



# IntelliJ IDEA Dojo

Daniel Hinojosa

Mastodon: [@dhinojosa@mastodon.social](https://@dhinojosa@mastodon.social)

Email: [dhinojosa@evolutionnext.com](mailto:dhinojosa@evolutionnext.com)



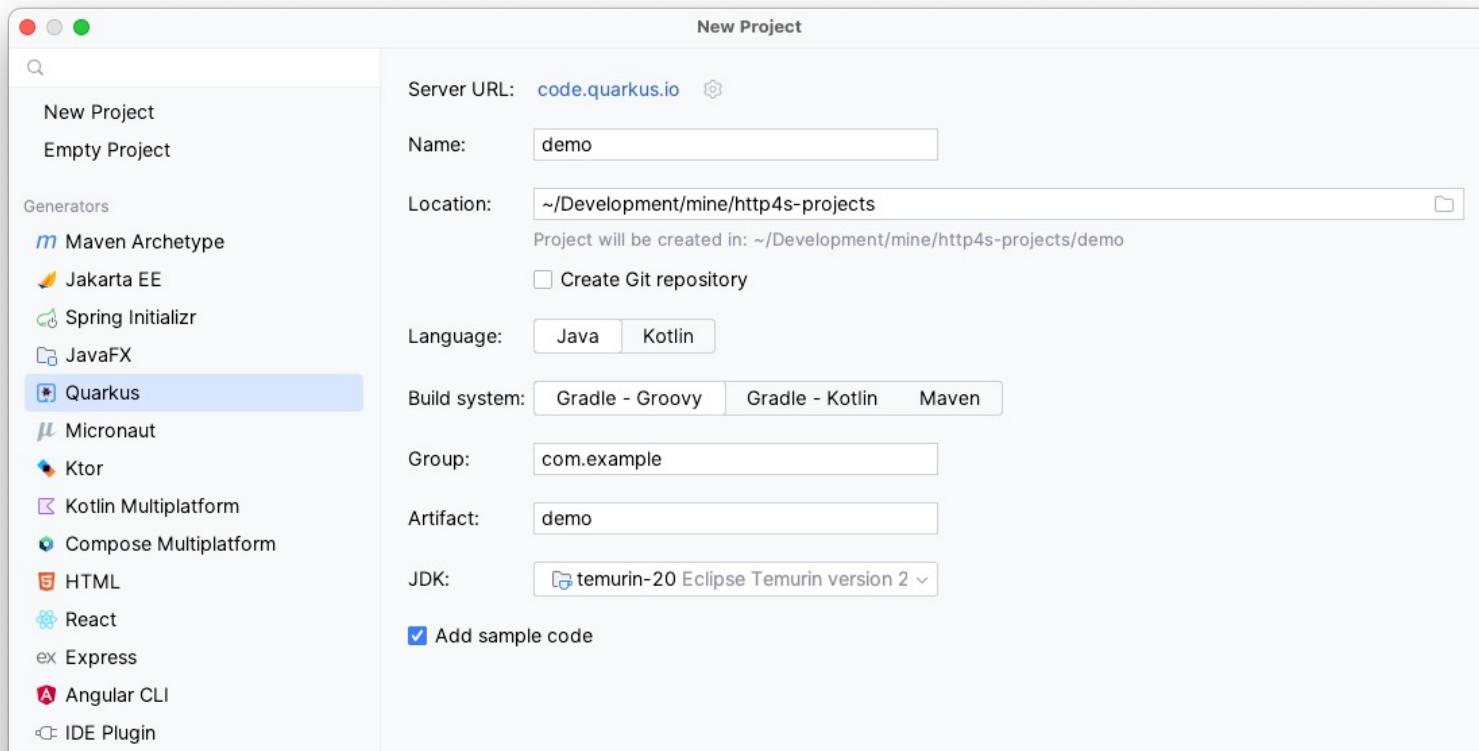


# New Projects



# New Projects

- Typically, projects are created outside the IDE, which a build tool. i.e., Maven, Gradle, etc.
- IntelliJ offers some good choices without resorting to “IntelliJ-only” type projects





