

Android 入门基础

周一航

字节跳动 Android 工程师

zhouyihang.cmmsz@bytedance.com



目录

- Android历史
- Android系统架构
- Android应用程序
- 开发第一个Android应用程序

Android历史





前智能机时代

- 功能机
- “伪智能机”
 - Symbian
 - Windows Phone (WinCE)
 - Black Berry
 - Palm OS (WebOS)

智能机元年

□ 2007年1月

iPhone问世



Today Apple is going to reinvent phone.

-- Steve Jobs

智能机元年

□ 2008年9月

第一台Android手机
上市



ANDROID
COMMUNITY.COM

HTC Dream (T-Mobile G1)

Android版本历史 1.1 Bender (2008)

- 应用市场
- Widget
- 通知



Android版本历史



Android版本历史 1.5 CUPCAKE (2009)

- 新的版本命名规则
- 软键盘



Android版本历史 1.6 DONUT (2009)

- CDMA
- 支持多屏幕尺寸



Android版本历史 2.0 ECLAIR (2009)

- Google地图
- HTML5浏览器



Android版本历史 2.2 FROYO (2010)

- WiFi热点
- Push推送



Android版本历史 2.3 Gingerbread (2010)

□ 前置摄像头



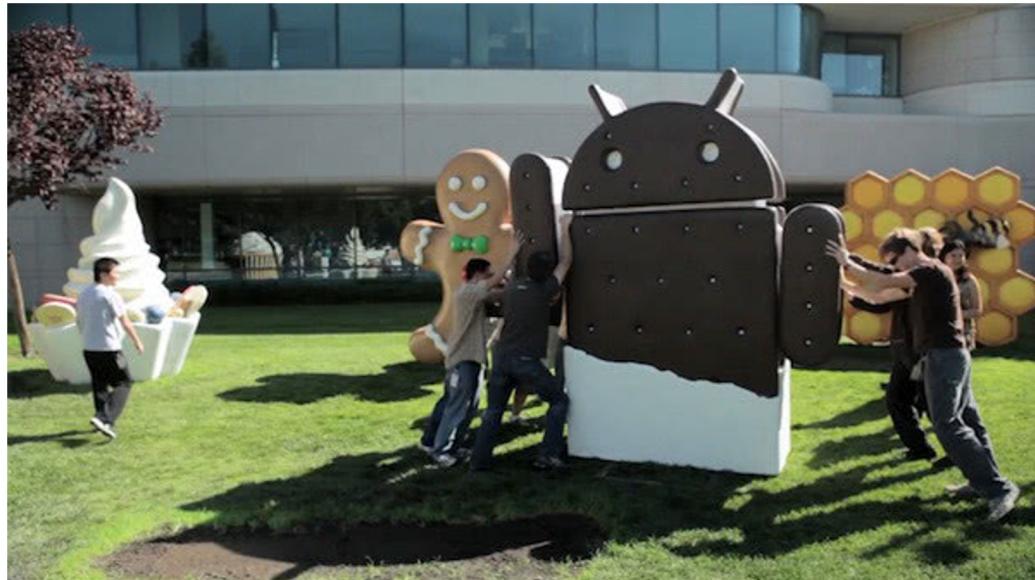
Android版本历史 3.0 HONEYCOMB (2011)

□ 支持平板电脑



Android版本历史 4.0 ICE CREAM SANDWICH (2011)

- 同时支持平板和手机
- 全新的内置apps



Android版本历史 4.1/2 JELLY BEAN (2012)

- 流畅的滑动体验 (黄油般丝滑)
- 多账号(平板)



Android版本历史 4.4 KITKAT (2013)

- 速度更快
- 性能更好
- 低端机更友好



Android版本历史 5.0 LOLLIPOP (2014)

