

演習資料(2) Android Studio でのアプリ開発

発展学習について

本資料は、勉強会資料の側面として、 画面上部のタイトル左端に 響 を記した 発展学習ページも設けています。

演習作業の合間に 余裕があれば、参照ください。



Android Studio での機能別ウィンドウやバーの一般的な呼称(一般名称) Android Studio を大勢の人と使う上で、意思疎通がしやすいようバーやウィンドウの一般名を示します。



ここで明示した各部位の名称は、一般的な用法に合わせていますが、本勉強会用の名称(用語)です。



Android Studio ナビゲーション・ツールバーの機能ボタン概要(1)

□ H Ø < > X □ □ < Q Q < + H P = > X ↓ P = < A + ? Q





Android Studio ナビゲーション・ツールバーの機能ボタン概要(2)







Android Studio のツールウィンドウ開閉

- メニューバーから開閉する
 メニューバー > View > Tool Windows で 指定のツールウィンドウが開閉します。
- ツールウィンドウ・バーから開閉する
 ツールウィンドウ・バーのツールウィンドウ・ボタンで、
 指定のツールウィンドウが開閉します。





Android Studio のツールウィンドウ開閉

 ツールウィンドウ・メニューから開く ステータス・バー左端の
 アイコンの マウスオーバ・メニューから、 指定したツールウィンドウを開きます。

> 上記アイコンをクリックにすると、 ウィンドウツールバーが開閉します。

4. Hide アイコンで閉じる ツールウィンドウ右上の - アイコンの クリックでツールウィンドウが閉じます。

マウスオーバーで、

メニューが開きます





Android Studio でのアプリ開発

演習1)新規プロジェクトを作ってみよう

演習1)の目的 アプリ開発 信語習得)の第一歩として、Hello World アプリ作成があります。 この演習では、自分で新規プロジェクト (Hello Worldアプリ)を作成してもらい、 エミュレーターで実行してもらうことで、アプリ開発の第一歩を踏み出してもらいます。

Hello World アプリの作成(新規プロジェクトの作成)

- プロジェクト新規作成の実行 Android Studio の起動画面の Quick Start から、New Project をクリックします。
- 新規プロジェクトの基本設定(1) Create New Projcet ウィンドウが開くので、 1 Application Name を HelloWorldApplication に変更し、
 2 Company Domain を helloworld.example.com に変更し、
 - 3. Project location を

C:¥Users¥ユーザ名¥AndroideStudioProjects ¥HelloWorldApplication (一行)に設定し、
※AndroideStudioProjects は、Android Studioの
※標準のプロジェクトディレクトリ名です。
4. Next | をクリックして次に進みます。



- 3. 新規プロジェクトの基本設定 (2)
 - 1. Phone and Tablet の Minimum SDK を 勉強会の対象最小Androidバージョンとなる API 15 Android 4.0.3 ICS) に変更し
 - 2 Next」をクリックして次に進みます。

😭 Create New Project		.
Select the form factors your app w	ill run on	
Different platforms require separate SDKs		
Phone and Tablet		
Minimum SDK	API 15: Android 4.0.3 (IceCreamSandwich)	
	Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately 84.3% of the devices that are active on the Google Play	
	Store. Help me choose.	
TV		
Minimum SDK		
🗌 Wear		
Minimum SDK	API 20: Android 4.4 (KitKat Wear)	
Glass (Not Installed)		
Minimum SDK		
	Previous Next Cance	Finish

- 新規プロジェクトの基本設定(3)
 1 Activity 表現タイプを Blank Activity with Fragment に指定し、
 2. Next」をクリックして次に進みます。
- 5. 新規プロジェクトの基本設定 (4) 1 Activity Name を HelloWorldActivity に変更し
 - 2 Layout Name を activity_hello_world に変更し
 - 3. Fragment Layout Name を fragment_hello_world に変更し
 - 4. Title を
 - HelloWorldActivity に変更し
 - 5. Next」をクリックして次に進みます。



- 新規プロジェクト生成 以下のような流れで、基本設定に従った Hello World プロジェクトが生成されます。
 - 1. Gradle Build プログレスが表示され Hello World プロジェクトの新規生成が 始まり、
 - 3 新規生成した Hello World プロジェクトを オープンした Android Studio が開きます。 (この例では、Tip ダイアログを伴っています。)





Hello World アプリの内容確認 発展学習として、生成されたプロジェクト 内容を確認してみましょう。





Hello World アプリの内容確認

 app/main/java/ com.example.helloworld. Helloworldapplication/ HelloWorldActivity.java Activty処理内容の定義 (画面処理内容の定義)





Hello World アプリの内容確認

2. app/main/res/layout/ activity_hello_world.xml Activity画面内容の定義 画面全体の表示内容定義

- 🚔 I	🚔 HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [app]¥app¥src¥main¥res¥layout¥activity_hello_world.xml - And 👝 💼 💽					
Eile	e <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analyze <u>R</u> efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S W</u> indow <u>H</u> elp					
	🚉 HelloWorldApplication) 😋 app े 💼 sc 🔪 🛅 main 🔪 📬 res 👌 🛅 layout 🤇 🧕 activity_hello_world.xml 🤇					
Ţ	Project ▼ ③ ≑ ♣ ► I ⓒ HelloWorldActivity.java × i @ activity_hello_world.xml × i @ fragment_hello_world.xml × i @ activity_hello_world.xml ×	p x -≡3 ⊙				
ure 🚺 <u>1</u> : Proj	 HelloWorldApplication (C:¥Us) CraneLayout xmins: android="http://schemas.android.com/tools" xmins: tools="http://schemas.android.com/tools" android: id="&tid/container" android: layout_width="match_parent" android: layout_neight="match_parent" 	Gradle 👘 Prev				
Struc	tools: ignore="MergeRootFrame" />	ew				
s :2: :	<pre>> androidTest > main</pre>	E Maven Projects ທ Command				
ild Variants	 drawabie-xhipi drawabie-xxhdp arawabie-xxhdp arawabie-xxhdp 	q				
Bu	activity_hell					
🗰 2: Favorites 🧯	imit fragment_he imi					
_	🔄 Terminal 📃 <u>0</u> : Messages 🍦 <u>6</u> : Android 🍃 <u>4</u> : Run 🧐 TODO 📃 Event Log	🗐 Gradle Console				
	8:1 CR	.LF ≑ UTF-8 🚡 🚭				



Hello World アプリの内容確認

3. app/main/res/layout/ fragment_hello_world.xml] Fragmentによる画面内容の定義 画面表示内容パーツの定義 (Activity画面内容の部分参照パーツ)

File	🚔 HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [app]¥app¥src¥main¥res¥layout¥fragment_hello_world.xml - A 👝 💷 💽				
	HelloWorldApplication 🖯 🗖 app 👌 🖿 sr	c > 🛅 main > 🛅 res > 🛅 layout > 🙍 fragment_hello_world.xml >			
ţ	🗊 Project 🔻 😳 幸 🕸 🗠 🗠	properties x 📄 gradlew x 📄 gradlew.bat x 📑 gradle-wrapper.properties x 🖻 fragment_hello_world.xml x 🖃	6 💽		
1: Pro	 HelloWorldApplication (C:¥Use idea 	© © RelativeLayout xmIns:android="http://schemas.android.com/apk/res/android" xmIns:tools="http://schemas.android.com/tools" android:leyend:id="schemas.android.com/tools"	Gradle		
	app	android:layout_width match_parent android:layout_height="match_parent"	te de la composición de la com		
ucture	ibs	android:paddingLeft= 640p android:paddingRight="640p"	review		
Z: Str	🔻 🛅 src	android:padding:lop-ibdp android:padding:Bottom="16dp"			
۷	 androidTest main 	tools:context=".HelloWorldActivity\$PlaceholderFragment">	Mave		
	▶ 🛅 java	<pre>viextview android:text="Hello world!" </pre>	n Proj		
	 Le res Tawable-hdpi 	android: layout_width-wrap_content android: layout_height="wrap_content" />	ß		
	▶ 🛅 drawable-mdpi	⊖	2		
	Interpretation of the second secon		mmo		
ية ا	▼ 🖻 layout		ander		
Variar	activity_hell				
blind	o fragment_he				
	values				
ß	talues-w820dp				
Favor	AndroidManifest.xn				
ivi *	J app.iml	Design Text			
	🔄 Terminal 📃 <u>0</u> : Messages 🛛 🏺 <u>6</u> : An	droid 🕨 4: Run 🧐 TODO 📃 Event Log 🔳 Gradle Console	:		
		17:1 CRLF \$ UTF-8 @	, ⊜		



Hello World アプリの内容確認

4. app/main/res/values/ dimens.xml 定数リソースの定義

	alloPracticeApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloPracticeApplication] - [app]¥app¥src¥main¥res¥values¥dimens.xml - Android Stu 🔤 💷 📧				
Eile	e <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analyze <u>R</u> efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp				
	」 ዘ ∅ ৺ ҂ 从 ြ 凶 Q ♀ → ዛ 傳 ┉ ▼ ▶ 微 및 9 風 \$ ● ?				
E					
ŧ	+ D Project + O + + + + he synt x C HelloPracticeActivity java x G fragment hello practice xml x G dimens xml x -==				
ojec	C HelloPracticeApplication (C:¥Users¥rie¥ApdroidStudioProtects¥HelloPracticeApplication)	= 8 ■ 8			
E H	Comparison of the second	m i			
-	dimen name="activity_horizontal_margin">18dc/dimen>	ande			
ø	dimension activity_vertical_margin >16dpc/dimen> build 5 6/resources>	4			
ictur	Dibs 6 di	A			
Str	V 🗅 src	100 1			
N V	androidTest	ierar			
•	🔻 🛅 main	chy			
	🕨 🛅 java				
	v 🛅 res	M			
	▶ 🖻 drawable-hdpi	aven			
	▶ 🖻 drawable-mdpi	Proj			
	▶ 🔁 drawable-xhdpi	R			
	▶ 🛅 drawable-xxhdpi	0			
she	Verification and the second se	0 0			
Varia	etivity_hello_practice.xml	adle			
pin	in fragment_hello_practice.xml				
	▶ menu				
	V E Values				
orites	Contractions and Contra				
Favo					
ini A	► Traines-w820dp				
-					
_	Q 3: Find Sector Find Topo + 6: Android Terminal Event Log Carale Consol	e			
	6:1 CRLF \$ OTHER	₽∰			



