggest you take a look at Beginners Eclipse Java IDE Training Course, one of the best course to learn how to use Eclipse effectively from Java programmer's perspective.

I was a big fan of Netbeans IDE before coming across Eclipse because I was from the J2ME background and Netbeans provide sophisticated IDE environment to enable you to build, debug and run your Java application in various mobile emulator including some advanced processing options e.g. preprocessing, setting up Exception breakpoint etc.

From that time I love to know keyboard shortcuts on IDEs and other tools, I used for development including Edit Plus, Microsoft Excel etc. Knowing your tool better certainly helps you to deliver more in less time and find something really quick when a production issue arise.

As I said, I was a NetBeans fan at the start of my career but hen I joined an investment bank I came across Eclipse IDE, since everybody in my team is using and Netbeans IDE was not available in company (don't know why that was not allowed net-beans even though it was free) I have decided to go Eclipse way.

Later I found that Eclipse was most suited for that distributed core Java application which runs on Eclipse and depends upon a proprietary Linux library. It wasn't possible to run the whole application in Windows and that time some of the cool features of eclipse e. g. Remote Debugging, Conditional Breakpoints, Exception breakpoints, and Ctrl+T and Ctrl+R kind of shortcuts really saved my life.

Here I am sharing a **list of 30 Eclipse keyboard shortcut**s which I found most useful and used in my day to day life while writing code, reading the code or debugging Java application in Eclipse.

30 Useful Eclipse Keyboard Shortcuts for Java Programmers

Here is my list of 30 chosen Eclipse IDE keyboard shortcuts for Java developers. It's useful for both core Java developers and Java web application development using Eclipse IDE for web development.

1) **Ctrl + Shift + T** for finding class even from jar

This keyboard shortcut in Eclipse is my most used and favorite shortcut. While working with a high-speed trading system has a complex code, I often need to find classes with the just blink of the eye. This eclipse keyboard shortcut is made just for that. No matter whether you have class in your application or inside any JAR, this shortcut will find it.

2) **Ctrl + Shift + R** for finding any resource (file), including config XML files

This is similar to above Eclipse shortcut with the only difference being that it can find out not only Java files but any files including XML, configs, and many others. However, this eclipse shortcut only finds files from your workspace and doesn't dig at the JAR level.

3) **Ctrl + 1** for quick fix

This is another beautiful Eclipse shortcut that can fix up an error in Eclipse. Whether it's missing declaration, missing semi-colon, or any import related error, this eclipse shortcut will help you to quickly sort that out.

4) **Ctrl + Shift + o** for organize imports

This is another Eclipse keyboard shortcut for fixing missing imports. This is particularly helpful if you copy some code from another file and import all dependencies. You can see the Beginners Eclipse Java IDE Training Course to learn more about code formatting shortcuts in Eclipse.

Eclipse Shortcut for Quick Navigation

In this section, we will see some Eclipse keyboard shortcuts that help us to quickly navigate within the file and between the file while reading and writing code in Eclipse.

7) **Ctrl + o** for quick outline going quickly to a method

9) **Alt + right** and **Alt + left** for going back and forth while editing

12) **Alt + Shift + W** to show a `classin` package explorer

13) **Ctrl + Shift + Up** and down for navigating from member to member (variables and methods)

15) **Ctrl + k** and **Ctrl + Shift +K** for find next/previous

16) Go to a type declaration: **F3**, this Eclipse shortcut is very useful to see function definition very quickly.

Apart from these keyboard shortcuts, learning how to debug Java programs in Eclipse is also a skill. There are so many debugging tools and features available in Eclipse, which many Java programmer don't know about, e.g. conditional debugging, hit count, and remote debugging.