- Material Design
- ART VM
- 双卡支持



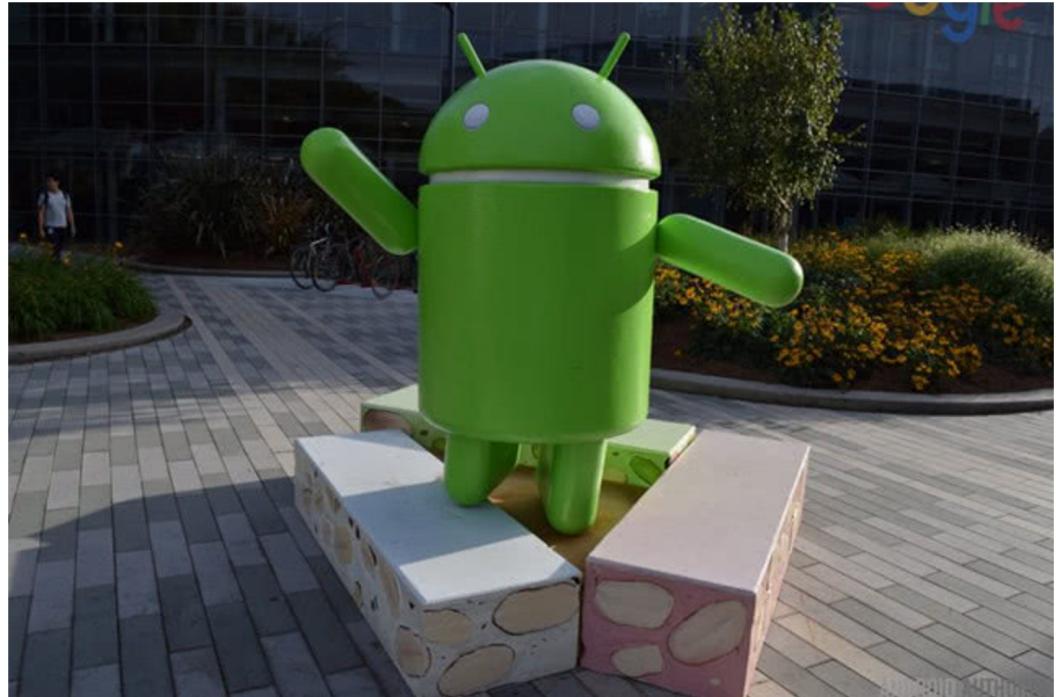
Android版本历史 6.0 MARSHMALLOW (2015)

- 成熟的权限控制



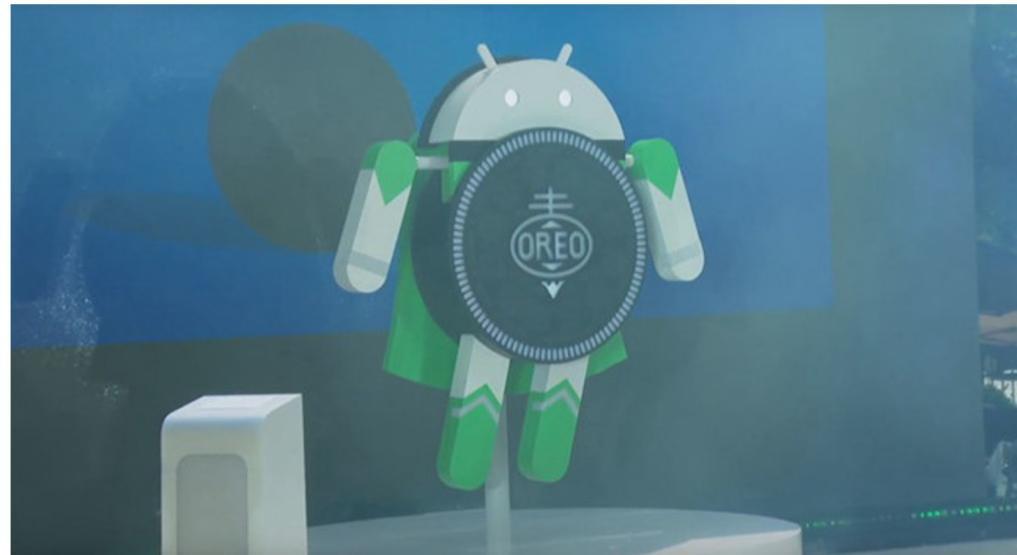
Android版本历史 7.0 NOUGAT (2016)

