

THE DEVELOPER'S HANDBOOK to Android Studio

⚙ Settings...

📁 Project Structure...

Ctrl

• File Properties •

Local History

💾 Save All

🔄 Sync Project with Gradle Files

Refresh Linked C++ Projects

🔄 Reload All from Disk

Repair IDE

Invalidate Caches...

Manage IDE Settings

New Projects Setup

Export

🖨 Print...

Power Save Mode

Exit by Saurabh Kumar

created by saurabh
SAURABH

****Android Studio**** गूगल (Google) द्वारा विकसित किया गया एक ****Official Integrated Development Environment (IDE)**** है। इसका मुख्य उपयोग एंड्रॉयड ऐप्स (Android Applications) बनाने, उन्हें टेस्ट करने और डिबग (Debug) करने के लिए किया जाता है।

अगर आसान शब्दों में कहें, तो जैसे खाना बनाने के लिए किचन और जरूरी बर्तनों की जरूरत होती है, वैसे ही एक शानदार एंड्रॉयड ऐप बनाने के लिए Android Studio की जरूरत होती है। यह डेवलपर्स को एक ही जगह पर कोडिंग, डिजाइनिंग और टेस्टिंग की सारी सुविधाएं देता है।

Android Studio का उपयोग क्यों करें? (Why Use Android Studio?)

मार्केट में ऐप बनाने के लिए और भी टूल्स हैं, लेकिन Android Studio को सबसे बेस्ट माना जाता है।

इसकी मुख्य वजहें निम्नलिखित हैं:

****गूगल का ऑफिशियल सपोर्ट (Official Support):**** चूंकि इसे गूगल खुद मेंटेन करता है, इसलिए एंड्रॉयड के सभी नए फीचर्स, अपडेट्स और सिक्योरिटी पैच सबसे पहले और सबसे बेहतर तरीके से इसी में मिलते हैं।

****विजुअल लेआउट एडिटर (Visual Layout Editor):**** इसमें आपको 'Drag-and-Drop' की सुविधा मिलती है। यानी आप ऐप का डिजाइन (Buttons, Text, Images) स्क्रीन पर खींचकर सेट कर सकते हैं और साथ ही उसका कोड भी देख सकते हैं।

****फास्ट एमुलेटर (Fast Emulator):**** ऐप बनाने के बाद उसे चेक करने के लिए आपको बार-बार असली फोन की जरूरत नहीं होती। Android Studio का अपना एक वर्चुअल फोन (Emulator) होता है, जिसमें आप अलग-अलग स्क्रीन साइज और एंड्रॉयड वर्जन पर अपने ऐप को टेस्ट कर सकते हैं।

****इंटेलिजेंट कोड एडिटर (Intelligent Code Editor):**** यह कोडिंग करते समय आपको सुझाव (Suggestions) देता है, कोड को ऑटो-कंप्लीट करता है और गलतियों को तुरंत पकड़ लेता है। इससे कोडिंग की स्पीड काफी बढ़ जाती है।

****कोटलिन और जावा सपोर्ट (Kotlin & Java Support):**** यह एंड्रॉयड की ऑफिशियल भाषाओं—

****Kotlin**** और ****Java**** को पूरी तरह सपोर्ट करता है। साथ ही C++ के लिए भी इसमें सपोर्ट मौजूद है।

****ग्रेडल बिल्ड सिस्टम (Gradle-Based Build System):**** इसकी मदद से आप अपने ऐप के अलग-अलग वर्जन (जैसे- Free Version और Paid Version) बहुत आसानी से मैनेज और बिल्ड कर सकते हैं।

निष्कर्ष (Conclusion)

अगर आप एंड्रॉयड ऐप डेवलपमेंट की दुनिया में कदम रखना चाहते हैं, तो Android Studio आपके लिए सबसे पहला और सबसे जरूरी टूल है। यह पूरी तरह से ****फ्री (Free)**** है और इसे Windows, Mac और Linux तीनों पर इस्तेमाल किया जा सकता है।

क्या आप इसके इंस्टॉलेशन या इस पर पहला ऐप बनाने के बारे में कुछ जानना चाहते हैं?

File & Project Management

- New >
-  Open... >
-  Profile or Debug APK
- Recent Projects >
- Close Project >
-  Settings... Ctrl+Alt+S
-  Project Structure... Ctrl+Alt+Shift+S
-  File Properties >
-  Local History >
-  Save All Ctrl+S
-  Sync Project with Gradle Files Ctrl+Shift+O
- Refresh Linked C++ Projects
-  Reload All from Disk Ctrl+Alt+Y
-  Repair IDE >
- Invalidate Caches...
- Manage IDE Settings >
- New Projects Setup
- Export >
-  Print... >
- Power Save Mode
- Exit

अध्याय 1: Android Studio का File Menu (Beginner Tutorial)

Android Studio में File Menu का उपयोग प्रोजेक्ट बनाने, खोलने, सेव करने, सेटिंग्स बदलने और IDE को मैनेज करने के लिए किया जाता है। नीचे File Menu के सभी विकल्पों का आसान हिंदी में विवरण दिया गया है।

1. New

उपयोग: नया प्रोजेक्ट, नई फाइल, नई Activity, Fragment या Module बनाने के लिए।

कब उपयोग करें?

नया Android App बनाना हो।

Project में नई Java/Kotlin Class जोड़नी हो।

नई XML Layout या Activity बनानी हो।

2. Open...

उपयोग: पहले से बने Android Project को खोलने के लिए।

उदाहरण: यदि आपने किसी दूसरे दिन बनाया हुआ प्रोजेक्ट खोलना है, तो Open पर क्लिक करें।

3. Profile or Debug APK

उपयोग: किसी APK File का Performance Check या Debug करने के लिए।

कब उपयोग करें?

APK की Speed और Memory Usage देखनी हो।

APK का Analysis करना हो।

4. Recent Projects

उपयोग: हाल ही में खोले गए Projects की सूची दिखाता है।

फायदा: बार-बार Folder खोजने की आवश्यकता नहीं पड़ती।

5. Close Project

उपयोग: वर्तमान Project को बंद करने के लिए।

ध्यान दें: इससे Project Delete नहीं होता, केवल बंद होता है।

6. Settings...

उपयोग: Android Studio की सभी Settings बदलने के लिए।

यहाँ से आप:

Theme बदल सकते हैं।

Font Size बदल सकते हैं।

SDK Configure कर सकते हैं।

Plugins Install कर सकते हैं।

7. Project Structure...

उपयोग: Project की Configuration बदलने के लिए।

यहाँ से आप:

SDK Version बदल सकते हैं।

Dependencies जोड़ सकते हैं।

Modules Manage कर सकते हैं।

8. File Properties

उपयोग: चुनी हुई File की जानकारी देखने के लिए।

9. Local History

उपयोग: File के पुराने Versions देखने और वापस लाने के लिए।

फायदा: गलती से Code Delete हो जाए तो वापस पाया जा सकता है।

10. Save All

उपयोग: सभी खुली हुई Files को एक साथ Save करता है।

Shortcut: Ctrl + S (या Auto Save सक्षम हो सकता है)

11. Sync Project with Gradle Files

उपयोग: Gradle Files में बदलाव करने के बाद Project को Sync करता है।

कब उपयोग करें?

नई Library जोड़ने के बाद।

build.gradle बदलने के बाद।

12. Refresh Linked C++ Projects

उपयोग: C++ (NDK) Project को Refresh करने के लिए।

नोट: सामान्य Android Apps में इसकी आवश्यकता नहीं होती।

13. Reload All from Disk

उपयोग: यदि Files बाहर से बदली गई हैं, तो उन्हें दोबारा Load करता है।

14. Repair IDE

उपयोग: Android Studio में आने वाली सामान्य समस्याओं को ठीक करने का प्रयास करता है।

15. Invalidate Caches...

उपयोग: Cache हटाकर Android Studio को Fresh Start देता है।

कब उपयोग करें?