If you really want to become a good Java developer, I suggest you improve your debugging skills. If you need some help, Eclipse Debugging Techniques And Tricks is the best place to start with.
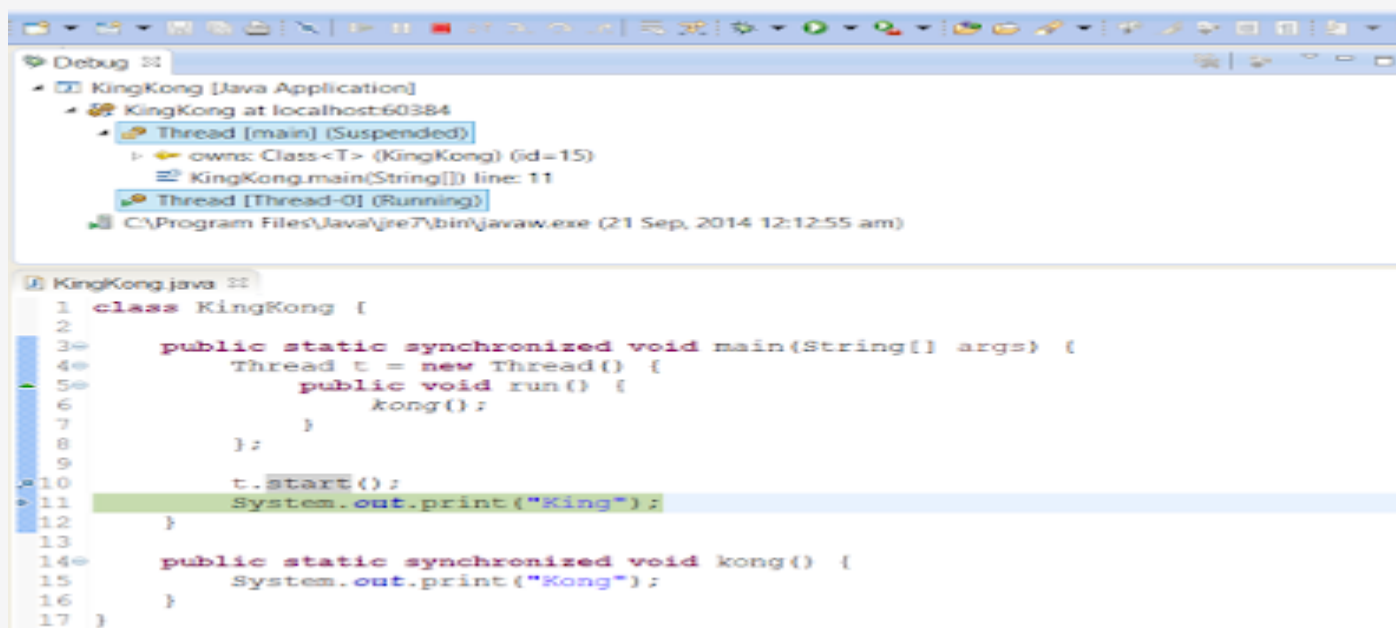
Eclipse Shortcut for Editing Code

These Eclipse shortcuts are very helpful for editing code in Eclipse:

5) **Ctrl + /** for commenting, uncommenting lines and blocks

6) **Ctrl + Shift + /** for commenting, uncommenting lines with block comment, see here for example.

8) Selecting class and pressing **F4** to see its Type hierarchy

10) **Ctrl + F4 or Ctrl + w** for closing current file

11) **Ctrl+Shirt+W for** closing all files.

14) **Ctrl + l** go to line

16) Select text and press **Ctrl + Shift + F** for formatting.

17) **Ctrl + F** for find, find/replace

18) **Ctrl + D** to delete a line

19) **Ctrl + Q** for going to last edited place

You can see this free course on Udemy called *Eclipse IDE for Beginners: Increase Your Java Productivity* to find live examples for most of these shortcuts.
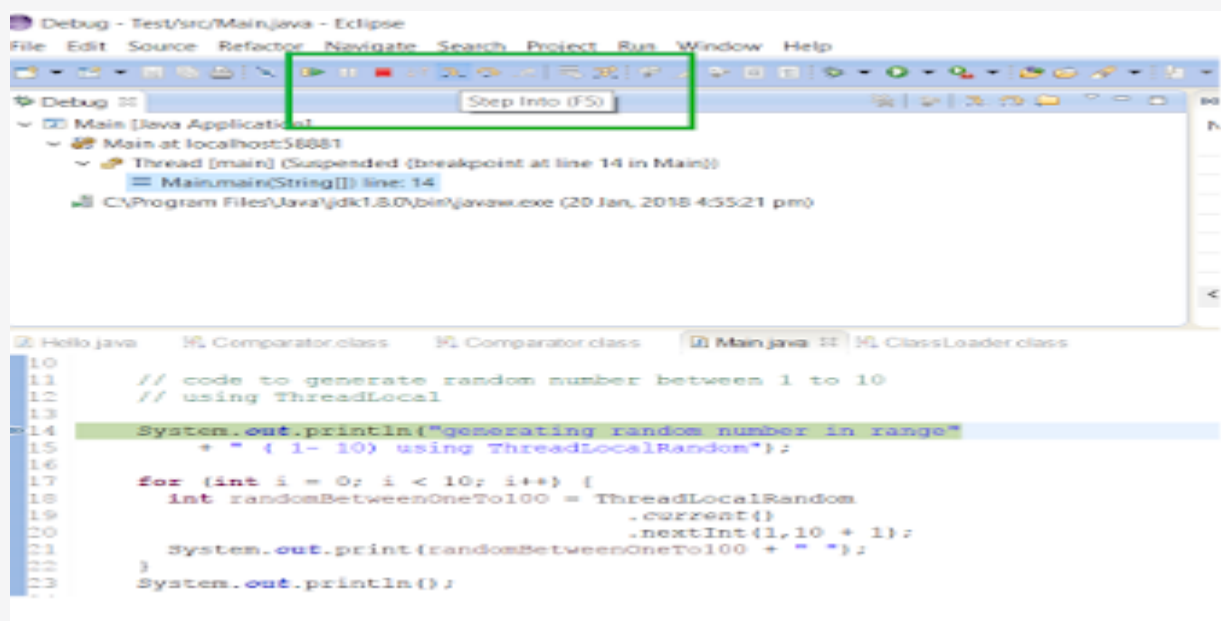