- 分屏
- 改进的多任务



Android版本历史 8.0 OREO (2017)

- 画中画
- 自适应图标



Android版本历史 9.0 PIE (2018)

- 刘海屏
- 增强通知栏





Android版本历史 10.0 – Q(2019)

- 折叠屏
- 隐私增强



Android版本历史 11.0 (2020)

- 隐私功能
- 增加新功能和API（用于
5G、共享、连接、媒体、
NNAPI、生物识别等方面
的 API ）



Android生态现状

- Apps数量和下载量领先
- 领先势头还在增大
- 碎片化严重(逐步解决)
 - 版本
 - 屏幕尺寸
 - 硬件能力

Android系统架构



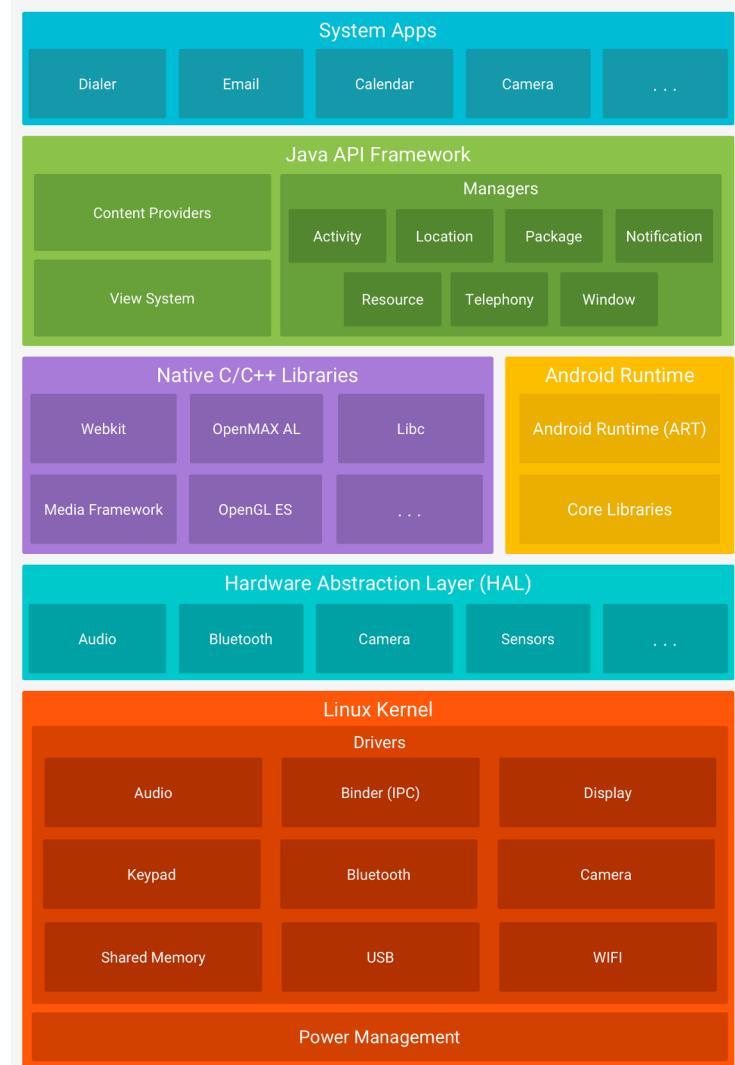
系统结构图

- ❑ Linux内核
- ❑ HAL
- ❑ Android Runtime
- ❑ C/C++ Libraries
- ❑ Java API Framework
- ❑ Apps