Hello World アプリの内容確認

5. app/main/res/values/ strings.xml stringリソースの定義

	HelloPracticeApplication - [C:¥Users¥rie¥AndroidStudioProjec	.cts¥HelloPracticeApplication] - [app]¥app¥src¥main¥res¥values¥strings.xml - Android Stu 👝 🔳 🛋		
Eile	<u>Edit View Navigate Code Analyze Refactor Build Run Iools VCS Window Help</u>			
	🗎 🕼 🔶 🔶 🚺 🛍 🔍 🐥 🔶 👫 🚺	🖡 app 🗸 🕨 🐘 🦞 🖽 🗣 🛄 🧯 📫 📍		
E	HelloPracticeApplication	a res E values i valu		
t	Project ▼ ⊕ ‡ ∯* I**	🔯 strings.xml × 🤄 HelloPracticeActivity.java × 🔯 fragment_hello_practice.xml × 📲 з		
Proj	HelloPracticeApplication (C:¥Users¥rie¥AndroidS	1 xml version="1.0" encoding="utf-8"?		
	🕨 🛅 .idea	2 G(resources) 3		
	🔻 🛅 app	4 <pre><string name="app_name">HelloPracticeApplication</string></pre>		
ane	build	5 <string name="hello_world">Hello World!</string>		
ruct	🛅 libs	7		
S.	🔻 🛅 src	8 Avresources>		
7	androidTest	3		
	🔻 🛅 main			
	java			
	🔻 📑 res			
	drawable-hdpi			
	drawable-mdpi			
	drawable-xhdpi			
	drawable-xxhdpi			
ants	V Dayout			
Vari	activity_hello_practice.xml			
. plin	in fragment_hello_practice.xml			
- -	▶ ⊡ menu			
	values			
1 Se	o dimens.xmi			
Favo	ostrings.xmi			
in i	styles.xmi			
*	Values-w820ap			
	🔍 <u>3</u> : Find 🧐 TODO 🌞 <u>6</u> : Android 🛛 🗔 Terminal	Event Log 🗐 Gradie Console		
		9:1 CRLF \$ UTF-8 & §		



Hello World アプリの内容確認

6. app/main/res/values/ styles.xml styleリソースの定義

	/loPracticeApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloPracticeApplication] - [app]¥app¥src¥main¥res¥values¥styles.xml - Android Stud 🧫 💷 🗾					
Eile	e <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e <u>R</u> efactor <u>B</u> uild	R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp				
	⊐					
	HelloPracticeApplication) 🕞 app) 💼 src) 💼 main) 📴 res) 💼 values) 💀 styles.xml)					
t	Project ▼ ③ ≑ 鼻- I+	oPracticeActivity.java × 🙆 fragment_hello_practice.xml × 🙆 dimens.xml × 🗟 styles.xml × 🖃	•			
Proje	HelloPracticeApplication (C:¥Users¥rie¥AndroidS	1 Okresources>	8			
÷.	🕨 🛅 .idea	2	nma			
=	🔻 🛅 app	3 Base application theme 4 □ <style dark&ctionbar"="" holo="" light="" name="#antheme" nament="android" theme=""></th><th>Inde</th></tr><tr><th>ų</th><th>build</th><th>5 <! Customize your theme here></th><th></th></tr><tr><th>nct n</th><td>🛅 libs</td><td>6 A </style> <td>Å</td>	Å			
Str	🔻 🛅 src	/ 8 ⊖	E Hi			
	androidTest	9	eraro			
	🔻 🛅 main		Ξhγ			
	▶ 🛅 java		m			
	🔻 📑 res		Ma			
	drawable-hdpi		Ven			
	🕨 🛅 drawable-mdpi		Proj			
	drawable-xhdpi		<u>B</u>			
	Irawable-xxhdpi		0			
ants	V Dayout		<u>م</u>			
Variè	activity_hello_practice.xml		adle			
dild	fragment_hello_practice.xml					
	menu					
	dimons yml					
orites	strings yml					
Favo	styles.xml					
ii	▶ values-w820dp					
_						
_	🔍 <u>3</u> : Find 🦓 TODO 🏺 <u>6</u> : Android 💿 Terminal	Event Log 🔄 Gradie Console	0			
		9:1 CRLF \$ UTF-8 %	鼓			



Hello World アプリの内容確認

7. app/main/ AndroidManifest.xml アプリ全体設定

	A HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [app]¥app¥src¥main¥AndroidManifest.xml - Android Studio (Bet 👝 💷 🔀				
	HelloWorldApplication \square app \square	src) main) Android Manifest.xml)			
oject	Project ▼ ⊕ 幸 ☆	ment_hello_world.xml × ③ app × ③ HelloWorldApplication × ③ settings.gradle × ④ AndroidMa	unifest.xml × ▼≣3 ()		
10 1: Pr	 Interview in application (C.+03) idea Image: Image in the image in the	<pre><xml <br="" encoding="utra" version="1.0">G(manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.helloworld.helloworldapplication" ></xml></pre>	aide m		
📢 Z: Structure	 build libs src androidTest main java res 	<pre></pre>	Maven Projects 🛍 Comma		
: Favorites 👘 Build Variants	AndroidNanirest.XX AddroidNanirest.XX Gitignore Japp.iml build.gradle proguard-rules.pro gradle gradle gradle gradle gradle.properties gradle.properties gradlew gradlew gradlew	<pre></pre>	nder		
*	local.properties				
	i Terminal <u>■ 0</u> : Messages 📫 <u>6</u> : A	ndroid 🕨 <u>4</u> : Run 🧐 TODO Event Log 11:47 CR	I Gradle Console ILF ≑ UTF-8 ≧ ∰		



Hello World アプリの内容確認

8. app/ app.iml *Android Studio*の基となった *IntelliJの app*モジュール設定

	HelloWorldApplication - [C:¥Users¥rie¥And	oidStudioProjects¥HelloWorldApplication] - [app]¥app¥app.iml - Android Studio (Beta) 0.8.9
E	le <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e	<u>R</u> efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp
E) 🖶 💋 🛹 🥕 🔏 🛍 🔍 🔍	
C	🖫 HelloWorldApplication 👌 📴 app 👌 🗊 ap	ıp.iml >
t	🗊 Project 👻 😨 🕸 🔭 🕪	C HelloWorldActivity.java × J app.iml × 🔽 activity_hello_world.xml × 🐻 fragment_hello_world.xml × -≡₄ (3
roie	HelloWorldApplication (C:¥Users)	(?xml version="1_0" encoding="IITE-8"?)
<u>م</u>	idea	<pre>(module external.linked.project.path="\$MODULE_DIR\$" external.root.project.path="\$MODULE_DIR\$/" ext</pre>
-		<pre><component name="FacetManager"></component></pre>
		<pre><facet name="Android-Gradle" type="android-gradle"></facet></pre>
ture	Dulla	<pre><cominguration <br="">dotion_name="GRADIE_PROJECT_PATH" value=":app" /></cominguration></pre>
truc	libs	<pre></pre>
0	V C Src	⟨/facet>
	Interview Int	<facet name="Android" type="android"></facet>
	🕨 🛅 main	<pre><contiguration <="" <contiguration="" pre=""></contiguration></pre>
	.gitignore	Coption name="statemptotic_botto_botto_bito;"
	🗾 app.iml	<pre><pre></pre> </pre> <pre></pre> <pre><</pre>
	build.gradle	<pre>coption name="ASSEMBLE_TEST_TASK_NAME" value="assembleDebugTest" /></pre>
	proguard-rules.pro	<pre><option <="" name="SUURUE_LEN_LASK_NAME" th="" value="generateDebugSources"></option></pre>
	▶ □ gradle	<pre>coption name="file" file" configuration while value="file" false" /> coption name="file" fullow USE CONFIGURATION" value="file" false" /> </pre>
		<pre><option name="MANIFEST_FILE_RELATIVE_PATH" value="/src/main/AndroidManifest.xml"></option></pre>
	 gldghore build and b 	<pre><option name="RES_FOLDER_RELATIVE_PATH" value="/src/main/res"></option></pre>
- the	C Dulid.gradie	<pre><option name="RES_FOLDERS_RELATIVE_PATH" value="file://\$MOULE_DIR%/src/main/res"></option></pre>
Vari	gradle.properties	<pre><option <="" name="ASSEIS_FULDER_KELAIIVE_FAIH" pre="" value="/src/main/assets"></option></pre>
P	gradlew	<pre></pre>
đ	gradlew.bat	
1	HelloWorldApplication.iml	<pre><component inherit-compiler-output="false" name="NewModuleRootManager"></component></pre>
8	local.properties	<pre><output url="file://\$MUDULE_DIK\$/build/intermediates/classes/debug"></output> <pre></pre></pre>
orit	settings.gradle	<pre><core de="" de<="" due="" th=""></core></pre>
Fa	External Libraries	<sourcefolder gen<="" istestsource="false" th="" url="file://\$MODULE_DIR\$/build/generated/source/r/debug"></sourcefolder>
Ś		<pre><sourcefolder <="" istestsource="false" pre="" url="file://\$MODULE_DIR\$/build/generated/source/aidl/debug"></sourcefolder></pre>
		<pre>{sourcetolder.url="tile"//SMUDILE_DIRS/build/generated/source/buildConfig/debug"_isTestSource=</pre>
	👿 Terminal 🛛 🧮 <u>0</u> : Messages 🛛 🏺 <u>6</u> : An	droid 🕨 🛓 : Run 🧌 TODO 📃 Event Log 🔳 Gradle Console
]	1:1 CRLF 🗧 UTF-8 🍙 🔂