IDE Slow हो।

Error बार-बार आए।

Indexing में समस्या हो।

16. Manage IDE Settings

उपयोग: Settings Import, Export और Backup करने के लिए।

17. New Projects Setup

उपयोग: नए Project के लिए Default Settings तय करने के लिए।

18. Export

उपयोग: Settings या Configuration Export करने के लिए।

19. Print...

उपयोग: Code या Document को Print करने के लिए।

20. Power Save Mode

उपयोग: Android Studio की Background Processing बंद करके Computer की Battery बचाता है।

ध्यान दें: इस Mode में Code Suggestions और कुछ Features सीमित हो सकते हैं।

21. Exit

उपयोग: Android Studio को पूरी तरह बंद करने के लिए।

Beginner Tip

यदि आप Android Studio सीखना शुरू कर रहे हैं, तो शुरुआत में आपको मुख्य रूप से इन विकल्पों का सबसे अधिक उपयोग होगा:

New

Open

Save All

Settings

Sync Project with Gradle Files

Invalidate Caches

Exit

Main Courses

-  Undo Ctrl+Z
-  Redo Ctrl+Shift+Z

History & Restoration

-  Cut Ctrl+X
-  Copy Ctrl+C
-  Copy Path/Reference... Ctrl+C
-  Paste >
-  Delete Delete

Search & Discovery

-  Search In Selection Ctrl+Alt+E
-  Find >
- Find Usages >

Selection Specials

-  Column Selection Mode Alt+Shift+Insert
-  Select All Ctrl+A
-  Add Carets to Ends of Selected Lines ... Alt+Shift+G
-  Extend Selection Ctrl+W
-  Shrink Selection Ctrl+Shift+W

Text Transmutations

-  Toggle Case Ctrl+Shift+U
-  Join Lines Ctrl+Shift+J
-  Duplicate Line Ctrl+D
- Fill Paragraph
-  Sort Lines
-  Reverse Lines
-  Transpose

Indentation & Organization

-  Indent Selection Tab
-  Unindent Line or Selection Shift+Tab
-  Convert Indents >

Specialized Finishes

-  Macros
-  Bookmarks
- Encode XML/HTML Special Characters

अध्याय 2: Android Studio का Edit Menu (Beginner Tutorial)

Edit Menu Android Studio का एक महत्वपूर्ण मेनू है। इसका उपयोग कोड को Edit, Copy, Paste, Delete, Search, Format और Manage करने के लिए किया जाता है। एक Android Developer के रूप में आप सबसे अधिक इसी मेनू का उपयोग करेंगे।

Undo (Ctrl + Z)

उपयोग: यदि आपने गलती से कोई बदलाव कर दिया है, तो उसे वापस लाने के लिए।

उदाहरण: गलती से कोई लाइन Delete हो गई, तो Undo करके उसे वापस ला सकते हैं।

Redo (Ctrl + Shift + Z)

उपयोग: Undo किए गए कार्य को दोबारा लागू करने के लिए।

Cut (Ctrl + X)

उपयोग: चुने गए Text, Code या File को हटाकर Clipboard में रखता है।

Copy (Ctrl + C)

उपयोग: चुने गए Text या Code की कॉपी बनाता है।

Copy Path/Reference

उपयोग: किसी File, Class, Method या Resource का Path या Reference कॉपी करता है।

उदाहरण: किसी Java Class का पूरा Path कॉपी करके दूसरे Developer को भेज सकते हैं।

Paste

उपयोग: Clipboard में मौजूद Text या Code को वर्तमान स्थान पर Paste करता है।

Delete

उपयोग: चुने गए Text, Code, File या Folder को हटाता है।

Search in Selection (Ctrl + Alt + E)

उपयोग: केवल चुने गए भाग (Selected Text) में Search करता है।

Find

उपयोग: किसी शब्द, Variable, Method या Text को वर्तमान File में खोजने के लिए।

Shortcut: Ctrl + F

Find Usages

उपयोग: किसी Variable, Method या Class का पूरे Project में कहाँ-कहाँ उपयोग हुआ है, यह दिखाता है।

Column Selection Mode (Alt + Shift + Insert)

उपयोग: Column के रूप में Text Select करने की सुविधा देता है।

फायदा: एक साथ कई लाइनों में Editing की जा सकती है।

Select All (Ctrl + A)

उपयोग: पूरी File का Text एक साथ Select करता है।

Add Carets to Ends of Selected Lines (Alt + Shift + G)

उपयोग: चुनी गई सभी लाइनों के अंत में Cursor जोड़ देता है, जिससे एक साथ कई लाइनों में टाइप किया जा सकता है।

Extend Selection (Ctrl + W)

उपयोग: Selection को धीरे-धीरे बढ़ाता है।

Shrink Selection (Ctrl + Shift + W)

उपयोग: बढ़ाए गए Selection को छोटा करता है।

Toggle Case (Ctrl + Shift + U)

उपयोग: Text को Lowercase से UPPERCASE या UPPERCASE से Lowercase में बदलता है।

Join Lines (Ctrl + Shift + J)

उपयोग: दो या अधिक लाइनों को एक लाइन में जोड़ता है।

Duplicate Line (Ctrl + D)

उपयोग: वर्तमान Line की एक नई Copy बना देता है।

Fill Paragraph

उपयोग: Paragraph को सही तरीके से Format करता है।

Sort Lines

उपयोग: चुनी गई लाइनों को Alphabetical क्रम में व्यवस्थित करता है।

Reverse Lines

उपयोग: चुनी गई लाइनों का क्रम उल्टा कर देता है।

Transpose

उपयोग: दो अक्षरों, शब्दों या लाइनों की जगह आपस में बदलता है।

Indent Selection (Tab)

उपयोग: चुने गए Code को दाईं ओर Indent करता है।

Unindent Line or Selection (Shift + Tab)

उपयोग: Code का Indentation कम करता है।

Convert Indents

उपयोग: Indentation को Tabs और Spaces के बीच बदलता है।

Macros

उपयोग: बार-बार किए जाने वाले कार्यों को Record करके बाद में दोबारा चलाने की सुविधा देता है।

Bookmarks

उपयोग: महत्वपूर्ण Code Lines पर Bookmark लगाकर बाद में तुरंत पहुँचने के लिए।

Encode XML/HTML Special Characters

उपयोग: XML और HTML के विशेष Characters (<, >, &, ") को सुरक्षित Format में बदलता है।

Beginner Tips

Android Studio में Edit Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

Undo

Redo

Cut

Copy

Paste

Find

Find Usages

Select All

Duplicate Line

Toggle Case

Navigation & Editor Views

- 📁 Tool Windows >
- 📁 Appearance >
 - Quick Definition Ctrl+Shift+I
 - Show Siblings
 - Quick Type Definition
 - Quick Documentation Ctrl+Q
 - Parameter Info Ctrl+P
 - Type Info Ctrl+Shift+P
 - Context Info Alt+Q
- 🔍 Jump to Source F4
 - Recent Locations Ctrl+Shift+E
 - Recent Files Ctrl+E
 - Recently Changed Files
 - Recent Changes Alt+Shift+C
- ↔ Compare With... Ctrl+D
- 📄 Compare with Clipboard
- Quick Switch Scheme... Ctrl`
- Active Editor >
 - Increase Font Size in All Editors Alt+Shift+.
 - Decrease Font Size in all Editors Alt+Shift+Comma
 - Reset Font Size in All Editors
 - Bidi Text Base Direction >

अध्याय 3: Android Studio का View Menu (Beginner Tutorial)

View Menu का उपयोग Android Studio के Interface (UI), Tool Windows, Editor View, Navigation तथा Display Settings को नियंत्रित करने के लिए किया जाता है। इस मेनू की सहायता से आप अपनी आवश्यकता के अनुसार Android Studio के कार्यक्षेत्र (Workspace) को व्यवस्थित कर सकते हैं।

Tool Windows

उपयोग: Android Studio की सभी Tool Windows जैसे Project, Logcat, Terminal, Build, Device Manager, Profiler आदि को खोलने या छिपाने के लिए।

उदाहरण: यदि Project Panel बंद हो गया है, तो View → Tool Windows → Project चुनकर उसे दोबारा खोल सकते हैं।

Appearance

उपयोग: Android Studio के Interface का रूप (Appearance) बदलने के लिए।

इसके अंतर्गत आप:

Toolbar दिखा या छिपा सकते हैं।

Status Bar दिखा या छिपा सकते हैं।

Navigation Bar चालू या बंद कर सकते हैं।

Full Screen Mode का उपयोग कर सकते हैं।

Quick Definition (Ctrl + Shift + I)

उपयोग: किसी Class, Method या Variable की परिभाषा (Definition) बिना File खोले दिखाता है।

Show Siblings

उपयोग: वर्तमान File से संबंधित अन्य Files या Classes दिखाता है।

Quick Type Definition

उपयोग: Cursor के नीचे मौजूद Variable या Object का Data Type तुरंत दिखाता है।

Quick Documentation (Ctrl + Q)

उपयोग: Class, Method या Function की Documentation तुरंत दिखाता है।

उदाहरण: किसी Method पर Cursor रखें और Ctrl + Q दबाएँ।

Parameter Info (Ctrl + P)

उपयोग: Method या Function के Parameters की जानकारी दिखाता है।

उदाहरण: Toast.makeText() लिखते समय उसके Parameters दिखाई देंगे।

Type Info (Ctrl + Shift + P)

उपयोग: Variable, Object या Expression का Type बताता है।

Context Info (Alt + Q)

उपयोग: Cursor जिस स्थान पर है, उससे संबंधित अतिरिक्त जानकारी दिखाता है।

Jump to Source (F4)

उपयोग: चुने गए Class, Method या Resource की Source File पर सीधे पहुँचाता है।

Recent Locations (Ctrl + Shift + E)

उपयोग: हाल ही में जिन स्थानों पर आपने Editing की है, उनकी सूची दिखाता है।

Recent Files (Ctrl + E)

उपयोग: हाल ही में खोली गई Files की सूची दिखाता है।

Recently Changed Files

उपयोग: हाल ही में जिन Files में बदलाव किए गए हैं, उन्हें दिखाता है।

Recent Changes (Alt + Shift + C)

उपयोग: Project में हाल ही में किए गए परिवर्तनों की सूची दिखाता है।

Compare With... (Ctrl + D)

उपयोग: वर्तमान File की तुलना किसी दूसरी File या Version से करता है।

Compare with Clipboard

उपयोग: वर्तमान File की तुलना Clipboard में कॉपी किए गए Text से करता है।

Quick Switch Scheme

उपयोग: Keymap, Theme, Color Scheme आदि को जल्दी बदलने के लिए।

Active Editor

उपयोग: वर्तमान में खुले हुए Editor से संबंधित विकल्प दिखाता है।

Increase Font Size in All Editors (Alt + Shift + .)