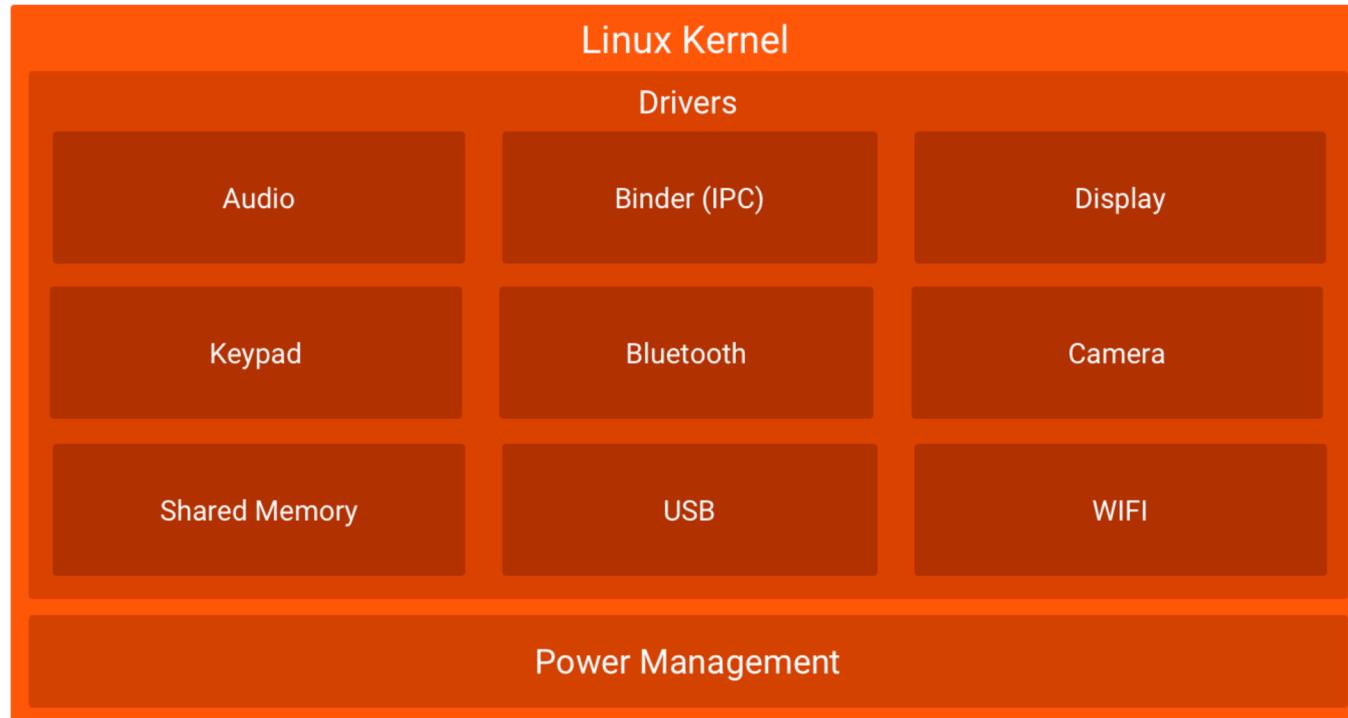
Android应用

Android系统

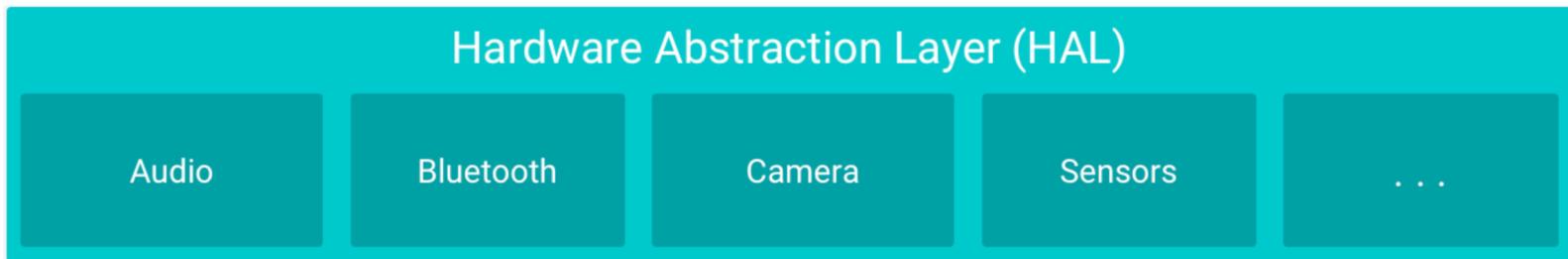
Linux内核



Linux内核



HAL硬件抽象层



Native Libraries & JVM



Framework

Java API Framework

Content Providers

View System

Managers

Activity

Location

Package

Notification

Resource

Telephony

Window



System Apps

System Apps

Dialer

Email

Calendar

Camera

...