Miscellaneous Eclipse Shortcuts

These are different Eclipse keyboard shortcuts that don't fit on any category, but they can be quite helpful while working in Eclipse.

20) **Ctrl + T** for toggling between supertype and subtype

21) Go to other open editors: **Ctrl + E**.

22) Move to one problem (i.e.: error, warning) to the next (or previous) in a file: **Ctrl +.** For next, and Ctrl +, for the previous problem

23) Hop back and forth through the files you have visited: **Alt + ← and Alt + →,** respectively

25) **CTRL+Shift+G**, which searches the workspace for references to the selected method or variable

26) **Ctrl+Shift+L** to view the listing for all Eclipse keyboard shortcuts

27) **Alt + Shift + j** to add Javadoc at any place in Java source file

28) **CTRL+SHIFT+P** to find closing brace. Place the cursor at the opening brace and use this.

29) **Alt+Shift+X, Q** to run Ant build file using keyboard shortcuts in Eclipse

30) **Ctrl + Shift +F** for autoformatting

Remember, we are technical people. If we are not learning, we are going backwards.

And, here is the nice image to remember these **useful Eclipse shortcuts for Java programmers**:
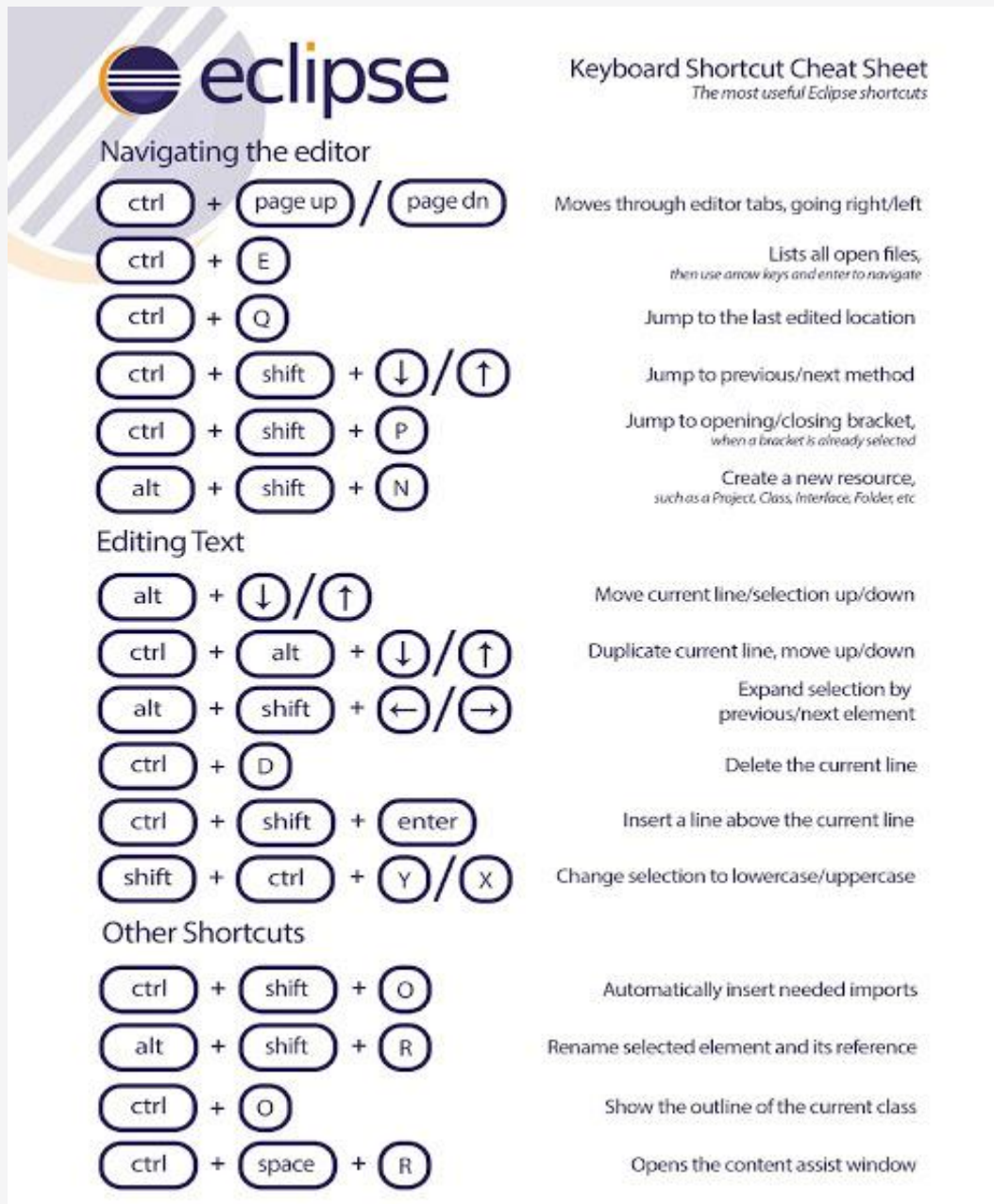
If you have some more useful **Eclipse keyboard shortcuts** then please post as comments, I will include them in this list. These Eclipse shortcuts will mostly work almost all Eclipse versions, e.g. 3.5, 3.6 Helios, Eclipse Ganymede and Indigo. Let me know if you face any issues while using these Eclipse shortcuts in any particular version of Eclipse IDE.