उपयोग: सभी Editors में Font Size बढ़ाता है।

Decrease Font Size in All Editors (Alt + Shift + ,)

उपयोग: सभी Editors में Font Size कम करता है।

Reset Font Size in All Editors

उपयोग: Font Size को Default स्थिति में वापस लाता है।

Bidi Text Base Direction

उपयोग: Text की दिशा Left-to-Right (LTR) या Right-to-Left (RTL) निर्धारित करता है।

नोट: यह विकल्प मुख्य रूप से अरबी, उर्दू और हिब्रू जैसी भाषाओं के लिए उपयोगी है।

Beginner Tips

Android Studio में View Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

Tool Windows

Appearance

Quick Documentation

Parameter Info

Jump to Source

Recent Files

Recent Locations

Increase Font Size in All Editors

Decrease Font Size in All Editors

नोट: View Menu आपको Android Studio के कार्यक्षेत्र को अपनी सुविधा के अनुसार व्यवस्थित करने की शक्ति देता है। यदि आप इसके प्रमुख विकल्पों का अभ्यास करेंगे, तो Coding अधिक तेज़, व्यवस्थित और आसान हो जाएगी।

← Back Ctrl+Alt+Left
→ Forward Ctrl+Alt+Right

🔍 Search Everywhere

Class... Ctrl+N

File... Ctrl+Shift+N

Symbol... Ctrl+Alt+Shift+N

Text... Ctrl+Alt+Shift+E

Line:Column... Ctrl+G

Next Highlighted Error F2

Previous Highlighted Error Shift+F2

Last Edit Location Ctrl+Shift+Backspace

Next Edit Location

Navigate in File >

Select In... Alt+F1

Jump to Navigation Bar Alt+Home

Declaration or Usages Ctrl+B

Definition Ctrl+Alt+B

Type Declaration Ctrl+Shift+B

Super Method Ctrl+U

Test Ctrl+Shift+T

Related Symbol... Ctrl+Alt+Home

Header/Source F10

File Structure Ctrl+F12

File Path Ctrl+Alt+F12

Imports Hierarchy Alt+Shift+H

Type Hierarchy Ctrl+H

Method Hierarchy Ctrl+Shift+H

Call Hierarchy Ctrl+Alt+H

अध्याय 4: Android Studio का Navigate Menu (Beginner Tutorial)

Navigate Menu का उपयोग Android Studio में किसी भी Class, File, Method, Variable, Error या Code Location तक तुरंत पहुँचने के लिए किया जाता है। बड़े Android Projects में यह Menu डेवलपर का बहुत समय बचाता है और Coding को तेज़ बनाता है।

Back (Ctrl + Alt + Left)

उपयोग: जिस स्थान पर आप पहले काम कर रहे थे, वहाँ वापस जाने के लिए।

उदाहरण: यदि आप MainActivity से LoginActivity में गए हैं, तो Back दबाकर फिर MainActivity पर लौट सकते हैं।

Forward (Ctrl + Alt + Right)

उपयोग: Back करने के बाद अगले स्थान पर वापस जाने के लिए।

Search Everywhere

उपयोग: पूरे Project में Class, File, Method, Setting, Action या Symbol खोजने के लिए।

Shortcut: Shift कुंजी को लगातार दो बार दबाएँ।

उदाहरण: MainActivity, AndroidManifest.xml या Gradle लिखकर तुरंत खोज सकते हैं।

Class... (Ctrl + N)

उपयोग: किसी भी Java या Kotlin Class को उसके नाम से तुरंत खोलने के लिए।

File... (Ctrl + Shift + N)

उपयोग: Project की किसी भी File को खोजकर खोलने के लिए।

Symbol... (Ctrl + Alt + Shift + N)

उपयोग: किसी Method, Function, Variable या Symbol को खोजने के लिए।

Text...

उपयोग: पूरे Project में किसी शब्द या Text को खोजने के लिए।

नोट: सामान्यतः इसका Shortcut Ctrl + Shift + F होता है।

Line:Column... (Ctrl + G)

उपयोग: किसी निश्चित Line Number और Column Number पर सीधे जाने के लिए।

उदाहरण: Line 250 पर जाना हो तो Ctrl + G दबाएँ।

Next Highlighted Error (F2)

उपयोग: अगले Error या Warning पर जाने के लिए।

Previous Highlighted Error (Shift + F2)

उपयोग: पिछले Error या Warning पर जाने के लिए।

Last Edit Location (Ctrl + Shift + Backspace)

उपयोग: जहाँ आपने आखिरी बार Code Edit किया था, वहाँ वापस ले जाता है।

Next Edit Location

उपयोग: अगली Edit की गई Location पर जाने के लिए।

Navigate in File

उपयोग: वर्तमान File के अंदर मौजूद Methods, Variables और Sections के बीच जल्दी पहुँचने के लिए।

Select In... (Alt + F1)

उपयोग: वर्तमान File को Project Window या अन्य Tool Window में Locate करता है।

Jump to Navigation Bar (Alt + Home)

उपयोग: Navigation Bar पर Cursor ले जाता है।

Declaration or Usages (Ctrl + B)

उपयोग: किसी Variable, Method या Class की Definition या उसके उपयोग (Usages) पर पहुँचने के लिए।

Definition (Ctrl + Alt + B)

उपयोग: Interface या Method की वास्तविक Implementation (Source Code) खोलता है।

Type Declaration (Ctrl + Shift + B)

उपयोग: Variable या Object के Data Type की Declaration पर पहुँचाता है।

Super Method (Ctrl + U)

उपयोग: यदि Method Parent Class से Override हुई है, तो Parent method पर ले जाता है।

Test (Ctrl + Shift + T)

उपयोग: संबंधित Test Class खोलने या नई Test Class बनाने के लिए।

Related Symbol... (Ctrl + Alt + Home)

उपयोग: वर्तमान Class या Method से संबंधित अन्य Symbols दिखाता है।

Header/Source (F10)

उपयोग: Header File और Source File के बीच जल्दी Switch करने के लिए।

File Structure (Ctrl + F12)

उपयोग: वर्तमान File के सभी Methods, Variables और Members की सूची दिखाता है।

File Path (Ctrl + Alt + F12)

उपयोग: वर्तमान File का पूरा Folder Path दिखाता है।

Imports Hierarchy (Alt + Shift + H)

उपयोग: Import की गई Classes और उनके संबंधों की जानकारी दिखाता है।

Type Hierarchy (Ctrl + H)

उपयोग: किसी Class की Parent Class और Child Classes का Hierarchy Tree दिखाता है।

Method Hierarchy (Ctrl + Shift + H)

उपयोग: किसी Method की Hierarchy और उसके संबंध दिखाता है।

Call Hierarchy (Ctrl + Alt + H)

उपयोग: यह दिखाता है कि किसी Method को Project में कहाँ-कहाँ Call किया गया है।

Beginner Tips

Android Studio में Navigate Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

- *Search Everywhere*
- *Class*
- *File*
- *Text*
- *Line:Column*
- *Declaration or Usages*
- *File Structure*
- *Type Hierarchy*
- *Call Hierarchy*
- *Back और Forward*

महत्वपूर्ण नोट: यदि आप Navigate Menu के Keyboard Shortcuts याद कर लेते हैं, तो बड़े Android Projects में भी किसी भी Class, File या Method तक कुछ ही सेकंड में पहुँच सकते हैं। इससे आपकी Coding Speed, Accuracy और Productivity में काफी सुधार होता है।

Code & Refactor Menu

Override Methods... Ctrl+O

Implement Methods... Ctrl+I

Delegate Methods...

Generate... Alt+Insert

Code Completion >

Inspect Code...

Inspect for Play Policy Insights...

Code Cleanup...

Analyze Code >

Analyze Stack Trace or Thread Dump...

Insert Live Template... Ctrl+J

Save as Live Template...