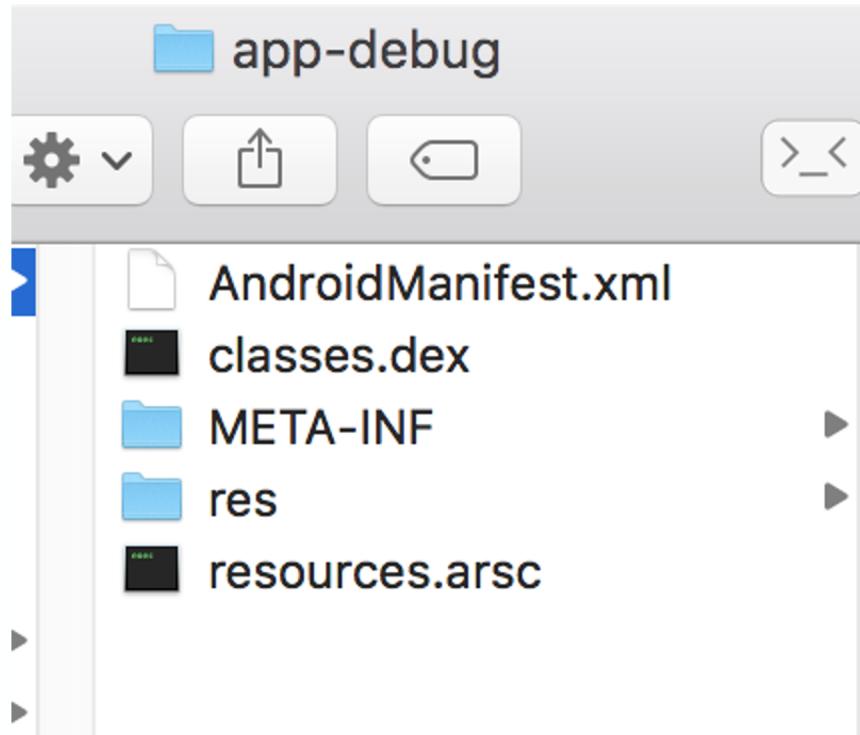
Android应用程序





APK

- APK (Android Package Kit)
 - 本质是一个ZIP文件
 - AndroidManifest
 - Dex
 - resources



沙箱技术

- 基于Linux用户权限管理
- 每个应用一个独立用户ID
- 每个进程有自己的VM
- Apps之间无法“直接”访问数据

应用组件

□ Activity

□ 页面容器

□ Service

□ 后台常驻任务