# Opening Projects



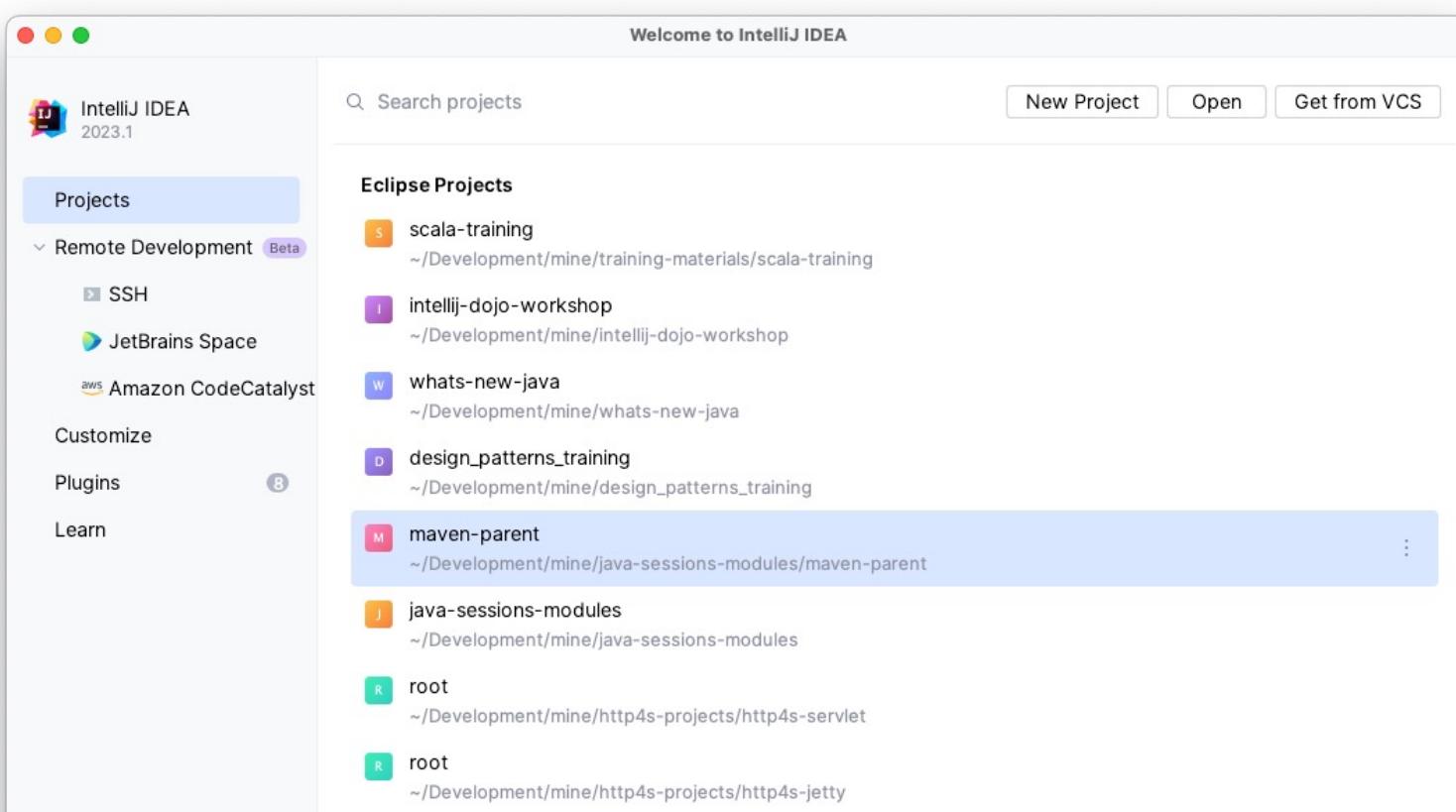
# Opening Projects

- Typically, projects are created outside the IDE, which a build tool. i.e., Maven, Gradle, etc.
- Therefore, really all that needs to be done is open the project.
- This can be done with **[File > Open]** or **[File > Recent Projects]**



# Opening Projects

- When you start IntelliJ for the first time, or when you close out of currently open projects, you can open your projects in the Welcome to IntelliJ window.





Command Line



# Opening Projects using the Command Line

- If you want to open a project into IntelliJ from command line you can do so from command line by typing `idea .` After some setup...