Surround With... Ctrl+Alt+T

Unwrap/Remove... Ctrl+Shift+Delete

Folding >

Comment with Line Comment Ctrl+/
>

Comment with Block Comment Ctrl+Shift+/
>

Reformat Code Ctrl+Alt+L

Reformat File... Ctrl+Alt+Shift+L

Auto-Indent Lines Ctrl+Alt+I

Optimize Imports Ctrl+Alt+O

Rearrange Code

Move Statement Down Ctrl+Shift+Down

Move Statement Up Ctrl+Shift+Up

Move Element Left Ctrl+Alt+Shift+Left

Move Element Right ... Ctrl+Alt+Shift+Right

Move Line Down Alt+Shift+Down

Move Line Up Alt+Shift+Up

Update Copyright...

अध्याय 5: Android Studio का Code Menu (Beginner Tutorial)

Code Menu Android Studio का सबसे महत्वपूर्ण मेनू है। इसका उपयोग कोड लिखने, Code Generate करने, Formatting करने, Comments जोड़ने, Imports व्यवस्थित करने तथा Code Quality सुधारने के लिए किया जाता है। यदि आप Android Development सीख रहे हैं, तो इस मेनू का नियमित उपयोग आपकी Coding Speed और Productivity दोनों बढ़ा देगा।

Override Methods... (Ctrl + O)

उपयोग: Parent Class के Methods को Override करने के लिए।

उदाहरण: onCreate(), onResume(), onDestroy() आदि Methods जोड़ने के लिए।

Implement Methods... (Ctrl + I)

उपयोग: Interface या Abstract Class के आवश्यक Methods को Implement करने के लिए।

Delegate Methods...

उपयोग: किसी Object के Methods को वर्तमान Class में Delegate करने के लिए।

Generate... (Alt + Insert)

उपयोग: Android Studio द्वारा स्वतः Code Generate करने के लिए।

इसमें आप बना सकते हैं:

- Constructor
- Getter & Setter
- toString()
- equals() और hashCode()
- Override Methods

Code Completion

उपयोग: Code लिखते समय Auto Suggestion और Auto Completion प्रदान करता है।

उदाहरण: Toast लिखते ही संबंधित सुझाव दिखाई देते हैं।

Inspect Code...

उपयोग: पूरे Project के Code में Errors, Warnings और Code Quality की जाँच करता है।

Inspect for Play Policy Insights...

उपयोग: Google Play Policy से संबंधित संभावित समस्याओं की पहचान करता है।

Code Cleanup...

उपयोग: अनावश्यक Code हटाकर Code को साफ और व्यवस्थित करता है।

Analyze Code

उपयोग: Code की गुणवत्ता, Performance और संभावित Bugs का विश्लेषण करता है।

Analyze Stack Trace or Thread Dump...

उपयोग: Crash Logs (Stack Trace) का विश्लेषण करके Error का कारण पता लगाने में सहायता करता है।

Insert Live Template... (Ctrl + J)

उपयोग: पहले से बने Code Templates जल्दी Insert करने के लिए।

उदाहरण: for, if, sout आदि Templates।

Save as Live Template...

उपयोग: अपने स्वयं के Code Snippets को Live Template के रूप में Save करने के लिए।

Surround With... (Ctrl + Alt + T)

उपयोग: चुने गए Code को if, try-catch, while आदि Blocks में बदलने के लिए।

Unwrap/Remove... (Ctrl + Shift + Delete)

उपयोग: if, try, while जैसे Block हटाकर केवल अंदर का Code रखता है।

Folding

उपयोग: बड़े Code Blocks को Collapse (छिपाना) या Expand (दिखाना) करने के लिए।

Comment with Line Comment (Ctrl + /)

उपयोग: किसी Line को Comment या Uncomment करने के लिए।

Comment with Block Comment (Ctrl + Shift + /)

उपयोग: कई Lines को एक साथ Block Comment में बदलने के लिए।

Reformat Code (Ctrl + Alt + L)

उपयोग: पूरे Code को Android Studio Coding Style के अनुसार व्यवस्थित करता है।

Reformat File... (Ctrl + Alt + Shift + L)

उपयोग: पूरी File को Format करने के लिए अतिरिक्त विकल्प प्रदान करता है।

Auto-Indent Lines (Ctrl + Alt + I)

उपयोग: चुनी गई Lines का Indentation स्वतः ठीक करता है।

Optimize Imports (Ctrl + Alt + O)

उपयोग: Unused Imports हटाता है और आवश्यक Imports को व्यवस्थित करता है।

Rearrange Code

उपयोग: Code को निर्धारित नियमों के अनुसार क्रमबद्ध करता है।

Move Statement Down (Ctrl + Shift + Down)

उपयोग: वर्तमान Statement को नीचे ले जाता है।

Move Statement Up (Ctrl + Shift + Up)

उपयोग: वर्तमान Statement को ऊपर ले जाता है।

Move Element Left (Ctrl + Alt + Shift + Left)

उपयोग: Code Element को बाईं ओर स्थानांतरित करता है।

Move Element Right (Ctrl + Alt + Shift + Right)

उपयोग: Code Element को दाईं ओर स्थानांतरित करता है।

Move Line Down (Alt + Shift + Down)

उपयोग: वर्तमान Line को नीचे ले जाता है।

Move Line Up (Alt + Shift + Up)

उपयोग: वर्तमान Line को ऊपर ले जाता है।

Update Copyright...

उपयोग: Source Files में Copyright Header जोड़ने या अपडेट करने के लिए।

Beginner Tips

Android Studio में Code Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

- *Generate*
- *Code Completion*
- *Insert Live Template*
- *Comment with Line Comment*
- *Comment with Block Comment*
- *Reformat Code*
- *Optimize Imports*
- *Auto-Indent Lines*
- *Override Methods*
- *Implement Methods*

महत्वपूर्ण नोट: यदि आप Code Menu के Keyboard Shortcuts का नियमित अभ्यास करते हैं, तो आप कम समय में अधिक साफ, व्यवस्थित और Professional Android Code लिख पाएँगे।

Refactor Menu

Refactor This... Ctrl+Alt+

Rename... S

Rename File...

Change Signature...

Extract/Introduce

Inline... Ctrl

Move...

Copy...

Safe Delete... Alt

Pull Members Up...

Pull Members Up...

Push Members Down...

Migrate Packages and Classes

Internationalize... Z

Modularize...

Remove Unused Resources...

Add Right-to-Left (RTL) Support...

Add Right-to-Left (RTL) Support...

Migrate to Non-Transitive R Classe...

Migrate BuildConfig to Gradle Build...

Migrate ResValueus to Gradle Build...

Default TargetSdk to CompileSdk...

अध्याय 6: Android Studio का Refactor Menu (Beginner Tutorial)

Refactor Menu Android Studio का एक शक्तिशाली (Powerful) मेनू है। इसका उपयोग Code को बेहतर, व्यवस्थित, सुरक्षित और Maintainable बनाने के लिए किया जाता है। Refactoring का अर्थ है Code की गुणवत्ता सुधारना, बिना उसके कार्य (Functionality) को बदले।

Refactor This... (Ctrl + Alt + Shift + T)

उपयोग: सभी Refactoring विकल्पों की सूची एक साथ खोलता है।

उदाहरण: Rename, Move, Extract Method आदि विकल्प यहीं से जल्दी चुने जा सकते हैं।

Rename... (Shift + F6)

उपयोग: किसी Class, Method, Variable, Package या Resource का नाम बदलने के लिए।

फायदा: Android Studio पूरे Project में जहाँ-जहाँ वह नाम उपयोग हुआ है, उसे स्वतः बदल देता है।

Rename File...

उपयोग: किसी File का नाम बदलने के लिए।

Change Signature... (Ctrl + F6)

उपयोग: Method के Parameters, Return Type या Method Name को बदलने के लिए।

उदाहरण: यदि किसी Method में नया Parameter जोड़ना हो।

Extract/Introduce

उपयोग: Code के किसी भाग को अलग Method, Variable, Constant, Field या Parameter में बदलने के लिए।

फायदा: Code छोटा, साफ और पुनः उपयोग योग्य (Reusable) बनता है।

Inline... (Ctrl + Alt + N)

उपयोग: Variable, Method या Constant को हटाकर उसकी Value सीधे Code में लिख देता है।

Move... (F6)

उपयोग: Class, File, Package या Resource को दूसरी जगह स्थानांतरित (Move) करता है।

Copy... (F5)

उपयोग: Class, File या Package की Copy बनाता है।

Safe Delete... (Alt + Delete)

उपयोग: किसी Class, Method, Variable या File को सुरक्षित रूप से Delete करता है।

फायदा: Delete करने से पहले Android Studio जाँच करता है कि कहीं उसका उपयोग तो नहीं हो रहा।

Pull Members Up...