Hello World アプリの内容確認

9. app/ build.gradle *Android Studio* ビルドシステムである *Gradleの app*モジュール・ビルド設定

	HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [app]¥app¥build.gradle - Android Studio (Beta) 0.8.9				
Eile	<u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e <u>R</u>	efactor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S W</u> indow <u>H</u> elp			
	₩ 💋 🛹 🔶 🛱 🖄 🔍 🔍 <	💠 🕂 👫 🗰 app 💌 🕨 🌺 🕵 🐓 💷 🗳 👫 ڭ 🐐 🥐	Q,		
	HelloWorldApplication 🗁 📴 build.gradle 👌				
ţ	🗊 Project 🔹 😌 崇 🌾 🚺	🕻 HelloWorldActivity.java x 🙍 activity_hello_world.xml x 🙍 fragment_hello_world.xml x 📀 app x	≡3	۲	
Proje	HelloWorldApplication (C:¥Users	apply plugin: 'com.android.application'		Grao	
÷	🕨 🛅 .idea	charles in I		1	
-	🔻 🛅 арр	compileSdkVersion 19		m	
<u>و</u>	build	buildToolsVersion "20.0.0"		Ma	
nct.	🛅 libs	e defaultConfig		èn	
ts.	🔻 🛅 src	minSdkVersion 15		Proj	
	androidTest	targetSdkVersion 19	_	<u>Å</u>	
	🕨 🛅 main	versionCode 1			
	.gitignore			R R	
	J app.iml	DuildTypes {		mn	
	📀 build.gradle	release {		land	
	proguard-rules.pro	proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'		9	
	gradle				
	.gitignore	productFlavors {			
뉟	📀 build.gradle				
/aria	📊 gradle.properties				
- Pi	📄 gradlew	⊖dependencies {			
Bu	gradlew.bat	<pre>compile fileTree(dir: 'libs', include: ['*.jar'])</pre>			
	HelloWorldApplication.iml				
ß	local.properties				
avor	settings.gradle				
in E	External Libraries				
*					
	👿 Terminal 📃 0: Messages 🛛 🌞 6: Andro	pid 🕨 🛓: Run 🌚 TODO 📃 Event Log 🔳 Gradi	e Console		
		26:1 CRLF + UT	F-8 ‡ 🚡	₿	
			_		



Hello World アプリの内容確認

10. gradle/

gradle-wrapper.properties Android Studio ビルドシステムである Gradleのラッパー設定 gradle-wrapper.jarは、ラッパーの実態

Eil	HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [HelloWorldApplication]¥gradle¥wrapper¥gradle-wrapper.prop e Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help	per	x
E	I 🗄 💋 < 🖈 🖧 🗊 🗗 🔍 🙊 💠 👫 🖷 app 🛛 🕨 🗰 🦞 🏭 🗳 🚆 🎽 🌞 🥊	Q,	
E	HelloWorldApplication C gradle C wrapper C gradle-wrapper.properties		
ţ	🗊 Project 🔻 🕹 🛊 👫 🕼 gradle.properties x 🕼 local.properties x 🗿 gradlew x 🗿 gradlew.bat x 🕼 gradle-wrapper.p	properties × 💵	۲
Proj	HelloWorldApplication (C:¥Users ##ed Apr 10 15:27:10 PDT 2013		Grad
=	▶ ☐ .idea distributionbase=tikADLE_USER_HUME distributionPathwrapper/dists	-	ō
	zipStoreBase=GRADLE_USER_HOME	_	m
ture	gradle zipStorePath=wrapper/dists distributions/cradle_1 12-sll_zin	-	Mave
truc		-	en Pr
2: 5	jj gradie-wrapper.jar		ojec
V			দ
	(a) build.oradle		*
	a gradle.properties		Com
	i gradlew		mano
	📄 gradlew.bat		ie,
	J HelloWorldApplication.iml		
	📊 local.properties		
뉟	📀 settings.gradle		
/aria	▶ 💼 External Libraries		
- Piir			
ā			
stic			
Favo			
iii			
	Interminar Intermin	windows-31j	⇔



Hello World アプリの内容確認

11. HelloWorldApplication.iml Android Studio の基となった IntelliJのプロジェクト設定

Eile	HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [HelloWorldApplication]¥HelloWorldApplication.iml - Android S e Edit View Navigate Qode Analyze Befactor Build Run Tools VCS Window Help	Stu 🗖 🗖	×
	I 🗄 💋 < 🖈 🖧 🗊 🗗 🔍 🙊 💠 🕂 🖷 app 🛛 🕨 🗰 🦞 🕵 📽 🗮 🖇 👖	C	٤.
E:	HelloWorldApplication		
4 2: Structure 11 1: Project	Project v @ + + * * C HelloWorldActivity.java x j app.inl x j HelloWorldApplication.(C:\Users ion="1.0" encoding="UIF-8"?>	xml x -= 	 Gradle Region Maven Projects Mi Commander
淋 2: Favorites 🐗 Build Variants		E Grada Consola	
	🔄 Terminal 📃 02: Messages 🍦 62: Android 🕨 42: Run 🧐 TODO 📃 Event Log 1:1 CRL	ਡ Gradle Console _F ≑ UTF-8 ੇ ਜ਼	. 🕀



Hello World アプリの内容確認

12. build.gradle

Android Studio ビルドシステムである Gradleのプロジェクト・ビルド設定

-	HelloworldApplication - [C:#Users#rie#AndroidStudioProjects#HellowondApplication] - [HelloworldApplication]#build.gradle - Android Studio (Beta) 0.8.9 👝 📄 💌			
Eile	e <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analyze <u>R</u> efactor <u>B</u> uild R <u>un</u> <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp	Foir Airward Coop Auguste Coop Auguste Kalarcou Bring Kalu Tools AC2 Muladow Helb		
	I 🖶 Ø 🛹 /> X 🗅 🗗 Q Q 4 → 14 🍓 app 🔪 🕨 🗰 9 🖽 9 🖽 6 🐺 7	Q,		
E	HelloWorldApplication			
ţ	g 🗊 Project 🔹 😌 🖶 🕸 🗈 🖿 ment_hello_world.xml x 💿 app x 💿 HelloWorldApplication x 💿 settings.gradle x 💿 AndroidManif			
Proje	🔻 🕞 HelloWorldApplication (C:#User /// Top-level build file where you can add configuration options common to all sub-projects/modules.	្រា ខ្មែរ		
1	🕨 🛅 .ldea	음		
-	▼ Dialdscript {	m		
ų	▶ build	3		
nctri	□ libs	iven		
£	► SPC	Proj		
14	i .gitignore	8		
•	app.iml // NOTE: Do not place your application dependencies here: they belong // NOTE: the individual working hild working files			
	build.gradle	2		
	■ proguard-rules.pro	om		
	▶ En gradle	land		
	gitignore grepositories {	e.		
	icenter()			
	a gradle.properties □ 1			
뉟	gradlew			
/aria	gradlew.bat			
- Pi	3 HelloWorldApplication.iml			
a ă	Ca local.properties			
样 2: Favorites	Illy External Libraries			
	🔟 Terminal 📃 <u>0</u> : Messages 📫 <u>6</u> : Android 🕨 <u>4</u> : Run 🧐 TODO 📃 Event Log	Gradle Console		
	19:2 C	RLF 🛊 UTF-8 🛊 🚡 🎡		



Hello World アプリの内容確認

13. settings.gradle

Android Studio ビルドシステムである Gradleのプロジェクト・モジュール設定

Eile	e Edit View Navigate Code Analyze Refactor Build Run Tools VCS, Window Help				
	⊨ ∅ ♥ ҂ ¼ ҧ ┓ ♀ ↓ ₩ ♠ ━ ▼ ► ♥ 및 ♥ ⊕ ♀ ₽ ↑	Q,			
E	HelloWorldApplication) 📀 settings.gradle)				
H I Project ▼					
roje	V The HelloWorldApplication (C:VUSer: Include ':app'	୍କ କୁ			
-	▶ □ .ldea	dle			
-	v To app	m			
e	▶ 🖻 build	3			
nctri		Nen			
ts.	▶ È src	Proj			
	.gitignore	r			
	ji app.iml				
	💿 build.gradle	8			
	📄 proguard-rules.pro	mm			
	▶ C gradle	ande			
	i .gitignore	ц			
	() build.gradle				
	a gradie.properties				
ants	gradlew				
Vari	g gradlew.bat				
nild	3) HelloWorldApplication.iml				
 ₩					
	sectings.grade				
orites	P III External Libraries				
Fav					
_		Q			
	🔤 Ierminai 🔤 <u>U</u> : Messages 🖷 <u>5</u> : Android 🕨 <u>4</u> : Run 😭 IODO 👘 Event Log 🔳 Gradie	Console			
	2.1 CKF # 01P				



Hello World アプリの内容確認

14. gradlew

Gradle起動シェルスクリプト(Linux用)

Eile	e <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analyze	Refactor Build Run Tools VCS Window Help
	₩ Ø 🗸 ∻ 🐰 🛍 🗖 🔍 🔍	↓ + + + + + + + + + + + + + + + + + +
C,	HelloWorldApplication	
ţ	🗊 Project 🔻 🕄 幸 🖛	🗊 HelloWorldApplication.iml x 👔 gradle.properties x 👔 local.properties x 🗐 gradlew x 📑 gradlew.bat x 📲
Proje	HelloWorldApplication (C:¥Users)	1:/usr/bin/env bash
ii R	 idea app build 	
Structure	 Ibuild libs src 	## Grade start up script for UN## ## ###############################
21		# Add default JYM options here. You can also use JAVA_OPTS and GRADLE_OPTS to pass JYM options to this script. DEFAULT_JYM_OPTS=""
	 build.gradle proguard-rules.pro 	APP_NAME="Gradle" APP_BASE_NAME='basename "\$0"`
	 gradie gitignore build gradle 	# Use the maximum available, or set MAX_FD != -1 to use that value. MAX_FD="maximum"
ы	gradle.properties	warn () { echo "\$*" }
uild Varian	gradlew.bat	die () { echo
₩ BL	local.properties settings.gradle	echo "\$#" echo exit 1
样 2: Favorites	External Libraries	} # OS specific support (must be 'true' or 'false'). <u>cycguin</u> =false meye=false
	🗵 Terminal 🧮 <u>0</u> : Messages 🛛 🏺 <u>6</u> : An	droid 🕨 🛓 : Run 🧐 TODO 📃 Event Log 🔄 Gradle Console
		1:1 LF ÷ UTF-8 ÷ %



Hello World アプリの内容確認

15. gradlew.bat

Gradle起動バッチスクリプト(Windos用)