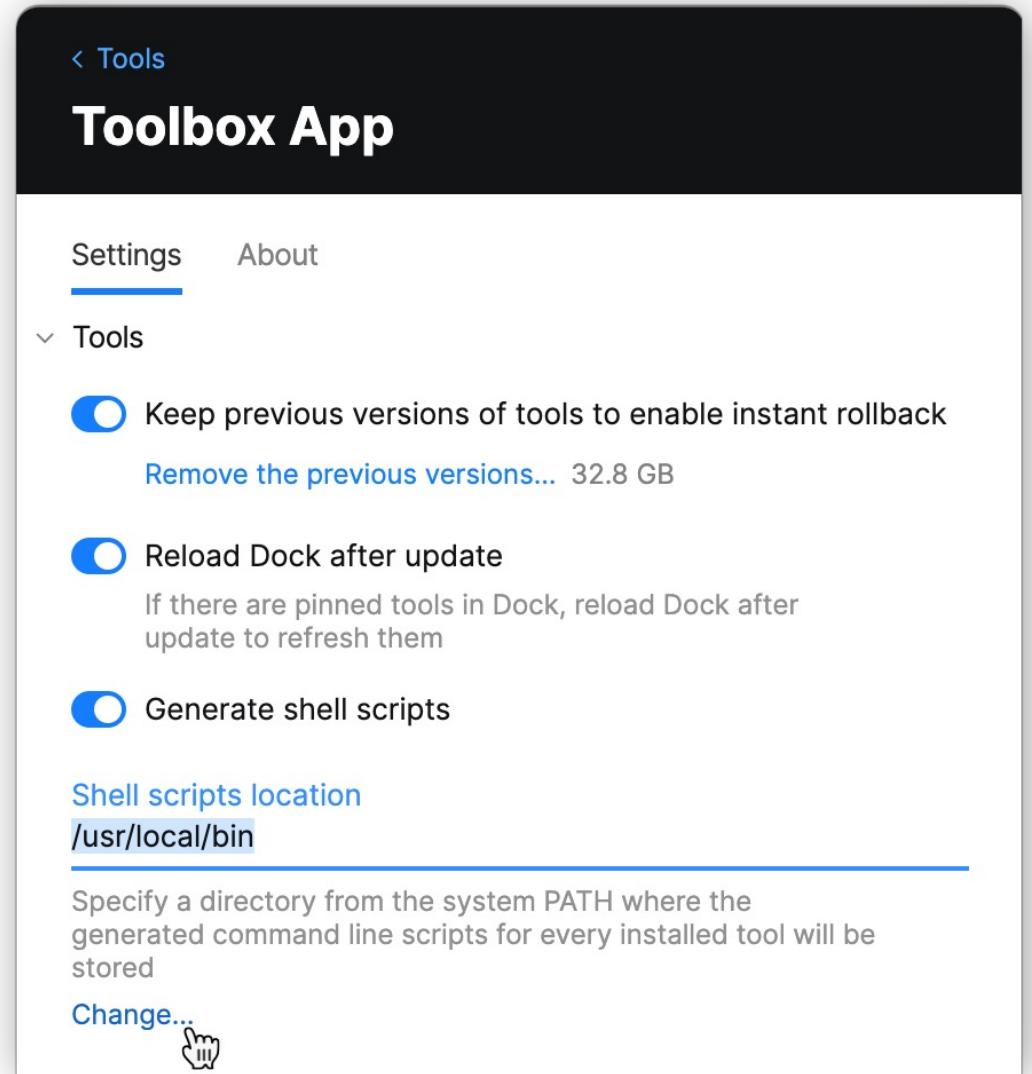
```
danno@MacBook-Pro:~/Development/mine/java-tdd
(base) -----  
~/Development/mine/java-tdd (answers*) » idea .■  
danno@MacBook-Pro
```

A screenshot of a macOS terminal window. The window title is "danno@MacBook-Pro:~/Development/mine/java-tdd". The prompt "(base) -----" is followed by the command " ~/Development/mine/java-tdd (answers\*) » idea .■". The cursor is positioned at the end of the command "idea .". The background of the terminal is light yellow.



# Generating Scripts

- In your JetBrains toolbox, click on the bolt. 
- Click on **Settings**
- Toggle on “Generate Shell Scripts” and either keep or override the location of the scripts
- For Windows users, put this location on your PATH



The screenshot shows the "Toolbox App" settings interface. At the top, there's a back arrow labeled "< Tools" and the title "Toolbox App". Below the title are two tabs: "Settings" (which is underlined) and "About". Under the "Tools" section, there are three toggle switches:

- Keep previous versions of tools to enable instant rollback  
[Remove the previous versions...](#) 32.8 GB
- Reload Dock after update  
If there are pinned tools in Dock, reload Dock after update to refresh them
- Generate shell scripts

A "Shell scripts location" field is set to `/usr/local/bin`. A note below it says: "Specify a directory from the system PATH where the generated command line scripts for every installed tool will be stored". At the bottom right of the field is a "Change..." button with a hand cursor icon pointing at it.