उपयोग: Child Class के Methods या Variables को Parent Class में भेजता है।

Push Members Down...

उपयोग: Parent Class के Members को Child Class में भेजता है।

Migrate Packages and Classes

उपयोग: Package या Class को नए Package या Location में स्थानांतरित करता है।

Internationalize...

उपयोग: App के Text को strings.xml में भेजकर बहुभाषी (Multi-language) App बनाने में सहायता करता है।

Modularize...

उपयोग: Project को अलग-अलग Modules में विभाजित करता है।

फायदा: बड़े Projects को Manage करना आसान हो जाता है।

Remove Unused Resources...

उपयोग: Project में मौजूद बेकार (Unused) Images, Strings, Layouts और अन्य Resources हटाता है।

फायदा: App का Size कम होता है।

Add Right-to-Left (RTL) Support...

उपयोग: App में Arabic, Urdu, Hebrew जैसी Right-to-Left भाषाओं का समर्थन जोड़ता है।

Migrate to Non-Transitive R Class...

उपयोग: Project को नए Android Resource System के अनुसार अपडेट करता है।

Migrate BuildConfig to Gradle Build...

उपयोग: BuildConfig Settings को Gradle Build Script में स्थानांतरित करता है।

Migrate ResValues to Gradle Build...

उपयोग: Resource Values को Gradle Configuration में स्थानांतरित करता है।

Default TargetSdk to CompileSdk...

उपयोग: Project की Target SDK Settings को नई Android Build Configuration के अनुसार अपडेट करता है।

Beginner Tips

Android Studio में Refactor Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

Rename

Move

Copy

Safe Delete

Extract Method / Extract Variable

Change Signature

Remove Unused Resources

Refactor This

महत्वपूर्ण नोट: शुरुआती Android Developers के लिए सबसे उपयोगी Refactor विकल्प Rename, Extract, Move और Safe Delete हैं। इनका सही उपयोग करने से आपका Code साफ, व्यवस्थित और Professional बनता है, साथ ही बड़ी परियोजनाओं में बदलाव करना भी आसान हो जाता है।

Build Menu

 Assemble 'app' Run Configuration

Compile Selected Modules

Compile All Sources

 Assemble Selected Modules

 Assemble Project

 Assemble Project with Tests

Generate

Generate App Bundles or APKs

Generate Signed App Bundle or
APK...

Analyze APK...

Analyze Build Performance

Clean Project

Clean and Assemble Project with
with Tests

Select Build Variant...

अध्याय 7: Android Studio का Build Menu (Beginner Tutorial)

Build Menu Android Studio का एक महत्वपूर्ण मेनू है। इसका उपयोग Project को Compile करने, APK या App Bundle (AAB) बनाने, Build Errors की जाँच करने तथा Build Process को Manage करने के लिए किया जाता है। जब आपका App तैयार हो जाता है, तो इसी Menu की सहायता से Install करने योग्य APK या Google Play Store के लिए AAB (Android App Bundle) बनाया जाता है।

Assemble 'app' Run Configuration

उपयोग: वर्तमान Run Configuration के अनुसार App Module का Build तैयार करता है।

उदाहरण: यदि आपका Module app है, तो यह उसका Build Generate करेगा।

Compile Selected Modules

उपयोग: केवल चुने गए Module को Compile करता है।

फायदा: पूरे Project की बजाय केवल आवश्यक Module Compile होने से समय बचता है।

Compile All Sources

उपयोग: Project की सभी Source Files (Java, Kotlin आदि) को Compile करता है।

Assemble Selected Modules

उपयोग: चुने गए Module का APK या Build Output तैयार करता है।

Assemble Project

उपयोग: पूरे Project का Build तैयार करता है।

नोट: यह App को Run नहीं करता, केवल Build Files तैयार करता है।

Assemble Project with Tests

उपयोग: Project का Build तैयार करने के साथ-साथ Test Modules भी Build करता है।

Generate

उपयोग: Build से संबंधित अतिरिक्त विकल्प खोलता है।

Generate App Bundles or APKs

उपयोग: Android App Bundle (AAB) या APK बनाने के विकल्प देता है।

इसमें आप बना सकते हैं:

- Debug APK
- Release APK
- Android App Bundle (.aab)

Generate Signed App Bundle or APK...

उपयोग: Google Play Store पर प्रकाशित करने के लिए Signed APK या Signed App Bundle बनाता है।

महत्वपूर्ण: इसके लिए Keystore File, Key Alias और Password की आवश्यकता होती है।

Analyze APK...

उपयोग: किसी APK का विश्लेषण (Analyze) करता है।

आप देख सकते हैं:

- APK Size
- Resources
- Classes
- Libraries
- Manifest

Analyze Build Performance

उपयोग: Build Process में कितना समय लग रहा है और कहाँ Performance कम है, इसका विश्लेषण करता है।

Clean Project

उपयोग: Project की पुरानी Build Files हटाता है।

कब उपयोग करें?

- Build Error आने पर
- Gradle समस्या होने पर
- Project सही Build न हो रहा हो

Clean and Assemble Project with Tests

उपयोग: पहले Project को Clean करता है, फिर दोबारा Build करता है और Tests भी तैयार करता है।

Select Build Variant...

उपयोग: Project का Build Variant चुनने के लिए।

सामान्य Build Variants:

- Debug – Testing और Development के लिए।
- Release – Play Store पर Publish करने के लिए।

Beginner Tips

Android Studio में Build Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

- *Generate App Bundles or APKs*
- *Generate Signed App Bundle or APK*
- *Clean Project*
- *Assemble Project*
- *Analyze APK*
- *Select Build Variant*

महत्वपूर्ण नोट

यदि आप Android App को केवल अपने मोबाइल पर चलाना चाहते हैं, तो Debug APK पर्याप्त है।

यदि आप अपना App Google Play Store पर प्रकाशित करना चाहते हैं, तो हमेशा Generate Signed App Bundle (.AAB) या Signed APK का उपयोग करें।

Beginner Advice: शुरुआत में आपको सबसे अधिक Clean Project, Generate App Bundles or APKs, Generate Signed App Bundle or APK और Select Build Variant का अभ्यास करना चाहिए। यही विकल्प Android App को सफलतापूर्वक Build और Publish करने के लिए सबसे अधिक उपयोग किए जाते हैं।

Run & Debug Menu

- ▶ Run 'app'
- ⛔ Apply Changes and Restart Activity (disabled...)
- ✂ Apply Code Changes (disabled...)
- 🐛 Debug 'app'
- ⌚ Run 'app' with Coverage
- 🔍 Profiler: Run 'app' as profileable...
- ▶ Run...
- 🐛 Debug...
- 🔍 Profile...
- ✂ Record Espresso Test
- 🔗 Attach to Process...
- ⚙ Edit Configurations...
- 🕒 Select Device...
- 🖐 Stop
 - Stop Background Processes
 - Show Running List
- 🔍 Debugging Actions
 - Toggle Breakpoint
 - 🔴 View Breakpoints...
- ⌚ Test History
- 📄 Import Tests from File...
- 📍 Manage Coverage Reports...
- 🔗 Attach Debugger to Android Process
- 📦 Backup App Data

अध्याय 8: Android Studio का Run & Debug Menu (Beginner Tutorial)

Run & Debug Menu Android Studio का सबसे महत्वपूर्ण Menu है। इसकी सहायता से आप अपने Android App को Run, Debug, Test, Profile और Monitor कर सकते हैं। जब भी आप अपना App मोबाइल या Emulator पर चलाना चाहते हैं, तो सबसे अधिक उपयोग इसी Menu का होता है।

Run 'app'

उपयोग: App को Emulator या Android Device पर चलाने (Run) के लिए।

Shortcut: Shift + F10

उदाहरण: Coding पूरी होने के बाद App को टेस्ट करने के लिए इसी विकल्प का उपयोग किया जाता है।

Apply Changes and Restart Activity

उपयोग: Code में किए गए छोटे बदलावों को बिना पूरा App दोबारा Install किए लागू करता है और Activity को Restart करता है।

नोट: यदि यह विकल्प Disabled दिखाई दे, तो इसका अर्थ है कि वर्तमान Project में इसका उपयोग संभव नहीं है।

Apply Code Changes

उपयोग: छोटे Code Changes को तुरंत Running App में लागू करता है।

फायदा: पूरा App दोबारा Build नहीं करना पड़ता।

Debug 'app'

उपयोग: App को Debug Mode में चलाता है ताकि Errors और Bugs आसानी से खोजे जा सकें।

Shortcut: Shift + F9

Run 'app' with Coverage

उपयोग: यह जाँचता है कि आपके Tests ने Code के कितने भाग को Execute किया है।

फायदा: Code Coverage Report प्राप्त होती है।

Profiler: Run 'app' as profileable

उपयोग: App की Performance जैसे CPU, Memory, Battery और Network Usage का विश्लेषण करता है।

Run...