Image: Section 2016 Image: Section 2016<	Eile	e <u>E</u> dit <u>V</u> iew <u>N</u> avigate <u>C</u> ode Analy <u>z</u> e	Refactor Build Run Iools VCS Window Help				
HelloWorldApplication Project Project Project Plane		# 💋 < 🄶 🔏 🛅 🗖 🔍 🞗					
Project © + b+ b Project Project Project Project Project	C	HelloWorldApplication					
Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Install plugne Ignore extension Pugne supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne Supporting files with *.bat are found Pugne Supporting files with *.bat are found supporting files with *.b	t	Project ▼ ③ ≑ 執・ I+	🗊 HelloWorldApplication.iml x 🛛 📊 gradle.properties x 🗐 local.properties x 🗐 gradlew x 🗐 gradlew.bat x 📲 🛛				
Image: Product Image	Proj	HelloWorldApplication (C:¥Users)	Plugins supporting files with *.bat are found Install plugins Ignore extension				
Image: State of the state	÷	🕨 🛅 .idea	@if "%DEBUG%" == "" @echo off				
Image: Section 2000 Image: Section 2000 I	-	🔻 🛅 app	8rem ####################################				
Terminal 0: Messages	ų	🕨 🛅 build	erem Brem Gradle startup script for Windows				
Strict 9 rem Frem	nctr	🛅 libs	Brem				
Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variables with windows NT shell Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable site with windows NT shell Image: Set local score for the variable site with windows NT shell <td>장</td> <td>Image: Signature Signat</td> <td>8rem ####################################</td>	장	Image: Signature Signat	8rem ####################################				
Image: State Stat		.gitignore	Bram Sat local scope for the variables with windows NT shell				
<pre> Build.gradle proguard-nules.pro proguard-nules.pro proguard-nules.pro proguard-nules.pro proguard-nules.pro proguard-nules.pro progradle gritignore gradle gradle.properties gradlew.bat gradlew.bat</pre>		J app.iml	if "XOSX"=="Windows_NT" set local				
Image: State of the second		build.gradle					
Image: Standard S		proguard-rules.pro	Brem Add default JYM options here. You can also use JAVA_OPTS and GRADLE_OPTS to pass JYM options to this script.				
Image: Set DIENAME = X* de0 if *XDIENAME = X* de0		🕨 🛅 gradle					
If * X0IRMAREX* == ** set 0IRMAREX* gradle.properties gradlew.bat gradlew.bat HelloWorldApplication.iml Idel of ined JAVA_HOME goto_findJavaFromJavaHome HelloWorldApplication.iml Idel of ined JAVA_HOME goto_findJavaFromJavaHome If * ZEIRMAREX* == *** set Idel of ined JAVA_HOME goto_findJavaFromJavaHome Idel of ined JAVA_HOME goto_findJavaFromJavaFromJavaHome Idel of ined JAVA_HOME goto_findJavaFromJavaFromJavaFromJavaFromJavaFromJavaFromJavaFromJavaFromJavaFromJavaFromJavaFr		.gitignore	set DIRNAME=% dp0 혁				
Set. AFF_DASC_MARE*X II0 Set. AFF_DASC_MARE*X II0 gradle.properties gradlew.bat if defined JAVA_HOME goto, findJavaFromJavaHome set JAVA_EXET=java.exe XJAVA_EXEX -version XNUL 22&1 if "XERRORLEVELX" = "0" goto, init echo.		Obuild.gradle	if "XDIRNAMEX" == "" set DIRNAME=.				
Gradlew Gradlew Gradlew.bat		gradle.properties	set APP HOME=XDIRNAMEX				
@ gradlew.bat @ rem Find java.exe if defined JAVA_HOME goto_findJavaFromJavaHome if defined JAVA_HOME goto_findJavaFromJavaHome iet JAVA_EXEE; ava.exe XJAVA_EXEE; ava.exe XJAVA_EXEE; ava.exe XJAVA_EXE; ava.exe if (* SERRELEVELX* = "0" goto init echo. echo. echo. echo. echo. echo. echo. echo. <	ท	aradlew					
Image: Section	riant	aradlew.bat	Brem Find java.exe				
set JAYA_EXE=java.exe JAYA_EXE JAYA_EX	e > E	I HelloWorldApplication.im	IT derined Java_nume gotto TindJavarromJavanome				
Im Commence of Sections.gradle XJAYA_EXEX -version >NUL 22%1 If "XEPROPELEVELX" == "0" goto init If "XEPROPELEVELX" == "0" goto init echo.	Build	De local properties	set JAVA_EXE=java.exe				
If Account strate - 0 goto Init echo. echo.	÷	(settings gradle	%JAVA_EXE% -version >NUL 2>&1				
echo. echo ERROR: JAVA_HOME is not set and no 'java' command could be found in your PATH. echo. echo Please set the JAVA_HOME variable in your environment to match the Terminal ■ 0: Messages ♣ 6: Android ▶ 4: Run ♀ TODO ■ Event Log ■ Gradle Console 1:1 CRLF • UTF-8 + ⊕ ⊕	v	External Libraries	IT SERMULLEVELS U gotto Init				
echo ERROR: JAVA_HOME is not set and no 'java' command could be found in your PATH. echo. echo. echo Please set the JAVA_HOME variable in your environment to match the Image: Terminal Image: Messages Image: Android ▶ 4: Run Image: TODO Image: Terminal Image: Messages Image: Topo Int: CRLF : UTF-8 : Image: Topo Int: CRLF : UTF-8 : Image: Topo	orite		echo.				
echo Please set the JAYA_HOME variable in your environment to match the Image: Terminal Image: Messages Image: Messages Image: Android Image: Messages Image: Android Image: Messages Image: Android Image: Messages Image: Messages Image: Messages Im	Favo		echo ERROR: JAVA_HOME is not set and no 'java' command could be found in your PATH.				
Image: Terminal Image: Messages Image: Android Image: Android Image: ToDO Im	ivi N		ecno. echo Please set the JAVA HDME variable in your environment to match the				
Image: Terminal Image: Messages Image: Android Image: Android Image: Android Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image: Topo Image:	*						
1:1 CRLF + UTF-8 + 6 🕀		🔄 Terminal 🛛 🧮 <u>0</u> : Messages 🛛 🌞 <u>6</u> : An	droid 🕨 🛓 : Run 🧐 TODO 📃 Event Log 🔳 Gradle Console				
			1:1 CRLF 🗧 UTF-8 🗧 🎡				



Hello World アプリの内容確認

16. gradle.properties

File	👷 HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [HelloWorldApplication]¥gradle.properties - Android Studio (Bet 👝 💿 💌 File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help			
	₩ Ø 🗸 🔶 🗶 🛍 🖉 🙊		Q,	
	HelloWorldApplication	erties		
ţ	🗊 Project 👻 😳 寺 泰- [[+	C HelloWorldActivity.java × I app.iml × I HelloWorldApplication.iml × I gradle.properties ×	+≣6 (€	
Proj	HelloWorldApplication (C:¥Users	# Project-wide Gradle settings.	l Grad	
÷i O	▶ 🛅 .idea	# IDE (e.g. Android Studio) users:	ī	
	▶ ⊡ app	# Settings specified in this file will override any Gradle settings	n	
ture	▶	# configured through the IDE.	Mave	
Struc	 giugnore build gradle 	# For more details on how to configure your build environment visit	5 Pa	
5		# http://www.gradie.org/docs/current/ <u>userguide</u> /build_environment.html	oject	
¥	gradlew	# Specifies the JVM arguments used for the daemon process.	0	
	gradlew.bat	# The setting is particularly useful for tweaking memory settings. # Default value: -Xmx10248m -XX:MaxPermSize=256m	1	
	J HelloWorldApplication.iml	# org.gradle.jvmargs=-Xmx2048m -XX:MaxPermSize=512m -XX:+HeapDumpOnOutOfMemoryError -Dfile.encoding=	UTF-8	
	🖬 local.properties	# When configured. Gradle will run in incubating parallel mode.	hande	
	settings.gradle	# This option should only be used with decoupled projects. More details, visit	4	
	External Libraries	# http://www.gradle.org/docs/ourrent/ <u>userguide</u> /multi_project_builds.html#sec:decoupled_projects # org.gradle.parallel=true		
👘 Build Variants				
¥ 2: Favorites				
_	🔄 Terminal 🛛 📃 0: Messages 🛛 🐳 6: And	Iroid 🕨 4: Run 🌚 TODO 📃 Event Log	Gradle Console	
		1:1 CRLF ‡	windows-31j 🏻 🔒 🛱	



Hello World アプリの内容確認

17. local.properties

File	🚍 HelloWorldApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloWorldApplication] - [HelloWorldApplication]¥local.properties - Android Studio (Beta) 👝 💽 🔤				
	H Ø 🗸 🔶 🕅 🛍 🔍	← ⇒ H ⊕ app v ▶ ★ 및 % ⊕ \$ 및 凿 ♣ ?	Q,		
	HelloWorldApplication	erties			
ţ	🗊 Project 👻 😧 崇 🏘 🔭 🚰	VorldActivity.java 🗴 🗊 app.iml 🗴 🗊 HelloWorldApplication.iml 🗴 📑 gradle.properties 🗴 📑 loca	I.properties × 🖃 (∂		
Proje	HelloWorldApplication (C:¥Use	## This file is automatically generated by Android Studio.			
÷.	🕨 🛅 .idea	# Do not modify this file YOUR CHANGES WILL BE ERASED!	<u>n</u>		
-	🕨 🛅 app	# # This file should *NOT* be checked into Version Control Systems	m		
ų	🕨 🛅 gradle	# as it contains information specific to your local configuration.	A		
nctri	.gitignore	# # Leasting of the CDK. This is only used by Orealis	len len		
ţ	💽 build.gradle	# Location of the SUK. This is only used by Gradie. # For customization when using a Version Control System, please read the	Proj		
	📊 gradle.properties	# header note.	ects -		
	gradlew	sdk.dir=C¥:¥¥Program Files¥¥Android¥¥android-studio¥¥sdk	-		
	📄 gradlew.bat		2		
	J HelloWorldApplication.iml		omn		
	📊 local.properties		land		
	settings.gradle		e,		
	External Libraries				
훤					
'aria					
jid 🗸					
Bu					
Ø					
avor					
E E					
*					
	🔟 Terminal 📃 <u>0</u> : Messages 🛛 🌞 <u>6</u> : A	ndroid 🕨 👍 Run 🧁 TODO 📃 Event Log	Gradle Console		
		1:1 CRLF ‡	windows-31j 🍙 👰		

Hello World アプリの内容確認 実をいうと... Android Studio には、プロジェクトの アクティビティ表示内容の初期値として Hello World表示が含まれています。