Tool Windows



# Tool Windows Navigation



**ALT-0...ALT-9**

Open Corresponding Tool Window

**CTRL-E**

View Recent Files, Includes Tool Windows

**CTRL-SHIFT+E**

View Recent Files, Includes Tool Windows

**ESC**

Return to the Editor

**CTRL+SHIFT+F12**

Maximize the Editor

**SHIFT+ESCAPE**

Close Current Tool Window

**CTRL+ALT+SHIFT+LEFT/ CTRL+ALT+SHIFT+RIGHT**

Shrink or Expand Tool Window



# Tool Windows Navigation



COMMAND(⌘) + 0...COMMAND(⌘) + 9

Open Corresponding Tool Window

COMMAND(⌘)-E

View Recent Files, Includes Tool Windows

COMMAND(⌘)-SHIFT+E

View Recent Edited Locations

ESCAPE(⊟)

Return to the Editor

COMMAND(⌘)+SHIFT(⇧)+F12

Maximize the Editor

SHIFT(⇧)+ESCAPE(⊟)

Close Current Tool Window

CONTROL(^) + OPTION(⌥) + LEFT(←)/

CONTROL(^) + OPTION(⌥) + RIGHT(→ )

Shrink or Expand Tool Window



Find Action



# Action Keys



**CTRL-SHIFT-A** Find Action

**CTRL-CTRL** Run Anything

**SHIFT-SHIFT** Search Everywhere



# Action Keys



SHIFT(⬆)+COMMAND(⌘) + A Find Action  
CONTROL(^) + CONTROL(^) Run Anything  
SHIFT(⬆)+ SHIFT(⬆) Search Everywhere



# Tab Management



# Tab Management Keys



**ALT+RIGHT/LEFT**

Go to Next or Previous Tab

**CTRL+F4**

Close Current Tab



# Tab Management Keys



**CONTROL(^) + RIGHT(→)/LEFT(←)**

Go to Next or Previous Tab

**COMMAND(⌘) + W**

Close Current Tab

**COMMAND(⌘) + SHIFT(⇧) + W**

Close All Tabs



# Dojo: Tool Windows



Finding Stuff



# Finding Items



**CTRL-N** Go to class

**CTRL-SHIFT-N** Go to file

**CTRL-ALT-SHIFT-N** Go to symbol



# Finding Items



COMMAND(⌘) + O Find Class

COMMAND(⌘) + SHIFT(⇧) + O Go to File

COMMAND(⌘) + OPTION(⌥) + O Go to Symbol



# Dojo: Finding Items



# Navigating Errors



# Navigating Errors



F2

Move to the Next Highlighted Issue

ALT+ENTER

Propose a Fix



# Navigating Errors



F2

Move to the Next Highlighted Issue

**OPTION(⌥) + RETURN(↵)**

Propose a Fix



# Dojo: Finding Errors



# Generating Files & Code



# Generating Files and Code



ALT+INSERT Generate File or Code



# Generating Files and Code



COMMAND(⌘) + N or  
CONTROL(^) + RETURN(↵)  
Generate File or Code



# Formatting Code



# Formatting Code



**CTRL + ALT + L**

Format All Code (includes imports)

**CTRL + ALT + O**

Optimize Imports

**CTRL + ALT + I**

Format Line by Line



# Formatting Code



**COMMAND(⌘) + OPTION(⌥) + L**

Format All Code (includes imports)

**CONTROL(^) + OPTION(⌥) + O**

Optimize Imports

**CONTROL(^) + OPTION(⌥) + I**

Format Line by Line



# Dojo: Generating Code



Structure



# File Structure



**ALT + 7**

View the Structure of the Class

**CTRL+F12**

File Structure Popup



# File Structure



**COMMAND(⌘) + 7**

View the Structure of the Class

**COMMAND(⌘) + F12**

File Structure Popup



The Blob



IRVIN S. TEAWORTH, JR.'s  
**THE BLOB**  
INDESCRIBABLE...  
INDESTRUCTIBLE...  
NOTHING CAN STOP IT!  
TONY LYN presents  
**THE BLOB**  
starring  
**STEVE MCQUEEN,**  
**ANETA CORSAUT, STEPHEN CHASE,**  
**EARL ROWE, OLIN HOWLIN,**  
**JOHN BENSON & GEORGE KARAS**

