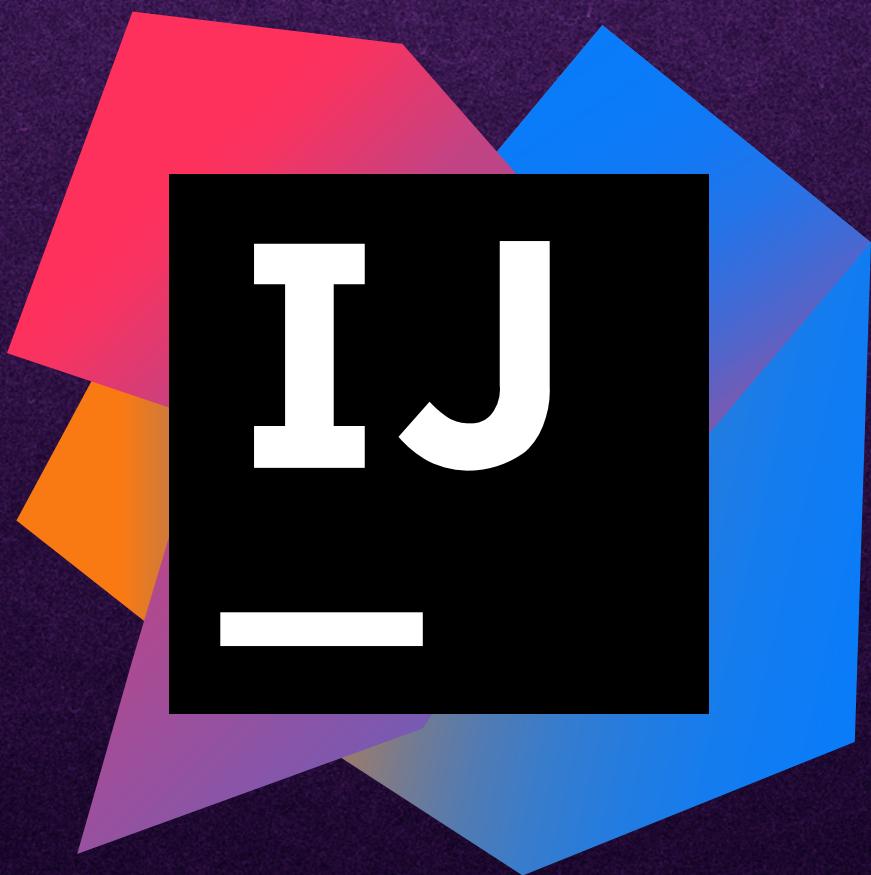


LESSER KNOWN FEATURES OF

IntelliJ Idea



SLIDE →



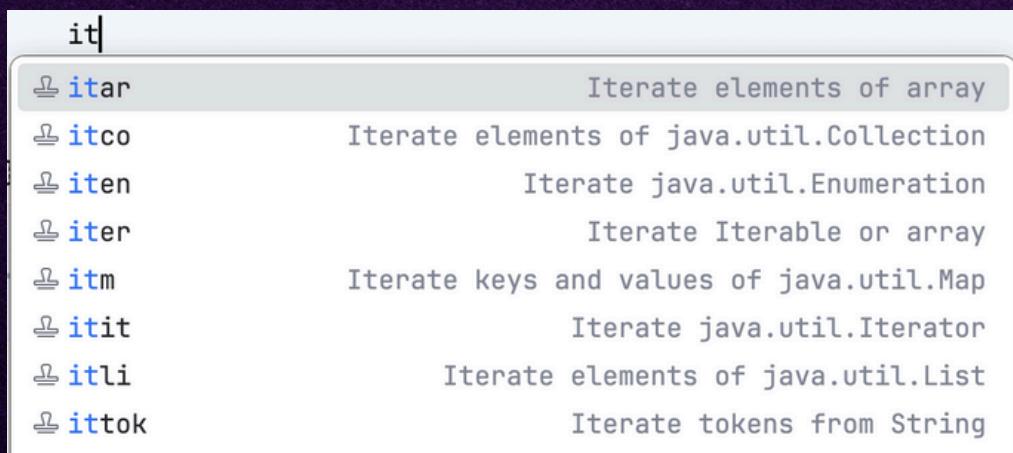
With ❤ by Mohit Kanwar

#CXO Incubator

LIVE TEMPLATES

Live Templates in IntelliJ IDEA are incredibly powerful for automating repetitive coding tasks and inserting commonly used code snippets quickly.