उपयोग: किसी विशेष Run Configuration के अनुसार Project चलाने के लिए।

Debug...

उपयोग: किसी विशेष Debug Configuration के साथ App Debug करने के लिए।

Profile...

उपयोग: चुनी गई Configuration के अनुसार App की Performance Monitor करने के लिए।

Record Espresso Test

उपयोग: UI Testing के लिए Espresso Test Record करता है।

उदाहरण: Button Click या Screen Navigation जैसी Activities को रिकॉर्ड करना।

Attach to Process...

उपयोग: पहले से चल रहे Android Process से Debugger जोड़ने के लिए।

Edit Configurations...

उपयोग: Run और Debug Configurations को बनाने, बदलने या Delete करने के लिए।

Select Device...

उपयोग: Emulator या Connected Android Device चुनने के लिए।

Stop

उपयोग: चल रहे App या Debug Session को बंद करता है।

Stop Background Processes

उपयोग: Android Studio के Background में चल रही Processes को रोकता है।

Show Running List

उपयोग: वर्तमान में चल रहे सभी Run और Debug Sessions की सूची दिखाता है

Debugging Actions

उपयोग: Debugging से संबंधित अतिरिक्त विकल्प उपलब्ध कराता है।

Toggle Breakpoint

उपयोग: किसी Line पर Breakpoint लगाने या हटाने के लिए।

फायदा: Program उसी स्थान पर रुक जाता है और आप Variables तथा Code की स्थिति देख सकते हैं।

View Breakpoints...

उपयोग: Project के सभी Breakpoints की सूची दिखाता है तथा उन्हें Manage करने की सुविधा देता है।

Test History

उपयोग: पहले चलाए गए सभी Tests का इतिहास (History) दिखाता है।

Import Tests from File...

उपयोग: किसी File से Test Configuration Import करने के लिए।

Manage Coverage Reports...

उपयोग: Code Coverage Reports को देखने, Export करने और Manage करने के लिए।

Attach Debugger to Android Process

उपयोग: पहले से चल रहे Android App से Debugger जोड़ने के लिए।

उदाहरण: यदि App पहले से Mobile में चल रहा है और उसे Debug करना हो।

Backup App Data

उपयोग: App के Data का Backup बनाने के लिए।

फायदा: Testing के दौरान User Data सुरक्षित रखा जा सकता है।

Beginner Tips

Android Studio में Run & Debug Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

Run 'app'

Debug 'app'

Select Device

Edit Configurations

Stop

Toggle Breakpoint

View Breakpoints

Profiler

महत्वपूर्ण नोट

यदि आप Android Development सीख रहे हैं, तो शुरुआत में मुख्य रूप से Run 'app', Debug 'app', Select Device, Stop और Toggle Breakpoint का अभ्यास करें। जब आपका App सही तरीके से चलने लगे, तब Profiler, Code Coverage और Espresso Testing जैसे उन्नत (Advanced) विकल्पों का उपयोग करना शुरू करें।

> Beginner Advice: Android App बनाते समय सबसे अधिक उपयोग Run, Debug और Breakpoint का होता है। इन तीनों सुविधाओं को अच्छी तरह सीख लेने पर आप अपने App की अधिकांश समस्याओं को स्वयं पहचानकर आसानी से ठीक कर सकेंगे।

Tools Menu



Device Manager



SDK Manager



Troubleshoot Device
Connections



Firebase



Apply Changes and Restart

Apply Changes and Restart Activity (disabled...)



App Links Assistant



Layout Inspector



Tasks & Contexts...

Analyze APK...

Generate Javadoc...

Create Command Line Launcher...



Services >



Markdown >



XML >



XML Actions >



Kotlin >



Groovy Console



Cling >



अध्याय 9: Android Studio का Tools Menu (Beginner Tutorial)

Tools Menu Android Studio का एक महत्वपूर्ण Menu है। इसकी सहायता से आप Android SDK, Emulator, Firebase, APK Analysis, Layout Inspection, Documentation, तथा अन्य Development Tools का उपयोग कर सकते हैं। यदि आप Android App Development सीख रहे हैं, तो यह Menu आपके लिए कई उपयोगी सुविधाएँ उपलब्ध कराता है।

Device Manager

उपयोग: Android Emulator (Virtual Device) बनाने, चलाने और प्रबंधित (Manage) करने के लिए।

उदाहरण: यदि आपके पास वास्तविक Android Mobile नहीं है, तो Device Manager से Virtual Phone बनाकर App चला सकते हैं।

SDK Manager

उपयोग: Android SDK, Platform Tools, Build Tools और Emulator Components को Install या Update करने के लिए।

फायदा: नए Android Versions के लिए आवश्यक SDK यहीं से डाउनलोड किए जाते हैं।

Troubleshoot Device Connections

उपयोग: यदि Android Device USB के माध्यम से Connect नहीं हो रहा है, तो उसकी समस्या का समाधान करने में सहायता करता है।

Firebase

उपयोग: Android App को Firebase Services से जोड़ने के लिए।

Firebase की प्रमुख सेवाएँ:

- Authentication
- Firestore Database
- Realtime Database
- Cloud Messaging (FCM)
- Analytics
- Crashlytics
- Cloud Storage

Apply Changes and Restart

उपयोग: छोटे Code Changes को बिना पूरा App दोबारा Install किए लागू करता है तथा Activity को Restart करता है।

App Links Assistant

उपयोग: Android App Links और Deep Links Configure करने में सहायता करता है।

उदाहरण: Website Link पर क्लिक करते ही आपका Android App खुल सकता है।

Layout Inspector

उपयोग: Running App के Layout का निरीक्षण (Inspect) करने के लिए।

फायदा:

- View Hierarchy देख सकते हैं।
- Layout की समस्याएँ पहचान सकते हैं।
- UI Debugging आसान हो जाती है।

Tasks & Contexts...

उपयोग: Development Tasks को Manage करने तथा IDE के Context को Save और Restore करने के लिए।

Analyze APK...

उपयोग: APK File का विश्लेषण (Analyze) करता है।

आप देख सकते हैं:

- APK Size
- Resources
- Classes
- Libraries
- AndroidManifest.xml

Generate Javadoc...

उपयोग: Java Classes और Methods की Documentation (Javadoc) तैयार करता है।

Create Command Line Launcher...

उपयोग: Android Studio को Command Line से शुरू करने के लिए Launcher बनाता है।

Services

उपयोग: Background Services, Connected Devices, Emulator तथा अन्य Running Services को Manage करता है।

Markdown

उपयोग: Markdown Files (.md) को Edit और Preview करने के लिए।

उदाहरण: README.md File।

XML

उपयोग: XML Files से संबंधित अतिरिक्त Tools और Options उपलब्ध कराता है।

XML Actions

उपयोग: XML Files को Validate, Format और Optimize करने के लिए।

Kotlin

उपयोग: Kotlin Language से संबंधित Tools और Utilities उपलब्ध कराता है।

उदाहरण:

- **Java to Kotlin Conversion**
- **Kotlin Configuration**

Groovy Console

उपयोग: Groovy Scripts चलाने और Testing करने के लिए।

Cling

उपयोग: Android Studio के कुछ Plugins या Internal Features से संबंधित विकल्प उपलब्ध कराता है। सामान्य Android Development में इसका उपयोग बहुत कम होता है।

Beginner Tips

Android Studio में Tools Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

- ***Device Manager***
- ***SDK Manager***
- ***Firebase***
- ***Layout Inspector***
- ***Analyze APK***
- ***Kotlin***
- ***XML***
- ***Troubleshoot Device Connections***

महत्वपूर्ण नोट

यदि आप Android Development की शुरुआत कर रहे हैं, तो सबसे पहले Device Manager, SDK Manager, Firebase और Layout Inspector का उपयोग करना सीखें। ये विकल्प लगभग हर Android Project में काम आते हैं।

Beginner Advice: Android App Development में सबसे अधिक उपयोग SDK Manager, Device Manager, Firebase और Analyze APK का होता है। इन चार विकल्पों को अच्छी तरह समझ लेने पर आप App Development, Testing और Debugging का अधिकांश कार्य आसानी से कर सकेंगे।

VCS Menu



Enable Version Control
Integration

VCS Operations Popup...



Apply Patch...

Apply Patch from Clipboard...

Get from Version Control...



Browse VCS Repository

Create Mercurial
Repository



Import into Subversion...

Share Project (Subversion)...

Create Git Repository...