12



# Selecting Code Blocks



CTRL + W Increase Code Block

CTRL + SHIFT + W Decrease Code Block



# Selecting Code Blocks



OPTION(⌥) + UP(↑) Increase Code Block

OPTION(⌥) + DOWN(↓) Decrease Code Block



# Editing Basics



# Editing Basics



**CTRL+Y** Delete Line at Caret

**CTRL+D** Duplicate Line at Caret

**CTRL+X** Cut Line at Caret

**CTRL+C** Copy Line at Caret

**CTRL+V** Paste Line at Caret

**CTRL+SHIFT+V** Paste From History



# Editing Basics



**COMMAND(⌘)+DELETE(⌫)**

Delete Line at Caret

**COMMAND(⌘)+D**

Duplicate Line at Caret

**COMMAND(⌘)+X**

Cut Line at Caret

**COMMAND(⌘)+C**

Copy Line at Caret

**COMMAND(⌘)+V**

Paste Line at Caret

**COMMAND(⌘)+SHIFT(⇧)+V**

Paste From History



# Prefer Tab over Enter

```
@Test  
void testMath() {  
    var result = Math.abs(30);  
}
```

Tab or Enter ?

```
new *  
  
@Test  
void testMath() {  
    var result = Math.dabs(30);  
}
```

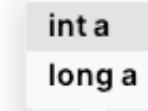
⑥ **decrementExact**(int a)  
⑥ **decrementExact**(long a)

int  
long



# What happens with Enter

```
new *  
@Test  
void testMath() {  
    var result = Math.decrementExact(Math.abs(30));  
}
```



A callout bubble is positioned above the word 'abs'. It contains two options: 'int a' and 'long a'. The word 'abs' is underlined with a red wavy line, indicating it is being considered for conversion.



# What happens with Tab

```
new *  
@Test  
void testMath() {  
    var result = Math.decrementExact(30);  
}
```



# Surround Blocks and Selections



# Surround Blocks and Selections



**CTRL + ALT + T** Surround Block or Selection



# Surround Blocks and Selections



COMMAND(⌘) + OPTION(⌥) + T

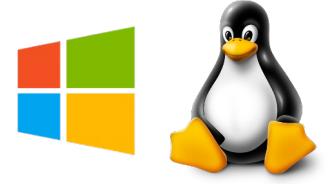
Surround Block or Selection



# Dojo: Using Blobs



Usages



**CTRL + ALT + B** Go to Implementation

**CTRL + B** Go to Declaration

**ALT + F7** Find Usages

**CTRL + F7** Find in Files

**CTRL + ALT + LEFT / RIGHT**

  Navigate back / forward



# Usages



**COMMAND(⌘) + OPTION(⌥) + B**

Go to Implementation

**COMMAND(⌘) + B**

Go to Declaration

**OPTION (⌥) + F7**

Find Usages

**COMMAND(⌘) + SHIFT(⇧) + F7**

Find In Files

**COMMAND(⌘) + OPTION(⌥) + LEFT(←) /  
RIGHT(→)**

Navigate back / forward



# Route "Find in Files" to Find Window

Find in Files 21 matches in 1 file  File mask: \*.**java** Cc W .\*

In Project Module Directory Scope

System.out.println(ansi())		Game.java 25
System.out.println(ansi())		Game.java 32
System.out.println(ansi().reset());		Game.java 42
System.out.println("You need to [H]it or [S]tand");		Game.java 75
System.out.println("You Busted, so you lose. " );		Game.java 89

Game.java src/main/java/com/jitterted/ebp/blackjack

```
22     Game game = new Game();
23
24     AnsiConsole.systemInstall();
25     System.out.println(ansi()
26             .bgBright(Ansi.Color.WH
27             .eraseScreen()
28             .cursor(1, 1)
29             .fgGreen().a("Welcome to
30             .fgRed().a(" JitterTed'")
```

Open results in new tab Open in Find Window



# Dojo: Usages



# Code Completion



# Code Completion



**CTRL + SPACE**

Basic Code Completion

**CTRL + ALT + SPACE**

Code Completion that shows inaccessible members

**CTRL + SHIFT + SPACE**

Type Sensitive Code Completion

**CTRL + SHIFT + ENTER**

Complete Statement

**CTRL + P**

Parameter Info

**CTRL + SHIFT+P**

Type Info



# Code Completion



**CONTROL(^) + SPACE**

Basic Code Completion

**CONTROL(^) + OPTION (⌥) + SPACE**

Code Completion that shows inaccessible members

**CONTROL(^) + SHIFT(⇧) + SPACE**

Type Sensitive Code Completion

**COMMAND(⌘) + SHIFT(⇧) + RETURN(⏎)**

Complete Statement

**COMMAND(⌘) + P**

Parameter Info

**CONTROL(^) + SHIFT(⇧) + P**

Type Info



Moving Code



# Moving Code



**CTRL + SHIFT + UP/DOWN**

Move either line or method, depending on caret location



# Moving Code



**SHIFT (⬆) + COMMAND(⌘) + UP(↑)/DOWN(↓)**

Move either line or method, depending on  
caret location



Dojo: Moving Code



Code Documentation  
& Help



# Code Documentation and Help



**CTRL + Q**

Quick Documentation Lookup

**CTRL + SHIFT + I**

Quick Definition Lookup

**ALT+Q**

Context Info

**CONTROL(^) + SHIFT (^) + P**

Type Info (Twice to show more)