このため Hello World アプリは、 すでに実装済みとなっています。



Hello World アプリの実行 (アプリのエミュレータ実行)

- モジュール選択確認 Android Studio 上部にあるナビゲーション・ ツールバーのモジュール選択リストボックスで、 appモジュール(ドロイドくん+app)を選択 します。
- モジュール実行
 Android Studio 上部にあるナビゲーション・ ツールバーの右向き三角アイコン(Run 'app')を クリックして、app モジュールを実行します。



 アプリ実行のGradleタスク起動 前ページでアプリ実行が要求されたので アプリ実行のGradleタスクが走りだします。

Android Studio 下部にるステータス・ バーにビルド・タスク進行が表示されます。



 アプリ実行デバイスの選択 実行準備が整えば、アプリ実行デバイスの 選択ダイアログが開きます。

> Choose running device は、 アプリ実行デバイスとして、 PCに接続された実機や、実行中のエミュ レータを選択する項目です。

Lunch emulator は、 アプリ実行デバイスとして起動する エミュレータを指定する項目です。

ここでは、Lunch emulator で 予め作成しておいたエミュレータ (Android_4.4.2)を選択してください。

アプリ実行するデバイスを選択しましたら 「OK」をクリックしてください。



 アプリ実行エミュレータの起動 前ページの操作で アプリを実行するエミュレータが起動します。

エミュレータ起動には、時間がかかります。 起動が終了すると、ロック画面になるので、 手動でロックを外すと(鍵アイコンを右スライド)、 Hello Word アプリが実行されます。




Hello Wordアプリ実行は、これで終了です。 発展学習として、Hello World アプリは、 画面に「Hello World!」と表示するだけですが、 Android では、たくさんの工程とソースファイルや ビルド設定が必要となることを もう一度御確認ください。



プロジェクト画面のクローズ

(Welcome画面にもどる)

1. メニューバーからプロジェクトを閉じる 画面上部のメニューバーから File > Close Project を選択して、 プロジェクト画面を閉じます。

(Welcome画面にもどります)



Android Studio 終了

(IDEを終了させる)

 メニューバーから終了する 画面上部のメニューバーから File > Exit を選択します。

> 終了確認ダイアログが開くので、 「Exit」をクリックして、 Android Studioを終了させます。





Android Studio におけるプロジェクトと設定の保管先 Android Studio を使ってみましたので発展学習として、ユーザ・ホームディレクトリ内に保管される Android Studio におけるプロジェクトの標準保管先と、設定内容の保管先を確認してみてください。

- プロジェクトの標準保管先 Android Studio におけるプロジェクトー式は、標準設定だと ユーザ¥アカウント名¥AndroidStudioProjects 以下に保管されます。 (例) C:¥Users¥rie¥AndroidStudioProjects (【注意】ユーザ・アカウントID: rie が操作した場合)
- 2. Android Studio IDE の設定内容 (Settings)の保管先

ユーザ¥アカウント名¥.AndroidStudioBeta以下に保管されます。

(Settingsで指定した設定は、上記の config ディレクトリに保管されます)
 (一時作業ファイルなどは、 上記の system ディレクトリに保管されます)
 (例) C:¥Users¥rie¥.AndroidStudioBeta (【注意】ユーザ・アカウントID: rie が操作した場合)

3. 【補足】Android Studio では、レジストリを使用しないそうです。

演習2) Example を動かしてみよう

Android Studio のプロジェクト・インポートについて Android Studio では、外部で作られたアプリなどのプロジェクトを 自分のプロジェクトとして、取り込むことができます。

演習2)の目的

この演習では、実際に意味のある挙動をしてくれる Android SDK に含まれるサンプル・プロジェクトを使い 自分の Android Studio にインポートした上で、アプリの挙動やソースの内容および Android Studio の機能を 自分の目と手を使って確認してもらいます。(ソースプログラムのシンタクスやAPIの意味は、問いません)

Android Studio のプロジェクト・インポートについて

プロジェクト・インポート時の注意事項

Android Studio では、インポートするプロジェクトのフォルダ自体を Android Studio の所有物として扱います。

このためインポートしたプロジェクトの内容を変更すると、 インポート元のプロジェクト内容が変更されてしまいます。

インポート元のプロジェクト内容を保持したい場合は、 プロジェクトのフォルダー式のコピーを作って、 コピーしたフォルダのプロジェクトをインポートしてください。

Android Studio のプロジェクト・インポートについて

インポート・プロジェクトのタイプ Android Studio では、大きく分けて下記3つのタイプがあります。 (ライブラリなどのプロジェクトは、ここに含めていません)

- 1. Eclipse プロジェクト Eclipse 用の スマートフォンまたはタブレットを対象としたプロジェクト
- Android Studio mobile プロジェクト
 Android Studio 用の スマートフォンまたはタブレットを対象としたプロジェクト
- Android Studio Wear プロジェクト
 Android Studio 用のAndroid Wear 単独、
 または Android Wear + mobile(スマートフォン)を対象としたプロジェクト

現時点では制限がありますが、 Eclipse 用のプロジェクトであっても、Android Studio 用のプロジェクトに自動変換できます。

Android Studio のプロジェクト・インポートについて

具体的なサンプル・プロジェクトのインポート手順例

- インポートするサンプル・プロジェクトのコピー作成 Android Studioのプロジェクト標準保管先(*1)に sample フォルダを作成し、 Android SDK 配置先(*2)の sample¥androidバージョン¥ 下のサンプル・プロジェクトをコピーします。
 - (*1) ユーザ¥アカウント名¥AndroidStudioProjects
 (例) C:¥Users¥rie¥AndroidStudioProjects(【注意】ユーザ・アカウントID: rie が操作した場合)
 - (*2) Android Studio インストール先¥sdk (例) C:¥Program Files¥Android¥android-studio¥sdk

2. プロジェクトのインポート

Welcome 画面の場合 : Quick Start > Import Project プロジェクト画面の場合 :メニューバー > File > Project Import 両者いずれでもファイル・ツリーから、コピーしたサンプル・プロジェクトのフォルダを選択します。

Android Studio のプロジェクト・インポートについて

(演習案)インポートするサンプル・プロジェクト

- Eclipse プロジェクト
 インストール先\sdk\samples\android-19\legacy\NotePad
 (簡易メモ帳アプリ)
 (遊べないのでデモとして見るだけですが JetBoy も楽しいかも?)
- Android Studio mobile プロジェクト インストール先\sdk\samples\android-19\ui\BasicAccessibillity (様々なUIの見本)
- Android Studio Wear プロジェクト インストール先\sdk\samples\android-20\wearable\Flashlight (Android Wear単独のアプリ/左フリックすると画面が明滅するだけのアプリです)

インポート手順例前半(Eclipse プロジェクト)	
Webcome to Android Studio	4. Eclipseプロジェクトを はのなりまう場合は、jarや うイブラリの変換を確認。 クイブラリの変換を確認。 <i>につっれい指定での」</i> Pot Protect from ADT (Eclipse Androut) Pot Pot Protect from ADT (Eclipse Androut) Pot
Import する Project を 選択し「 0K」をクリック	Previous Finish Cancel Help

インポート手順例 前半 (Android Studio mobile / wear プロジェクト)				
Android Stude Welcome to Android Studio	3. (Import 変換出力先指定なし) Import元を出力先に使います。 4. (ライブラリ変換なし)			
迭択し「UK」をクリック				

インポート手順例 後半(Eclipse /	Android Studio プロジェクト)	
Sync Android SDKs		
The path 'C:/Program Files/Android/android-studio/sdk' does not refer to an Android SDK.	Welcome to Android Studio	Contracted - [C:HUsersWeitMandroidStudioRrojectsWickehed] - [NotePad]Wingort-summary.bt - Android Studio (Seta) 0
Android Studio will use its default SDK instead: '/home/rie/bin/android-studio/sdk'	Recent Projects Quick Start HelloWorldApplication	Import-summary.bt import-summary.bt Import-summary.bt import-summary.bt x Import-summary.bt import-summary.bt x Import-summary.bt import-summary.bt x
and will modify the project's local properties file.	Building 'NotePad' Gradle project info Gradle: Build Cancel	Image: Sector and Libraries 2 Image: Sector and Libraries 3 Image: Sector and Libraries 4 Moved Files: 3 Image: Sector and Libraries 4 Moved Files: 3 Image: Sector and Libraries 5 4 Moved Files: Image: Sector and Libraries 6 4 Moved Files: Image: Sector and Libraries 6 4 4 Image: Sector and Libraries 7 Eclipse movies use a different directory structure than ADT Image: Sector and Libraries 7 Eclipse movies use a different directory structure than ADT Image: Sector and Libraries 7 Eclipse movies use a different directory structure than ADT Image: Sector and Libraries 8 * Android function feetant Image: Sector and Libraries 8 * Android function feetant
	Configure	10 *
5. プロジェクトの基本設定の 変換を行う場合は、確認が 表示されます。	6.Import 処理が始まります。	7.Import 処理終了
(変換を行うので「OK」をクリック)		26 (This import summary is for your information only, and can be deleted 27 after import once you are satisfied with the results.)
		Terminal ♣ 5: Android

Android Studio でよく使う設定 Android Studio は、高機能な IDE (開発環境)ですが、標準では、いくつかの機能がオフに設定されています。 ここでは、サンプル・プロジェクトのソース確認や、アプリ開発時に役立つ代表的な設定を紹介します。

- 1. 行番号を表示させる。
 - メニューバー > File > Settings... > (Settingsダイアログ -IDE Settings-) Editor > Appearance show line numbers (行番号表示) にチェックを入れます。
- 2. マウスオーバーで、Javadocをポップアップさせる。
 - メニューバー > File > Settings... > (Settingsダイアログ -IDE Settings-) Editor > -other- Show quick doc on mouse move (マウスオーバJavadoc表示) にチェクを入れます。
- ショートカットキーを変更する(ショートカットキー設定一覧を確認する)
 英語表記ですが、機能名とショートカットキー割当ての一覧が表示されます。
 メニューバー > File > Settings... > (Settingsダイアログ IDE Settings-) Keymap > Editor Actions や Main menu など(Main menuは、メニューバーへの割当)
 虫眼鏡アイコンを利用すれば、機能名やキーによる項目検索もできます。