Just start typing in the right context to view the list of available templates



iten = while loop
iter = for loop
inn = if not null
psf = public static final

IntelliJ also provides a capability of creating custom live templates, that you can use to fasten your code development process.



With ❤ by Mohit Kanwar

SCRATCH FILES



The screenshot shows a code editor interface with a sidebar navigation. The sidebar lists 'External Libraries', 'Scratches and Consoles' (which is expanded), 'Extensions', and 'Scratches'. Under 'Scratches', there are several files: 'BreakParallelWork.java' (selected), 'scratch.java', 'scratch_2.java', 'scratch_3.java', 'scratch_4.java', 'scratch_5.java', and 'scratch_6.java'. The main editor area displays the Java code for 'BreakParallelWork.java':

```
1 import java.util.HashSet;
2 import java.util.Set;
3
4
5
6
7 public class BreakParallelWork {
8
9     public static void main(String[] args) {
10         Set<Integer> dataSet = new HashSet<Integer>();
11         dataSet.add(1);
12         dataSet.add(2);
13         dataSet.add(3);
14         dataSet.add(4);
15         dataSet.add(5);
16 }
```

Quickly want to test a code behaviour without impacting your project?

Want to manage some POCs?

Want to try a code in another language as your current project?

Minimize all files, and reach the bottom of the files navigator. Start enjoying the scratch files feature.



With ❤️ by Mohit Kanwar

CONDITIONAL DEBUG POINTS

A screenshot of an IDE interface, likely IntelliJ IDEA, showing a Java code editor. The code is a simple 'Scratch' class with a main method that prints integers from 0 to 50,000,000. A conditional breakpoint is set on the third line of the main method. A tooltip window titled 'Line and lambdas breakpoint' is open over the breakpoint, containing the following settings:

- Enabled
- Suspend: All Thread
- Condition:

The 'Done' button is at the bottom right of the tooltip.

The code runs until the condition of $i == 29000$ is met. At that point, you can start your debugging.