# Code Documentation and Help



**CONTROL(^) + J**

Quick Documentation

**OPTION(⌥) + SPACE or COMMAND (⌘) + Y**

Quick Definition Lookup

**CONTROL(^) + SHIFT (⇧) + Q**

Context Info

**CONTROL(^) + SHIFT (⇧) + P**

Type Info (Twice to show more)



# Dojo: Documentation and Info



Running



# Running



All runs are contextual, this includes main methods, tests, test folders, HTTP files, and more.

**CTRL + SHIFT + F10**

Run and mark your context configuration

**SHIFT + F10**

Run the last context configuration

**ALT+SHIFT+F10**

Select context configuration and run



# Running



All runs are contextual, this includes main methods, tests, test folders, HTTP files, and more.

**CONTROL(^) + SHIFT (⇧) + R**

Run and mark your context configuration

**CONTROL(^) + R**

Run the last context configuration

**CONTROL(^) + OPTION(⌥) + R**

Select context configuration and run



Testing



**CTRL + SHIFT + T**

Toggle Between Test and Production



Testing



COMMAND(⌘) + SHIFT (⇧) + T

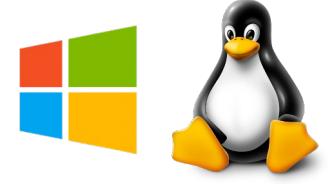
Toggle Between Test and Production



# Dojo: Running and Testing



Debugging



**CTRL+F8**

Toggle Breakpoint

**CTRL + SHIFT+ F8**

View Breakpoints

**CTRL+SHIFT+F9**

Run Context Configuration Debug

**SHIFT+F9**

Debug the Last Context Configuration

**ALT+SHIFT+F9**

Select Context Configuration and Debug



# Debugging



**COMMAND(⌘) + F8**

Toggle Breakpoint

**COMMAND(⌘) + SHIFT (⇧) + F8**

View Breakpoints

**CONTROL(^) + SHIFT (⇧) + D**

Debug and Mark your Context Configuration

**CONTROL(^) + D**

Run Previous Test Selection

**CONTROL(^) + OPTION(⌥) + R**

Select Context Configuration and Debug



# Dojo: Debugging



# Refactoring



# Refactoring (Part 1)



**CTRL + ALT+ SHIFT + T**

Refactor This...

**F5**

Copy item

**F6**

Move item

**SHIFT+F6**

Rename globally

**ALT+DELETE**

Safe Delete



# Refactoring (Part 2)



**Ctrl + Alt + N** Inline

**Ctrl + Alt + M** Extract Method

**Ctrl + Alt + V** Extract Variable

**Ctrl + Alt + F** Extract Field

**Ctrl + Alt + C** Extract Constant

**Ctrl + Alt + P** Extract Parameter



# Refactoring (Part 1)



**CONTROL(^) + T**

Refactor This

**F5**

Copy item

**F6**

Move item

**SHIFT(⇧)+F6**

Rename globally

**COMMAND(⌘)+DELETE**

Safe Delete



# Refactoring (Part 2)



COMMAND(⌘) + OPTION(⌥) + N

Inline

COMMAND(⌘) + OPTION(⌥) + M

Extract Method

COMMAND(⌘) + OPTION(⌥) + V

Extract Variable



# Refactoring (Part 3)



COMMAND(⌘) + OPTION(⌥) + F

Extract Field

COMMAND(⌘) + OPTION(⌥) + C

Extract Constant

COMMAND(⌘) + OPTION(⌥) + P

Extract Parameter



Dojo: Refactoring



HTML/XML Editing



# HTML/XML Editing

<http://docs.emmet.io/abbreviations/syntax/>

<https://docs.emmet.io/cheat-sheet/>



Dojo: Hey Emmet



# Version Control



# Version Control



**ALT + 0**

Version Control Window

**CTRL + K**

Commit

**CTRL+T**

Update Project

**CTRL + SHIFT + K**

Push

**ALT + ` (Backquote)**

Version Control Quick Help



# Version Control



**COMMAND(⌘) + 0**

Version Control Window

**COMMAND(⌘) + K**

Commit

**COMMAND(⌘) + T**

Update Project

**COMMAND(⌘) + SHIFT(⇧) + K**

Push

**CONTROL(^) + V**

Version Control Quick Help



# Version Control Conflicts



# Version Control Conflicts

Merge Revisions for /Users/danno/tmp/beginning-java\_2023-05-30/src/main/java/com/salesforce/Person.java