Android Studio でよく使う設定

4. TODOを明示する。

Pattern で指定したキーワードを先頭に持つコメントを青色で表示し、マーカーバーにマークをつけます。 またツールウィンドウ・バーの「TODO」をクリックすれば、TODOを検索結果一覧が表示されます。

メニューバー > File > Settings... > (Settingsダイアログ -IDE Settings-) TODO > 標準で用意されている Pattern "¥btodo¥b." と "¥bfixme¥b."の Case Sensitive に チェクを入れれば "todo" と "fixme" がTODOキーワドとして使えます。

(他にも便利な機能がありますので、御自身で Settings ダイアログを確認してみてください)

		番号を表示する hellopracticeapplication〉 HelloPracticeActivity〉
Settings	Editor > Appearance	Construings.xml x C HelloPracticeActivity.java x -=== A //todo 演習5) ボタン押下でメッセージを出そう //演習内容に沿って、このメソッドに処理を追加してください。 Boverride proble vaid opViewC
Scopes Scopes Spelling Template Data Languages Template Data Languages Template Data Languages Terminal Version Control DE Settings Appearance Console Folding Debugger Editor Editor Smart Keys Appearance Colors & Fonts Editor Tabs Code Folding Code Completion Auto Import Postfix Completion	✓ Use anti-aliased font ✓ Caret blinking (ms): 500 Use block caret ✓ ✓ Show right margin (configured in Code Style options) ✓ ✓ Show with mumbers Show wethod separators Show whitespaces ✓ ✓ Show vertical indent guides ✓ ✓ Show vertical indent guides ✓ ✓ Show breadcrumbs (Reopen editor for changes to take effect) Show CSS color preview icon in gutter Show CSS color preview as background ✓ ✓ Enable HTML/XML tag tree highlighting Levels to highlight: Levels to highlight: 6 🗢 Opacity: 0.1 ≑	public void onview * com * example * hellopractice publication * HelloPracticeActivity.java * *********************************

Android Studio でよく使う設定

٩)	Editor		
Gradle	Recent files limit: 50		
Language Injections	Console commands history size: 300		
Schemas and DTDs Scopes	Error highlighting	Formatt	
Spelling Tasks	Error stripe mark min height (pixels):		
Template Data Languages	Autoreparse delay (ms): 300		
Terminal Version Control	$\textcircled{\sc original}$ 'Next Error' action goes to high priority problems only	Refacto	
IDE Settings	☑ Suppress with @SuppressWarnings (for Java 5.0 only)	\checkmark	
Appearance Console Folding	Other		
Editor	Strip trailing spaces on Save: Modified Lines		
Emmet (Zen Coding)	Ensure line feed at file end on Save		
External Diff Tools External Tools	Show quick doc on mouse move Delay (ms): 500		
File and Code Templates			

マウスオー	バーでJavadoc	をポッフ	゚アップする	
	メソッドやクラス マウスオーバール Javadocが表示	などでの こより されます	ractice hellopracticeapplication	C HelloPracticeActivity
Formatt	68 69 70 71 ■ 72 73 74	80verride public void	5) ポタン押下でメッセージを出そう みって、このメソッドに処理を追加してくださ onViewCreated(View view, Bundle savedIns Documentation for onViewCreated(View, P	tanceState) [aundle)
	74 75 76 77 78 79 80 81 81 82 83 84 84	com. example.ht public void onY Description cop Caled immedatel returned, but be subclasses a chan been completely its parent at this	ellopractice.hellopracticeapplicati iewCreated(<u>Yiew</u> view, <u>Bundle</u> savedInstanceState) ied from class: <u>Fragment</u> , y after <u>onCreateView(Layout Inflater, View</u>) fore any saved state has been restored int to ce to initiate themselves once they know t created. The fragment's view hierarchy is ni- point.	iroup, Bundle) has ther ivew hierarchy has ot however attached to
Help	86 87	*/		

Android Studio でよく使う設定

Main Menu は、メニュー・バーに相当します。 階層構造をたどれば、メニューから使える 各機能のショートカットが確認できます。

ettings		×
	Keymap	
olding Debugy	Keymaps: Default	Reset Delete
Emmet (Zen Coding)	₹ ₹ 8-	۵ 🗊
xternal Diff Tools	🔁 Main menu	
xternal Tools	File	
le and Code Templates	Fi Edit	
le Types	View	
eneral	Tool Windows	
TTP Proxy	Ouick Definition	Chult Chiffs I
nages	Chow Siblings	Ctri+Shitt+1
itentions	Ouide Degumentation	
eymap Tarada tara	Quick Documentation	Ctrl+Q Alt+Button2 Click
ve Templates	External Documentation	Shift+F1
etitus attu Tooluars	Parameter Info	Ctrl+P
asswords	Context Info	Alt+Q
ath Variables	Error Description	Ctrl+F1
uains	Jump to Source	F4
uick Lists	Show Source	Ctrl+Enter



発展 学習

サンプル・プロジェクトのインポート

Android Studio でよく使うコードアシスト機能など ここでは、サンプル・プロジェクトのソース確認や、アプリ開発時に役立つ代表的な機能を紹介します。 いずれもエディタエリアのソースコードを対象にした機能です。

- 1. カーソル位置キーワードのコード補完(候補一覧表示) メニューバー > Code > Completion > Basic (ショートカット: CTRL + Space)
- キーワードの宣言元/使用先の一覧表示 カーソル位置にあたるキーワードが使用されている箇所の一覧を検索して表示します。 ショートカット :ALT+F7 右クリック・ポップアップメニュー :Find Usage を選択
- 3. カーソル位置キーワードのソースへジャンプ

メニューバー > View > Tool Windows > Jump to Source (ショートカット:F4) (標準設定では、メニューバーからの機能選択が非表示になっているようです)

Android Studio でよく使うコードアシスト機能など

4. その他

メニューバーの Navigate 機能は、カーソル位置の状況に合わせて、 (宣言元や継承元など)目的先へのジャンプや、(継承関係などの)関係一覧表示などが行なえます。 Navigate > Type Hierarchy (ショートカット: CTRL + H) で、クラス継承構造の一覧表示など。

他にも便利な機能がありますので、御自身で確認してみてください)



Android Studio でよく使うアシスト機能

発展

カーソル位置キーワードのソースへジャンプ

java com		
C HelloPracticeActivity.java × m	C HelloPracticeActivity.java × C Activity.java ×	
1 package com.example.he カーソル位置(例Activity)で 2 #import F4を実行します。 16 Public class HelloPracticeActivity extends Activity { 18 Public class HelloPracticeActivity extends Activity {	633 * should start a fallink Serviced unlead takes place. This allows 641 * th 641 * in 7 Public class Activity extends Context ThemeWrapper	
20 80verride 21 protected void onCreate(Bundle savedInstanceState) { 22 super.onCreate(savedInstanceState); 23 setContentView(R.layout.activity.hello_practice); 24 if (savedInstanceState == null) { 25 getFragmentManager().beginTransaction() 26 .add(R.id.container, new PlaceholderFragment()) 27 .commit(); 28 } 30 31	646 implements LayoutInflater.Factory2, 647 Window.Callback, KeyEvent.Callback, 648 OnCreateContextMenuListener, ComponentCallbacks2, 649 Window.ONWindowDismissedCallback { 650 private static final String TAG = "Activity"; 651 private static final boolean DEBUG_LIFECYCLE = false; 652 /** Standard activity result: operation canceled. */ 654 public static final int RESULT_CANCELED = 0; 655 /** Standard activity result: operation succeeded. */ 656 public static final int RESULT_OK = -1; 657 /** Standard activity result: operation succeeded. */	



演習3)に進む前に

全体演習サンプル・プロジェクトのインポート

プロジェクトのインポートやソースの見方や触り方に慣れてもらったと思います。 以降の演習では、実際にコードを自分で書いたり変更したりして、動作を確認することで、 プログラミングへのより深い理解と学習成果を実感してもらいます。

ここで、以降の演習で利用する、 演習サンプル・プロジェクトをインポートしてください。 演習サンプル・プロジェクトは、以下のURLからダウンロードできます。

http://cch-lab.com/reference/AndroidBasicStudy_20141018/HelloPracticeApplication.zip

演習サンプルは、Hello World を少し改修しただけの簡単な内容です。

演習3)メッセージを変更してみよう

ソースの入力や変更を行う演習サンプルをインポートされましたね。 オリジナルの HelloWorld アプリのメッセージは、"Hello world!"と英語でした。 演習の手始めに、メッセージを日本語にしてみましょう。

演習サンプル HelloPracticeアプリの画面表示を定義している、 app/src/main/res/layout/fragment_hello_practice.xml の Text タブを開いてください。

【指定ファイルを開く操作手順】

左ツールウィンドウの Project を開き、 HelloPracticeApplication プロジェクトのファイルツリーから、 app/src/main/res/layout/fragment_hello_practice.xml をダブルクリックします。

エディタエリアに fragment_hello_practice.xml が開きますので、 エディタエリア下部にある Text タブを選択してください。



メッセージの変更(1)

 TextView 要素の text 属性の内容を "@string/hello_world" から "こんにちは、世界!"に変更してみてください。

TextViewの初期内容は下記の通りです

<TextView

android:text="@string/hello_world"
android:layout_width="wrap_content"
android:layout height="wrap content" />

右隣のPreview表示を確認します。
 Preview は、Textで定義した画面内容をイメージ化するウィンドウです。
 メッセージが、Hello world! から変化していることを確認できたでしょうか。

【学んだこと】 画面に文字を表示するには、TextView 要素を使います。 メッセージの内容は、text 属性に指定します。



メッセージの変更(2)

 Android Studio からの警告を確認する マウスカーソルを text 属性の変更した内容に移動(マウスオーバ)してください "should use @string resource"(@string リソースを使うべき)とポップアップされ Android Studio から怒られることを確認します。

【学習ポイント】 @string リソースとは何か? 何故、怒られるのかを考えてみてください。

メッセージの変更(2)

変更を元に戻します
 怒られたままでは嫌なので、変更を元に戻しましょう。
 text 属性の内容を "@string/hello_world" に戻します。

右隣のPreview表示を確認します。 メッセージが、Hello world! に戻っていることを確認します。

【学習ポイント】 何故、表示が Hello world! に戻ったのか? @string/hello_world の実態はどこにあるのかを想像してみましょう。

勿体づけましたが、次のページから、 @stringリソースの実態がどこにあるのか種明かしをします。 それでは、app/src/res/values/string.xmlを開いて下さい。