Share Project
on GitLab



Share Project
on GitHub



अध्याय 10: Android Studio का VCS (Version Control System) Menu (Beginner Tutorial)

VCS (Version Control System) Menu का उपयोग Project के सभी बदलावों (Changes) को सुरक्षित रखने, उनकी History देखने तथा Team के साथ मिलकर Project पर कार्य करने के लिए किया जाता है। Android Studio में VCS के माध्यम से आप Git, GitHub, GitLab, Subversion (SVN) आदि Version Control Systems का उपयोग कर सकते हैं।

यदि आप Professional Android Developer बनना चाहते हैं, तो VCS सीखना बहुत आवश्यक है।

Enable Version Control Integration

उपयोग: Project में Version Control (जैसे Git) को सक्रिय (Enable) करने के लिए।

उदाहरण: नया Project बनाने के बाद Git Repository शुरू करने के लिए।

फायदा: Project के सभी Changes सुरक्षित रहते हैं।

VCS Operations Popup...

उपयोग: Version Control से संबंधित सभी प्रमुख Commands एक ही Popup Window में दिखाता है।

उदाहरण:

- Commit
- Push
- Pull
- Branch
- History

Apply Patch...

उपयोग: किसी Patch File (.patch) को Project में लागू करने के लिए।

उदाहरण: यदि किसी Developer ने Patch File भेजी हो, तो उसे सीधे Project में जोड़ सकते हैं।

Apply Patch from Clipboard...

उपयोग: Clipboard में Copy किए गए Patch को सीधे Project में लागू करता है।

Get from Version Control...

उपयोग: GitHub, GitLab या किसी Remote Repository से Project Download (Clone) करने के लिए।

उदाहरण: GitHub Repository का URL देकर पूरा Project अपने Computer में प्राप्त कर सकते हैं।

Browse VCS Repository

उपयोग: Remote Repository की Files और Branches देखने के लिए।

Create Mercurial Repository

उपयोग: Project के लिए Mercurial (Hg) Repository बनाता है।

नोट: वर्तमान समय में Android Development में इसका उपयोग बहुत कम होता है।

Import into Subversion...

उपयोग: Project को Subversion (SVN) Repository में Import करने के लिए।

Share Project (Subversion)...

उपयोग: Project को SVN Server पर Share करने के लिए।

Create Git Repository...

उपयोग: वर्तमान Project में नई Git Repository बनाता है।

फायदा: इसके बाद आप Commit, Push, Pull और Branch जैसी सुविधाओं का उपयोग कर सकते हैं।

Share Project on GitLab

उपयोग: Project को सीधे GitLab पर Upload (Share) करने के लिए।

फायदा: Team के साथ मिलकर Project पर कार्य करना आसान हो जाता है।

Share Project on GitHub

उपयोग: Project को सीधे GitHub Repository पर Upload करने के लिए।

उदाहरण: अपना Android Project GitHub पर Publish करना।

Version Control के महत्वपूर्ण शब्द

Repository (Repo)

Project को सुरक्षित रखने का स्थान।

Commit

Project में किए गए बदलावों को Save करना।

Push

अपने Computer से GitHub या GitLab पर Changes Upload करना।

Pull

GitHub या GitLab से नए Changes अपने Computer में Download करना।

Clone

Remote Repository की पूरी Copy अपने Computer पर Download करना।

Branch

मुख्य Project को प्रभावित किए बिना नई Feature पर काम करने के लिए अलग Branch बनाना।

Merge

दो Branches के Changes को एक साथ जोड़ना।

Beginner Tips

Android Studio में VCS Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

- *Enable Version Control Integration*
- *Create Git Repository*

- *Get from Version Control*
- *Share Project on GitHub*
- *Share Project on GitLab*
- *VCS Operations Popup*

GitHub Workflow (सरल उदाहरण)

1. नया Android Project बनाएँ।
2. Enable Version Control Integration चुनकर Git Enable करें।
3. Create Git Repository बनाएँ।
4. पहली बार Commit करें।
5. Share Project on GitHub चुनें।
6. Project GitHub पर Upload हो जाएगा।
7. भविष्य में बदलाव करने पर Commit और Push करें।
8. यदि Team Member ने बदलाव किए हैं, तो Pull करके उन्हें अपने Project में प्राप्त करें।

महत्वपूर्ण नोट

यदि आप Android App Development में नौकरी (Job), Freelancing या Open Source Projects पर काम करना चाहते हैं, तो Git और GitHub सीखना अनिवार्य है। लगभग सभी Software Companies Version Control System का उपयोग करती हैं।

Beginner Advice: शुरुआत में Enable Version Control Integration, Create Git Repository, Get from Version Control और Share Project on GitHub का अभ्यास करें। एक बार Git का मूल कार्य समझ में आ जाए, तो Team के साथ Project पर काम करना, Code Backup रखना और पुराने Changes को वापस लाना बहुत आसान हो जाता है।

Windows Menu

 Main Window Control
Integration
Main Window Actions...

 Layouts
Active Tool Window

 Get from Version Control...
Browse VCS Repository

 Editor Navigation
Editor Tabs



Next Project Window

← Previous Project Window

→ Previous Project Window

✓ My Application

अध्याय 11: Android Studio का Window Menu (Beginner Tutorial)

Window Menu Android Studio की विभिन्न Windows, Tool Panels, Editor Tabs और Project Windows को नियंत्रित (Control) करने के लिए उपयोग किया जाता है। इसकी सहायता से आप IDE (Integrated Development Environment) की कार्य-क्षेत्र (Workspace) को अपनी आवश्यकता के अनुसार व्यवस्थित कर सकते हैं। बड़े Projects पर कार्य करते समय यह Menu आपकी Productivity बढ़ाने में सहायता करता है।

Main Window Control Integration

उपयोग: Android Studio की मुख्य Window (Main Window) को नियंत्रित करने तथा Operating System के साथ उसका समन्वय (Integration) बनाए रखने के लिए।

फायदा: Window का व्यवहार अधिक व्यवस्थित रहता है।

Main Window Actions...

उपयोग: Main Window से संबंधित अतिरिक्त Commands और विकल्प उपलब्ध कराता है।

उदाहरण:

- Window को सक्रिय करना
- Window को Restore करना
- Window का आकार बदलना

Layouts

उपयोग: Android Studio के Workspace Layout को बदलने, Save करने या Restore करने के लिए।

फायदा: यदि आपने अपनी पसंद के अनुसार Panels व्यवस्थित किए हैं, तो उन्हें भविष्य के लिए सुरक्षित रख सकते हैं।

Active Tool Window

उपयोग: वर्तमान में खुली हुई Tool Window (जैसे Project, Logcat, Terminal, Build आदि) को नियंत्रित करने के लिए।

उदाहरण: यदि Logcat खुला है, तो उसी Window के विकल्प यहाँ दिखाई देंगे।

Get from Version Control...

उपयोग: GitHub, GitLab या अन्य Version Control Repository से Project डाउनलोड (Clone) करने के लिए।

उदाहरण: किसी GitHub Repository का URL देकर पूरा Project अपने Computer में प्राप्त कर सकते हैं।

Browse VCS Repository

उपयोग: Remote Version Control Repository की Files, Branches और History देखने के लिए।

Editor Navigation

उपयोग: खुले हुए Editor में विभिन्न Files और Tabs के बीच आसानी से Navigation करने के लिए।

फायदा: कई Files पर एक साथ कार्य करते समय तेजी से Switch कर सकते हैं।

Editor Tabs

उपयोग: Editor Tabs की स्थिति (Position), क्रम (Order), दृश्य (Appearance) तथा व्यवहार (Behavior) को नियंत्रित करने के लिए।

उदाहरण:

- Tabs ऊपर या नीचे दिखाना।
- Tabs बंद करना।
- Tabs का क्रम बदलना।

Next Project Window

उपयोग: यदि Android Studio में एक से अधिक Projects खुले हैं, तो अगले Project Window पर जाने के लिए।

Previous Project Window

उपयोग: पिछले खुले हुए Project Window पर वापस जाने के लिए।

Next Project Window (यदि सूची में दोबारा दिखाई दे)

उपयोग: अगले खुले हुए Project पर Switch करने के लिए।

नोट: कुछ Android Studio Versions में यह विकल्प अलग क्रम में दिखाई दे सकता है।

My Application

उपयोग: वर्तमान में खुले हुए Project या Application Window का नाम दिखाता है।

उदाहरण: यदि आपके Project का नाम My Application है, तो उस Project पर सीधे Switch करने के लिए इस विकल्प का उपयोग किया जा सकता है।

Beginner Tips

Android Studio में Window Menu के सबसे अधिक उपयोग होने वाले विकल्प हैं:

- *Layouts*
- *Active Tool Window*
- *Editor Navigation*
- *Editor Tabs*
- *Next Project Window*
- *Previous Project Window*

Window Menu से संबंधित महत्वपूर्ण Tool Windows

Android Studio में सामान्यतः निम्न Tool Windows का उपयोग सबसे अधिक किया जाता है:

- Project – Project की सभी Files और Folders देखने के लिए।
- Logcat – App के Logs और Errors देखने के लिए।
- Terminal – Gradle एवं Git Commands चलाने के लिए।

- Build – Build Process की जानकारी देखने के लिए।
- Run – App Run होने की स्थिति देखने के लिए।
- Debug – Debugging के दौरान Variables और Breakpoints देखने के लिए.
- Profiler – CPU, Memory और Network Usage का विश्लेषण करने के लिए।

महत्वपूर्ण नोट

यदि आप एक ही समय में कई Android Projects पर कार्य करते हैं, तो Window Menu आपको उनके बीच आसानी से Switch करने तथा Workspace को व्यवस्थित रखने में सहायता करता है।

Beginner Advice: शुरुआत में आपको Editor Tabs, Layouts और Active Tool Window का उपयोग अच्छी तरह सीखना चाहिए। जैसे-जैसे आप बड़े Android Projects पर कार्य करेंगे, Window Menu आपके लिए Workspace Management का एक महत्वपूर्ण उपकरण बन जाएगा।

Help & Diagnostic Chronicle



Find Action... Ctrl+Shift+A

Android Studio Help

IntelliJ IDEA Help

What's New in Android Studio

Getting Started

Android Studio on YouTube

Licenses

Keyboard Shortcuts PDF

Submit a Bug Report...