3 changes. 5 conflicts.

```
Result
```

Your version

```
import java.util.Optional;
import java.util.StringJoiner;
import java.util.function.Supplier;

public class Person {
    private final String firstName;
    private final String lastName;
    private final String middleName;
    private final LocalDate birthLocalDate;
    private final Supplier<LocalDate> todaysLocalDate;

    protected Person(String firstName, String middleName,
                     String lastName, LocalDate birthLocalDate,
                     Supplier<LocalDate> todaysLocalDate) {
        this.firstName = firstName;
        this.middleName = middleName;
        this.lastName = lastName;
        this.birthLocalDate = birthLocalDate;
        this.todaysLocalDate = todaysLocalDate;
    }

    public static Person withNoMiddleName(String firstName,
                                         String lastName,
                                         LocalDate birthLocalDate) {
        return new Person(firstName, null, lastName, birthLocalDate);
    }

    public static Person withMiddleName(String firstName, String middleName,
                                       LocalDate birthDay) {
        return new Person(firstName, middleName, lastName, birthDay);
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String getFullName() {
        if (middleName == null) {
            return String.format("%s %s", firstName, lastName);
        } else {
            return String.format("%s %s %s", firstName, middleName, lastName);
        }
    }
}
```

Changes from server (revision cb0e7b28fb39483ca282a29a799003a19a57251)

```
import java.util.Objects;
import java.util.Optional;
import java.util.StringJoiner;
import java.util.function.Supplier;

public class Person {
    private String firstName;
    private String lastName;
    private String middleName;
    private LocalDate birthDate;
    private final Supplier<LocalDate> todaysDate;

    protected Person(String firstName, String middleName,
                     String lastName, LocalDate birthDate,
                     Supplier<LocalDate> todaysDate) {
        this.firstName = firstName;
        this.middleName = middleName;
        this.lastName = lastName;
        this.birthDate = birthDate;
        this.todaysDate = todaysDate;
    }

    public static Person withNoMiddleName(String firstName,
                                         String lastName,
                                         LocalDate birthDate) {
        return new Person(firstName, null, lastName, birthDate);
    }

    public static Person withMiddleName(String firstName, String middleName,
                                       LocalDate birthdate) {
        return new Person(firstName, middleName, lastName, birthdate);
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public Optional<String> getMiddleName() {
        return Optional.ofNullable(middleName);
    }

    public int getAge() {
        return (int) ChronoUnit.YEARS.between(birthDate, todaysDate);
    }
}
```

Accept Left   Accept Right   Cancel   Apply



# Accept or Merge

Conflicts

Merging branch **Discover\_IDEA** into branch **check\_lost\_content**

Name	Yours (check...)	Theirs (Discov...	
current.help.version ~/IdeaProjects/help-sources	Modified	Modified	<b>Accept Yours</b>
Discover_IntelliJ_IDEA.xml ~/IdeaProjects/help-sources/intellij-pl	Modified	Modified	<b>Accept Theirs</b>
ij.tree ~/IdeaProjects/help-sources/intellij-platform	Modified	Modified	<b>Merge...</b>
Manage_branches.xml ~/IdeaProjects/help-sources/intellij-platfc	Modified	Modified	
project.ihp ~/IdeaProjects/help-sources/intellij-platform	Modified	Modified	

Group files by directory

**Close**



# Color Meanings

Merge Revisions for /Users/anna.gasparyan/IdeaProjects/help-sources/intellij-platform/topics/Big\_Data\_Tools\_Configuration.xml

8 changes. 1 conflict.

Deleted line

Modified line

Newly added line

Conflicting changes

Result		Changes from server (revision ad635f2ba...)	
Mandatory parameters:</p>	49	<p>Mandatory parameters:</p>	49
<list>	50	<list>	50
<li><control>URL</c	51	<li><control>URL</cont	51
<li><control>Login<	52	<li><control>Login</co	52
</list>	53	</list>	53
<p> Optionally, you can s	53	<p> Optionally, you can set	54
<list>	55	<list>	55
<li><control>Name</c	56	<li><control>Name</cont	56
<li><control>Login a	57	<li><control>Login as a	57
<li><control>Enable	58	<li><control>Per projec	58
By default, all	59	Deselect it if you	59
<li><control>Per pro	60	<li><control>Enable con	60
Deselect it if y	61	By default, the new	61
<li><control>Scala V	62	<li><control>Scala Vers	62
plugin bundles.	63	plugin bundles. If	63
<li><control>Enable	64	<li><control>Enable HTT	64
connection with	65	connection with the	65

Accept Left    Accept Right    Cancel    Apply



# Accepting Conflicts

The screenshot shows two code editors side-by-side, likely from an IDE like IntelliJ IDEA, illustrating the concept of accepting conflicts during refactoring.