メッセージの変更(3)

 Hello World! 定義元リソース name 属性が hello_world の string 要素を見てください。 text 内容が、Hello world! になっているリソースがあります。 これが、@string/hello_world リソースの実態(定義元)です。

@string/hello_world リソースの初期内容は下記の通りです
<string name="hello_world">Hello world!</string>

 hello_world リソースの内容変更 name 属性が "hello_world" の string 要素の text を Hello world! から こんにちは、怒られない世界! に変更してみてください。

	res 🔉 🛅 values 🛛 🖬	strings.xml			
÷	🖻 strings.xml 🗴	\bigcirc HelloPracticeActivity.java ×	o fragment_hello	_practice.>	
c	1 xml versi</td 2 ⊖ <resources< td=""> 3 4 <string< td=""></string<></resources<>	ion="1.0" encoding="utf-8"?> > : name="app_name">HelloPractices	Application <th>></th> <th></th>	>	
	5 ≺string 6 ≺string 7	<pre></pre>			
	8 A/resources				
			5 <string 6 <string 7</string </string 	: <mark>name="app_name"</mark> >HelloPractic <mark>: name="hello_world"</mark> >こんにち! : name="action_settings">Sett	eApplication は、怒られない世界!K <mark>/string></mark> ings
			8 A/resources	s>	

メッセージの変更(3)

3.リソース変更の確認

fragment_hello_practice.xml の Text タブを開き 右隣のPreview表示が、入力内容に変化していることを確認します。

TextView の text 属性にマウスカーソルをかざしても Android Studio から怒られないことを確認してください。

【学んだこと】

@string/hello_world の実態は、
 strings.xml の name 属性が hello_world の string 要素データである。
 つまり stringリソースは、@string/ + name 属性で指定できる。
 以上は、リソースの基本ですので覚えてくださいね。

英語⇔日本語の切り替えをする場合、英語と日本語で name 属性が同じ stringリソースを定義して、 画面表示設定で、その stringリソースを利用すればリソース先を切り替えるだけで英語⇔日本語が対応できますから、 Android Studioがテキストの直書きを怒ったのです。

strings.xml × C HelloPracticeActivity.java ×	stragment_hello_practice.xml	lello_practice.xml
1 xml version="1.0" encoding="utf-8"? 2 ⊖ <resources> 3 4 5 5 6 <string name="app_name">HelloPractices 7 8 9 9</string></resources>	Application <b string> 、怒られない世界! <mark>//string></mark> ngs <b string>	ent_hello_practice.xml × =s Preview テキスト内容が"こんにちは、怒られない世界!"の hello_worldという@stringリソースを定義します。 snt-matcn_parent t="64dp" ht="64dp" = "16de" tom @stringリソース内容が、 llg メッセージに反映されています。
fragment_hello_practice.xml では 「@stringリソースを使うべし」と 怒られません。	10 11 CrextView android:text="android:layout android:layout android:layout Android:layout Android:layout Android:layout Android:layout Android:layout Android:layout Android:layout Android:layout	"@string/hello_world" t_width="wrap_content" t_height="wrap_content" />
演習4)ボタンを配置してみよう

今の画面には、メッセージしか表示されないので変化がありません。 画面にボタンを配置してみましょう。

HelloWorldアプリの画面表示を定義している、 app/src/main/res/layout/fragment_hello_practice.xmlの Design タブを開いてください。



- 画面にボタンを追加配置する。
 Palette の Widgets から Button をマウスで選んで、
 右隣のデバイスの画面中央にドラッグ&ドロップしてください。
- 2. 画面のボタン配置を確認する。 Text タブを開き、右隣りのデバイスの画面中央に「New Button」というラベルのボタンが配置され、 エディタエリアのXML内容に、下記の Button 要素が追加されていることを確認します。 <Button android:layout width="wrap content" android:layout height="wrap content" android:text="New Button" android:id="@+id/button" android:layout centerVertical="true" android:layout centerHorizontal="true" /> 【学んだこと】 画面にボタンを表示するには、Button 要素を使います。 画面にボタンを配置するには、Design タブでのドラッグ&ドロップで行えます。 Paletteから、CheckBoxなど他のWidget(UIパーツ)を選んでドラッグ&ドロップもできます。



3. エミュレータで、ボタンをクリックしてみる エミュレータを起動して、画面表示を確認してから 「New Button」ボタンを押下してください。

> ボタンをクリックしても、何も反応が返らないないことを確認します。 「ボタンのクリックで何をするのか」をプログラミングしていないのですから当然です。

【学習ポイント】 ボタンを配置しただけでは、クリック時の反応が返りません。 ボタン押下時に反応するには、別途対処が必要になりますが、 どうすればよいのかを想像してみましょう。



演習5)ボタン押下でメッセージを出そう

画面にボタンを配置しただけでは、 ボタンをクリックしても何も反応が起きませんでした。

ボタンのクリックで何かを行うためには、別途ボタン押下時の処理が必要と判かりましたので、 HelloPracticeアプリ画面の表示処理に、ボタン押下時の処理プログラムを追加します。

HelloPracticeアプリ画面の表示や動作の処理を定義している、 app/src/main/java/com.example.hellopractice.hellopracticeapplication/ HelloPracticeActivity.javaを開いて下さい。

この演習で行うこと…

```
ここでは、ボタン押下でメッセージを表示するよう、
画面表示元である HelloPracticeActivity クラスの
PlaceholderFragment サブクラスの onViewCreated メソッドに、
以下のボタン押下時のアクションを指定するプログラムを入力します。
```

```
Button button = (Button) view.findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Toast.makeText(getActivity(), "クリック!", Toast.LENGTH_LONG).show();
    }
});
```

このプログラミング・コードの内容は、今は「おまじない」と思ってください。



ボタン押下時のメッセージ表示処理追加プログラミング

 1行目の "Button button = (Button) view.findViewById(R.id.button);" の入力 初めに "Button "(スペース含む)まで入力してみましょう。 "android.widget.Button ? Alt + Enter" と指摘されませんか?

Android Studio が「Button は android.widget.Button のことなの?」と尋ねているのです。 ここでは、Alt + Enter を入力してください。

すると、自動的に import android.widget.Button; がソース先頭あたりに追加されます。 これが、Android Studio(IntelliJ)でインテンションと呼ばれる、コード修正の提案機能です。

```
続いて、Button button = (Button) view.find まで入力すると
"findViewByld(int id) view" が提案されますので Enter で採用しましょう。
これで Button button = (Button) view.findViewByld() まで入力されました。
```

続いてカッコ内で R.id. まで入力すると "button" が提案されますので Enter で採用します。 最後に残ったカッコの後ろに ; を入力すれば、一行目の入力が完了しました。



83 //上記の 84 */ 85 Button b	^{内容を自分で記述してみましょう。} tton = (Button) view.find	入力すると ます。
86 A } 87 A } 88 } 89 } 0: Messages	In a findViewById (int id) IndFocus () IndFocus () IndFindViewsWithText (ArrayList <view> outViews, Chars IndViewWithTag (Object tag) IndFinishTemporaryDetach () IndFinishTemporaryDetach ()</view>	View View View View void
oo //トコの		1
84 */ 85 ₽ Button be 86 ⊖ } 87 ⊖ } 88 }	tton = (Button) view.findViewById() @IdRes int id コードが入力(お	/iewByld を [⊤] ると 補完)されます。
<u>0</u> : Messages to '()'	Event د	Log E Gradle Console



 2行目の "button.setOnClickListener(new View.OnClickListener() {"の入力 初めに button.setOn まで入力してください。 "setOnClickListener(OnClickListener I) void" が提案されますので Entere で採用すると、 button.setOnClickListener(); まで入力されますが、 カッコに赤く下線が引かれて警告されていると思います。

```
次に、カッコの中にカーソルを移動して、"new "(スペース含む)を入力したら
CTRL+Space を入力してください。(メニューバー > Code > Complection > Basic でも可)
"onClickListner(){...} (android.view.view)" が提案されますので Enter で採用します。
```

```
すると、一気に下記の内容が入力されます。
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
```

```
});
先ほどの CTRL+Space が、Android Studio の<mark>Basicコード補完機能</mark>です。
```





3. 残った Toast.makeText(getActivity(), "クリック!", Toast.LENGTH_LONG).show(); の入力 最後に onClick メソッドの中身を入力しましょう。

Toast. まで入力しましたら、

"Toast.makeText(Context context, CharSequence text, int dulation)... Toast"が 提案されるので Enter で採用して、Toast.makeText() まで自動入力します。

続いてカッコ内で getA まで入力して "getActivity() Activity" の提案を Enter で採用して、 引数区切りの,を入力します。

メッセージは、そのまま "クリック!", を入力しましょう。

最後に Toast. まで入力し "Toast.LENGTH_LONG" の提案を Enter で採用すれば、引数入力の終了です。

まだ閉じカッコに下線がついて警告されていますので .s まで入力して "show() void" を Enter で採用すると、 自動的に末尾の ; まで補完されます。

以上でメッセージ出力行の Toast.makeText(getActivity(), "クリック!", Toast.LENGTH_LONG).show(); …が完成しました。







ボタン押下でメッセージが出るか確認
 エミュレータを起動して、画面表示を確認してから、「New Button」ボタンを押下してください。
 今度は、「クリック!」と画面にメッセージが表示されたと思います。

プログラムしたコード内容は、ボタンのクリック時の処理先に、メッセージの出力処理を設定しています。

プログラミング用語で説明すると

「ボタン・オブジェクトのクリック・イベントリスナに、

クリック・イベントの処理としてメッセージ出力を行うイベントハンドラを設定している」となります。

ソースコードの入力体験は、いかがでしたか Android Studio は、コードの文脈に合わせてコード入力の提案や自動入力(補完)をしてくれる 便利な IDE ですね。





id は、オブジェクトを特定する 大切な識別子

演習5)の「おまじない」の1行目(*1)では、

ボタン・オブジェクトを画面(view)から findViewByld メソッドで抽出しています。

(*1)"Button button = (Button) view.findViewById(R.id.button);"

では、ボタン・オブジェクトを特定した R.id.button は、どこで定義されているでしょう。 実は、R.id.button は、

画面表示を定義している fragment_hello_practice.xml に配置した Button 要素 <Button

android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="New Button" android:id="@+id/button" android:layout_centerVertical="true" android:layout_centerHorizontal="true" />