Further Learning

Related Eclipse tutorials for Java Developers

If you like this tutorial and wants to learn more productivity tips, tricks and new features of Eclipse IDE, then you can also see following guides and tutorials:

How to attach source code of a JAR file in Eclipse? (steps)
How to debug a Java Program in Eclipse IDE? (guide)
How to do the static import in Java Eclipse? (solution)
How to solve Eclipse No Java Virtual Machine found the error? (solution)
How to fix the JAR dependency not working in Eclipse Maven Plugin? (answer)
How to fix Eclipse Not able to connect Internet or Market Place error? (solution)
How do you create JUnit tests in Eclipse IDE? (guide)
Eclipse shortcut for System.out.println() statements (shortcut)
How to escape String literal in a Java file when posted on Eclipse? (solution)
How to compare two files in Eclipse IDE? (guide)
How to increase console buffer size in Eclipse? (solution)
How to make an executable JAR file in Eclipse? (steps)
How to fix Unsupported major.minor version 51.0 error in Eclipse? (solution)
How to fix must override superclass method error in Eclipse? (solution)
How to decompile class file in Eclipse? (solution)
How to set java.library.path in Eclipse IDE? (steps)

Download Building Reactive Microservices in Java: Asynchronous and Event-Based Application Design. Brought to you in partnership with Red Hat.

Java Partner Resources

Developing Reactive Microservices: Enterprise Implementation in Java
Lightbend
A Reference Guide to Stream Processing
Hazelcast
Build vs Buy a Data Quality Solution: Which is Best for You?
Melissa Data
Learn how to refactor a monolithic application to work your way toward a scalable and resilient microsystem.
Lightbend
Predictive Analytics + Big Data Quality: A Love Story
Melissa Data
Advanced Linux Commands [Cheat Sheet]
Red Hat Developer Program
Level up your code with a Pro IDE
JetBrains
Deep insight into your code with IntelliJ IDEA.
JetBrains
Microservices for Java Developers: A Hands-On Introduction to Frameworks & Containers
Red Hat Developer Program
Modern Java EE Design Patterns: Building Scalable Architecture for Sustainable Enterprise Development
Red Hat Developer Program
Designing Reactive Systems: The Role Of Actors In Distributed Architecture
Lightbend
Building Reactive Microservices in Java: Asynchronous and Event-Based Application Design
Red Hat Developer Program

Java Partner Resources

Developing Reactive Microservices: Enterprise Implementation in Java

Lightbend
A Reference Guide to Stream Processing
Hazelcast
Build vs Buy a Data Quality Solution: Which is Best for You?
Melissa Data
Learn how to refactor a monolithic application to work your way toward a scalable and resilient microsystem.
Lightbend