**Left Editor:**

```
    45    46    47    48    49    50    51    52    53    54    55    56    57    58    59
  public re} } public re} } public re} } @Override public
```

**Right Editor:**

```
days, t hDay, LocalDa 32 33 34 String middl 35 @ , birthDay, l 36 ✕✗ Accept 37 38 39 40 41 42
```

In both editors, there are conflict markers (red highlights and crossed-out code) and "Accept" buttons. The "Accept" button in the right editor has a small icon of a person with a checkmark. The code appears to be part of a Java class definition, involving methods for setting birth day and local date, and possibly overriding another method.



# Ignoring Conflicts

A screenshot of a code editor showing a conflict between two 'public' declarations. The code is as follows:

```
45    46    47    48    49    pub
, lastName);
ame, middleName,
        50    51    52    53    54    pub
        55    56    57    58    59    @Ov
        59 ⏵    pub
```

The line 'ame, middleName,' is highlighted in green. The line ' 55' is highlighted in red. A tooltip box labeled 'Ignore' is positioned over the green highlight.

A screenshot of a code editor showing a conflict between 'sDate.getDate()' and 'else;' statements. The code is as follows:

```
47
48
49
50
51
52
53
54
55
56
57
58
59
```

The line 'sDate.getDate()' is highlighted in red. A tooltip box labeled 'Ignore' and 'Ctrl+click to resolve conflict' is positioned over the red highlight.



# Version Control Conflicts



F7

Next Difference

SHIFT+F7

Previous Difference



# Version Control



F7

Next Difference

SHIFT(↑) + F7

Previous Difference



**Column Selection Mode**



# Column Selection Mode



**ALT+SHIFT+INSERT** Column Selection Mode



# Column Selection Mode



SHIFT(⬆)+COMMAND(⌘) + 8

Column Selection Mode



Multicursors



# Multicursors



**ALT+SHIFT+ [Click]**

Start Multicursors

Delete Single Multicursor

Note: Can be used in selections (blobs) as well

**ALT+J**

Select Next Word Occurrence

**ALT+SHIFT+J**

Select All Word Occurrences

Can be used with Find...

**ALT + SHIFT + G**

Add Carets At The End of Lines in Selected Block



# Multicursors



**OPTION(⌥) + SHIFT(⇧) + [Click]**

Start Multicursors

Delete Single Multicursor

Note: Can be used with selections (blobs) as well

**CONTROL(^) + G**

Select Next Word Occurrence

**CONTROL(^) + COMMAND(⌘) + G**

Select All Word Occurrences

Can be used with Find...

**OPTION(⌥) + SHIFT(⇧) + G**

Add Carets At The End of Lines in Selected Block



# Hierarchies



# Hierarchies



**CTRL+H**

Type Hierarchy - Show Parent and Child Classes

**CTRL+SHIFT+H**

Method Hierarchy – Show subclasses where the method overrides the selected method as well as what is overridden

**CTRL+ALT+H**

Call Hierarchy – Callers (Supertypes) or Callees (Subtypes) of a method, if on a field it will show the methods that interact with it



# Hierarchies



**CONTROL(^) + H**

Type Hierarchy - Show Parent and Child Classes

**CONTROL(^) + SHIFT(↑) + H**

Method Hierarchy – Show subclasses where the method overrides the selected method as well as what is overridden

**CONTROL(^) + OPTION(⌥) + H**

Call Hierarchy – Callers (Supertypes) or Callees (Subtypes) of a method, if on a field it will show the methods that interact with it



Presentation Mode



# Build Tools Support



# Build Tools Support



- Maven
- Gradle
- SBT
- Leiningen
- Rake
- Buildr
- Ant
- Gant



# Live Templates



# Live Templates



- Use live templates to insert common constructs into your code, such as
  - Loops,
  - Conditions,
  - Various declarations
  - Print statements



# Dojo: Live Templates



# Alternative Languages



# Alternate Languages



- Kotlin
- Scala
- Groovy
- Ruby/JRuby
- Python
- JavaScript
- Haskell
- Other Languages



Audience Favorites

# Thanks!

Email: dhinojosa@evolutionnext.com  
Mastodon: <https://mastodon.social/@dhinojosa>