…の id 属性の内容(button)を利用しているのです。

画面には、ボタンやテキストなど複数のUIパーツ(Wiget)が配置できます。 各Widgetが特定できるよう id 属性の内容は、固有の値となるよう注意してください。

発展 学習

メッセージと一緒に Log を出力しよう

Android には、ログ出力を行う logcat と呼ばれる標準の API があります。 logcat は、アプリ開発時のデバッグ出力などに役立ちますので実践してみましょう。 ログ出力は、画面下の Android DDMS ツールウィンドウ内の logcat に表示されます。

それでは、演習5)のボタン押下時の onClick メソッドでの Toast を使ったメッセージ出力の下に、下記のログ出力を追加してください。

Log.d("Practice", "クリック・ログ出力!");

Ancdoid Studio のコード修正提案の採用やコード補完に慣れましたか? 入力が終わりましたら、ナビゲーション・ツールバーの Run ボタンをクリックしてアプリを実行しましょう。

Android DDMS ツールウィンドウの logcat に、

ボタン押下とともに"クリック・ログ出力"が出力されることを確認してください。 入力したコードの概要は、logcatへのデバッグ・レベルでのメッセージ出力となっています。

発展 学習

メッセージと一緒に Log を出力しよう



シメッセージと一緒にWebページを開こう

メッセージやログを出力するだけでは、物足りませんか? Android には、他のアプリとの連携を行う Intent という仕組みがあります。 ここでは、Intent を使って Webブラウザで Google を開いてみましょう。 *(Intentの仕組みや概念などは、長くなるので触れません…ごめんなさい)*

それでは、演習5)のボタン押下時の onClick メソッドでの Toast を使ったメッセージ出力の下に、下記のWebブラウザ連携を追加してください。

Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://google.co.jp/"));
startActivity(intent);

アプリを実行して、ボタン押下とともにブラウザが起動することを確認しましょう。

メッセージと一緒にWebページを開こう 学習

発展

89 90	//上記の内容を自分で記述してみましょう。 */)c	🗅 java 🔪 💼 com 🔪 💼 example 🔪 💼 hellopractice 🔪 💼 hellopracticeapplic
91	Button button = (Button) view.findViewById(R.id. <i>button</i>);			c HelloPracticeActivity.java × 🙍 fragment_hello_practice.xml ×
92 🕞	<pre>button.setOnClickListener(new View.OnClickListener() { @Override</pre>	C	dS	package com.example.hellopractice.hellopracticeapplication
94 o †	<pre>public void onClick(View v) {</pre>			3 pimport android.app.Activity;
95	Toast. <i>makeText</i> (getActivity(), "クリック!" , Toast. <i>LENGTH_LONG</i>).show();			4 import android.app.ActionBar;
96 97 98	Log.d("Practice", "クリック・ログ出力!");			b inport android.app.Fragment; 6 import android.content.Intent; 7 import android.net.Uri;
99	Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://google.co.jp"));			import android os. Bundle; IntentやUriの
100 101	startActivity(intent); }			10 import android.view.LayoutInflat 11 import android.view.Menu: 11 にも注意
102 🔶	});			12 import android.view.MenuItem;





アプリのコード変更を確認しよう

この演習サンプルは、皆さんも新規作成した HelloWorld アプリを元に作られています。 参考用にオリジナルの HelloWorld アプリのソース・ファイルを original_copy に保管しました。 (オリジナルと現在のファイルの区別がつくよう、オリジナルの拡張子を .original にしています。)

Android Studio では、ファイル内容の比較もできますので、オリジナルとの変更を確認しましょう。

例 Activyti の変更をチェックする場合

- 1. Project ツールウィンドウで HelloPracticeActivity.java を選択します。
- 2. CTRL キーを押しながら HelloPracticeActivity.java.original を選択して、右クリックします。
- ポップアップ・メニューから、Compare Two File を選択します。
 ファイル比較ダイアログが開き、両者の違いが表示されます。



アプリのコード変更を確認しよう

BelloPracticeApplication - [C:¥Users¥rie¥AndroidStudioProjects¥HelloPracticeApplication] - [app]¥app¥src¥main¥java¥c □ ■ ■ D # の (
Image: specific sector with the specific sector sect	① ○ ↑ ↓ Ignore whitespace: Do not ignore ▼ Highlight: By word ▼ ? ▷ ♦- HelloPracticeActivity.java (C:¥Users¥rie¥AndroidStudioProjects¥HelloPrac HelloPracticeActivity.java.original (C:¥Users¥rie¥AndroidStudioProjects LF CRLF				
♥ © com.example.hellopractice.hells	package com.example.hellopractice.hellopractice.pplication; 1 1 package com.example.hellopractice.hellopracte.hellopractice.hellopracter.hellopractice.hellopractice				
activity_hello_print activity_hello_print app_build.gradle.cd app_build.gradle.cd fragment_hello_prictice.xml.original app_build.gradle.cd fragment_hello_prictice.xml.original fragment_hello_prictice.xml.original belief Compare Two Files を右ポップアップで選択 Terminal ● §: Android ● 2: Message Compare Two Files Compare Two Files	import android.view.Wiewimentitem, 12 import android.view.Wiewimentitem, 13 import android.view.Viewimentitem, 13 import android.view.Viewimentitem, 14 import android.view.Viewimentitem, 14 import android.view.Viewimentitem, 14 import android.view.Viewimentitem, 15 import android.view.Viewimentitem, 16 import android.widget.Button; 16 import android.widget.Toast; 17 18 18 19 13 20 20 20 super.onCreate(Bundle savedInstanceState) { 4 differences Deleted Changed Inserted				
Compare two selected files					



実機でデバッグしてみる

PCを使ったアプリの動作確認(デバッグ)について

本演習では、エミュレータを使って動作確認(デバッグ)をしてみましたが、 エミュレータの代わりにPCと実機をUSBケーブルでつないでアプリの動作確認(デバッグ)をすることもできます。

ただし、PCと実機をUSBケーブルでつないでデバッグを行う場合、 「Android USBドライバ」と「開発者向けオプションでのUSBデバッグ指定」が必要になります。 上記の補説については、次ページ以降を御参照ください。



実機でデバッグしてみる

Android USB デバイスドライバのインストールについて

- 1. Windows 7 の場合
 - 実機とPCをUSBで接続した際に、「ドライバが見つかりません。」と通知された場合の対処例 ネット検索などで実機用のAndroid USBドライバがあった場合は、そちらを優先してください。 残念ながら、全ての実機が下記方法で対処できるとは限りません。
 - 1 コントロールバネルから、システムのデバイスマネージャーを開きます。
 - デバイスマネージャーで、実機が「ほかのデバイス」(ドライバ不明のデバイス)となっていたら、 USBドライバのインストールを行うため、ドライバーソフトウェアの更新ダイアログを開きます。
 - 3-1 ドライバーソフトウェアの検索を利用する場合は、 ドライバー検索場所を Android Studio インストール先¥sdk¥extras¥google¥usb_driver に指定して、 USBドライバーのインストールを行います。
 - 3-2 デバイスドライバーの一覧を利用する場合は、
 Android Device を選択して「次へ」クリックで、ドライバーソフトウェアの更新ダイアログに進み
 Android Composite ADB Interface を選択して、
 USBドライバーのインストールを行います。





Android USB デバイスドライバのインストールについて

2. Linux の場合

実機とPCをUSBで接続してコンソールで、adb devices を実行した際のリストが空の場合は、 Android Developers > tools > Using Hardware Devices に従って、 /etc/udev/rules.d/51-android.rules ファイルに実機のUSB Vender Id 行を追加してください。



実機でデバッグしてみる

開発者向けオプションでのUSBデバッグ指定について

- 実機がICS(Android 4.0.3)以前の場合 Android の設定画面に「開発者向けオプション」という 開発者(デベロッパー)用の設定項目が標準でありますので、 「開発者向けオプション」から、デバッグ項目にある「USBデバッグ」のチェックをONにしてください。
- 実機がJelly Bean (Android 4.1, 4.2, 4.3)以降の場合
 「開発者向けオプション」は、一般の人が間違って利用しないよう初期状態だと非表示になりました。

「開発者向けオプション」を表示するには、

Android 設定画面から「端末情報」を開き「ビルド番号」を連続タップしてください。 連続タップを続けると「これでデベロッパーになりました!」と表示され開発者(デベロッパー)になります。

上記の操作以降は、Android の設定画面に「開発者向けオプション」が表示されますので、 「開発者向けオプション」から、デバッグ項目にある「USBデバッグ」のチェックをONにしてください。



プロジェクトのビルドについて

- プロジェクトのビルド ナビゲーション・ツールバーの Make Project ボタンをクリックして、プロジェクトのビルドを行いましょう。 画面下部のツールウィンドウに Gradle Console が開き BUILD SUCCESSFUL と表示されましたら、 プロジェクトのビルドに成功です。
- トラブルシュート ビルドに失敗しましたら、先ずは、Gradle Console のエラーメッセージを確認してください。 適切に設定したつもりなのにビルドに失敗したときは、 以下を試してみてください。
 - Sync Project With Gradle File ボタンをクリックして、 プロジェクト構成設定とGradle設定の同期をとってみる。
 - 2. メニュー > Build > Clean Project 選択して、プロジェクトを初期化してみる。
 - 3. メニュー > Invalidate Caches / Restart... を選択して、キャッシュをクリアして再起動してみる。 (要するに「リセットボタンを押す」ですね。 これは、エラーやフリーズなど、ビルド以外にも応用できます)
 - バージョンアップを待ちましょう
 ネット検索で類似の問題がないかを検索して、解決法が見つかることもありますが、
 Android Studio は、まだBETA版ですのでバグの場合もあります…。

おさらい

今日できるようになったこと

- Android Studioを使って、Androidのアプリを 新規作成・実行できるようになった。
- サンプルコードをインポートできるようになった。
- レイアウトをいじったりボタンを配置して クリック動作を定義できるようになった。

おつかれさまでした

演習資料(2) Android Studio でのアプリ開発

本演習は、これで終了です

おつかれさまでした


本演習資料作成にあたり、以下のサイトを参考にしました。

1. Android Studio 全般 Android Studio最速入門~効率的にコーディングするための使い方 <u>http://gihyo.jp/dev/serial/01/android_studio</u>