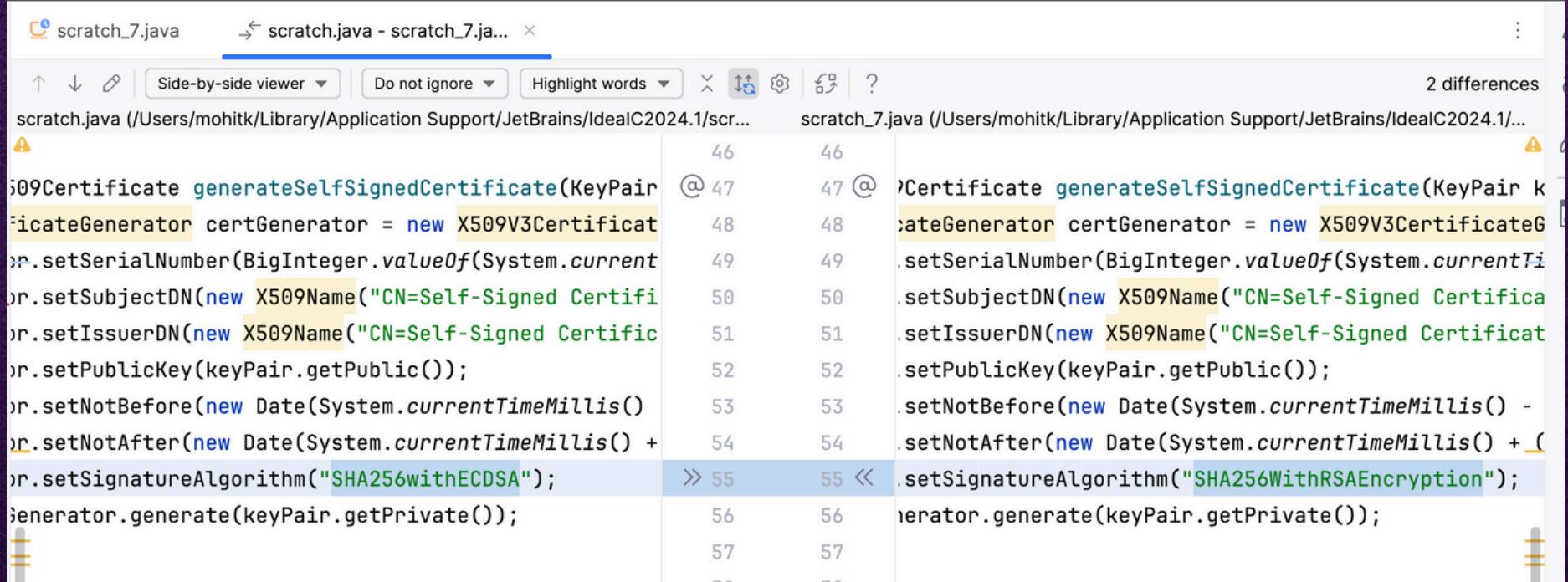
Want to take control of code execution while debugging under specific conditions?

Add a conditional debug point!



With ❤ by Mohit Kanwar

COMPARE FILES



The screenshot shows a comparison between two Java files in IntelliJ IDEA. The left file is 'scratch.java' and the right file is 'scratch_7.java'. The code compares the generation of a self-signed certificate. In the first file, the signature algorithm is 'SHA256withECDSA'. In the second file, it is changed to 'SHA256WithRSAEncryption'. Other changes include the addition of a 'setNotAfter' date and the removal of a 'setNotBefore' date.

```
scratch.java (/Users/mohitk/Library/Application Support/JetBrains/IdealC2024.1/scr... scratch_7.java (/Users/mohitk/Library/Application Support/JetBrains/IdealC2024.1/...
46   46
47 @ 47 @ ?Certificate generateSelfSignedCertificate(KeyPair k
48   48  ateGenerator certGenerator = new X509V3Certificat
49   49 .setSerialNumber(BigInteger.valueOf(System.currentTi
50   50 .setSubjectDN(new X509Name("CN=Self-Signed Certifi
51   51 .setIssuerDN(new X509Name("CN=Self-Signed Certific
52   52 .setPublicKey(keyPair.getPublic());
53   53 .setNotBefore(new Date(System.currentTimeMillis() - 
54   54 .setNotAfter(new Date(System.currentTimeMillis() + (
55 >> 55 << .setSignatureAlgorithm("SHA256WithRSAEncryption");
56   56 erator.generate(keyPair.getPrivate());
57   57
```

Comparing two seemingly similar files? IntelliJ highlights the differences for you.

This is really helpful specially when used with another feature that IntelliJ provides - Git conflict resolver.



With ❤ by Mohit Kanwar

WORKING WITH LONG FILES

```
14  class SecureEncryptionExample2 {  
66      public static byte[] encrypt(String message) throws Exception {  
70  
71          er = Cipher.getInstance(transformation: "RSA");  
72          (Cipher.ENCRYPT_MODE, publicKey);  
73          er.doFinal(message.getBytes(StandardCharsets.UTF_8));  
74  
75
```

**When working with long files,
IntelliJ pins the class name and
method names on top , Giving you
the context, where you are!**



With ❤️ by Mohit Kanwar

Hi, I'm Mohit Kanwar

Architect | Xebia

**I help Fintechs and Banks develop
their software solutions**