Show Log in Explorer

Collect Logs and Diagnostic Data

Analyze Memory Usage



Capture Memory Snapshot

Diagnostic Tools



Change Memory Settings

Edit Custom Properties...

Edit Custom VM Options...

Delete Leftover IDE Directories...

Check for Updates...

About

Chapter: 12 Help & Diagnostic Chronicle (Help Menu)

Introduction

Android Studio का Help मेनू डेवलपर्स के लिए सहायता, दस्तावेज़ (Documentation), अपडेट, लॉग, डायग्नोस्टिक टूल्स और सिस्टम सेटिंग्स तक पहुँच प्रदान करता है। यदि आपको किसी समस्या का समाधान चाहिए, IDE की परफॉर्मेंस जाँचनी है या नवीनतम अपडेट इंस्टॉल करना है, तो Help Menu सबसे महत्वपूर्ण स्थान है।

चित्र में Help Menu को एक प्राचीन पुस्तक (Chronicle) की तरह डिज़ाइन किया गया है, जो आधुनिक सॉफ्टवेयर डेवलपमेंट और मध्यकालीन शैली का सुंदर मिश्रण प्रस्तुत करता है।

1. Find Action... (Ctrl + Shift + A)

कार्य:

Android Studio की किसी भी कमांड, सेटिंग, विंडो या फीचर को तुरंत खोजने के लिए।

उपयोग:

- किसी भी कमांड को तुरंत ढूँढना।
- सेटिंग्स खोलना।
- टूल विंडो खोलना।
- Plugin खोजने के लिए।

शॉर्टकट:

Ctrl + Shift + A

2. Android Studio Help

यह विकल्प Android Studio की आधिकारिक सहायता (Documentation) खोलता है।

यहाँ आपको मिलता है:

- Installation Guide
- Project Setup
- Layout Design
- Emulator Guide
- Debugging
- APK Build
- App Publishing

यह Beginner से लेकर Advanced Developer तक सभी के लिए उपयोगी है।

3. IntelliJ IDEA Help

Android Studio, IntelliJ IDEA पर आधारित है।

इस विकल्प में IntelliJ की Documentation मिलती है, जिसमें शामिल हैं:

- Editor Features

- Code Navigation
- Refactoring
- Version Control
- Plugins
- Productivity Tips

4. What's New in Android Studio

यह विकल्प वर्तमान संस्करण में जोड़े गए नए Features दिखाता है।

इसमें जानकारी मिलती है:

- नए Tools
- Bug Fixes
- Performance Improvements
- UI Changes
- AI Features
- New Emulator Features

5. Getting Started

यह नए उपयोगकर्ताओं के लिए शुरुआती मार्गदर्शिका है।

इसमें शामिल है:

- पहला Project बनाना
- Emulator चलाना
- App Run करना
- Gradle Build
- Layout बनाना
- Kotlin और Java का परिचय

6. Android Studio on YouTube

Google के आधिकारिक YouTube चैनल पर Android Studio से संबंधित वीडियो Tutorials उपलब्ध होते हैं।

इनमें शामिल हैं:

- Beginner Tutorials
- UI Design
- Jetpack Compose
- Firebase
- Performance

- Debugging
- Material Design

7. Licenses

यह विकल्प Android Studio में प्रयुक्त Open Source Libraries एवं उनके License Agreements दिखाता है।

यह मुख्यतः कानूनी जानकारी के लिए उपयोगी है।

8. Keyboard Shortcuts PDF

यह Android Studio के सभी Keyboard Shortcuts का PDF उपलब्ध कराता है।

उदाहरण:

- Copy
- Paste
- Rename
- Search
- Refactor
- Run
- Debug

शॉर्टकट सीखने से कार्य की गति कई गुना बढ़ जाती है।

9. Submit a Bug Report

यदि Android Studio में कोई समस्या आती है तो आप सीधे Google को Bug Report भेज सकते हैं।

Bug Report में शामिल हो सकते हैं:

- Error Logs
- Crash Details
- Screenshots
- System Information

10. Show Log in Explorer

Android Studio द्वारा बनाए गए Log Files खोलता है।

इन Logs में दर्ज होता है:

- Errors
- Warnings
- Gradle Output
- IDE Messages

Debugging के दौरान यह विकल्प बहुत उपयोगी है।

11. Collect Logs and Diagnostic Data

यह विकल्प सभी आवश्यक Diagnostic Files एकत्र करता है।

इनमें शामिल होते हैं:

- IDE Logs
- Crash Reports
- JVM Information
- Memory Information
- Configuration Files

इनका उपयोग Technical Support के लिए किया जाता है।

12. Analyze Memory Usage

Android Studio की Memory खपत का विश्लेषण करता है।

यह दिखाता है:

- RAM उपयोग
- Memory Leak
- Object Allocation
- Garbage Collection

यदि IDE धीमा चल रहा हो, तो यह विकल्प समस्या पहचानने में मदद करता है।

13. Capture Memory Snapshot

यह JVM Heap का Snapshot बनाता है।

Heap Snapshot से पता चलता है:

- कौन-सी Objects सबसे अधिक Memory ले रही हैं।
- Memory Leak कहाँ है।
- Garbage Collection की स्थिति क्या है।

यह Advanced Debugging के लिए उपयोगी है।

14. Diagnostic Tools

इसमें विभिन्न Diagnostic Utilities उपलब्ध होती हैं।

इनका उपयोग किया जाता है:

- Performance Monitoring
- Thread Analysis
- Internal Debugging
- Cache Inspection

15. Change Memory Settings

यदि Android Studio धीमा चल रहा है, तो यहाँ से IDE की Memory बढ़ाई या घटाई जा सकती है।

उदाहरण:

- Heap Size बढ़ाना
- RAM Allocation बदलना
- Performance सुधारना

16. Edit Custom Properties

यह Android Studio की विशेष Configuration Properties संपादित करने के लिए उपयोग किया जाता है।

Advanced Users इसका उपयोग IDE के व्यवहार को अनुकूलित (Customize) करने के लिए करते हैं।

17. Edit Custom VM Options

Java Virtual Machine (JVM) के विकल्प संपादित किए जाते हैं।

उदाहरण:

- Heap Size
- Garbage Collector
- JVM Arguments
- Performance Optimization

गलत परिवर्तन IDE को अस्थिर बना सकते हैं, इसलिए सावधानी आवश्यक है।

18. Delete Leftover IDE Directories

पुराने Android Studio Versions द्वारा छोड़ी गई अनावश्यक Files और Directories हटाता है।

इससे:

- Disk Space खाली होता है।
- पुराने Configuration हटते हैं।
- IDE साफ़ और व्यवस्थित रहता है।

19. Check for Updates

Android Studio के नवीनतम संस्करण की जाँच करता है।

यदि नया संस्करण उपलब्ध हो, तो आप:

- Download
- Install
- Restart

कर सकते हैं।

अपडेट करने से नए Features, Security Fixes और Bug Fixes प्राप्त होते हैं।

20. About

यह Android Studio की जानकारी प्रदर्शित करता है।

इसमें शामिल हैं:

- Version Number
- Build Number
- Runtime
- JVM Version
- Operating System
- Installed Plugins

तकनीकी सहायता लेते समय यह जानकारी अक्सर आवश्यक होती है।

सारांश

Help Menu Android Studio का सहायता एवं रखरखाव केंद्र है। यहाँ Documentation, Tutorials, Updates, Diagnostic Tools, Memory Analysis, Log Files और System Configuration जैसी सभी महत्वपूर्ण सुविधाएँ एक ही स्थान पर उपलब्ध हैं। किसी भी Android Developer के लिए Help Menu का ज्ञान आवश्यक है, क्योंकि यह समस्याओं के समाधान, IDE की Performance सुधारने और नवीनतम सुविधाओं का लाभ उठाने में महत्वपूर्ण भूमिका निभाता है।