□ Content Provider

□ 数据提供

□ Broadcast Receiver

□ 广播消息接收器



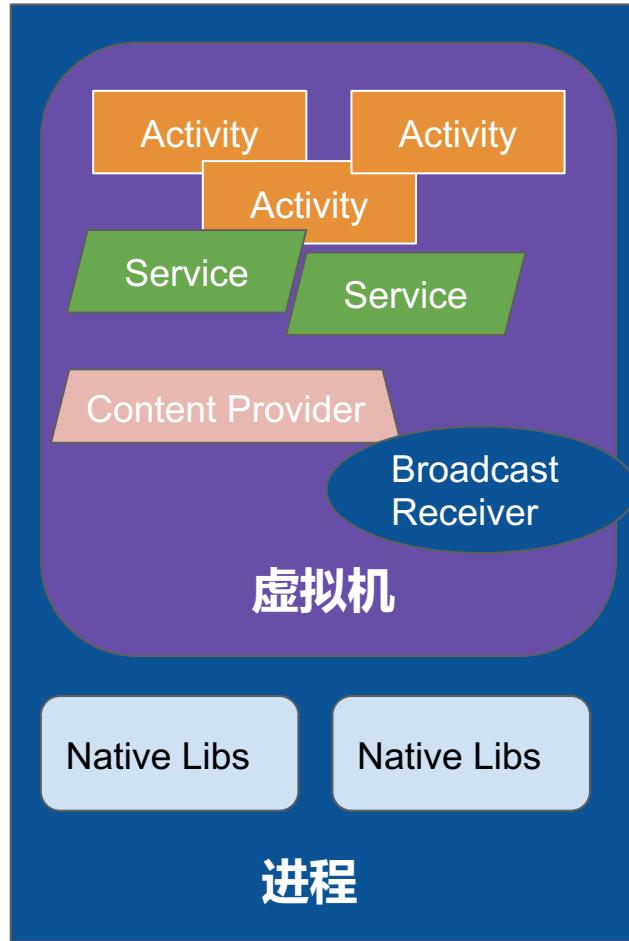
问题

□ 如果要实现一个音乐播放的APP，可以用到哪些组件？



问题

□ 应用组件，进程，虚拟机....这些都是什么关系？





问题

□ 四大组件中有哪些不是必须的？为什么？

开发第一个 **Android**应用程序

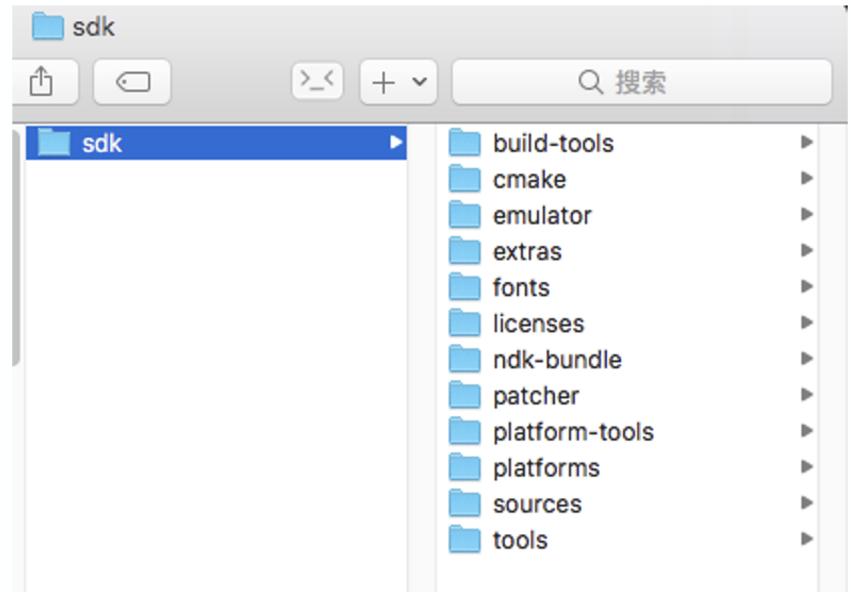


Android Studio

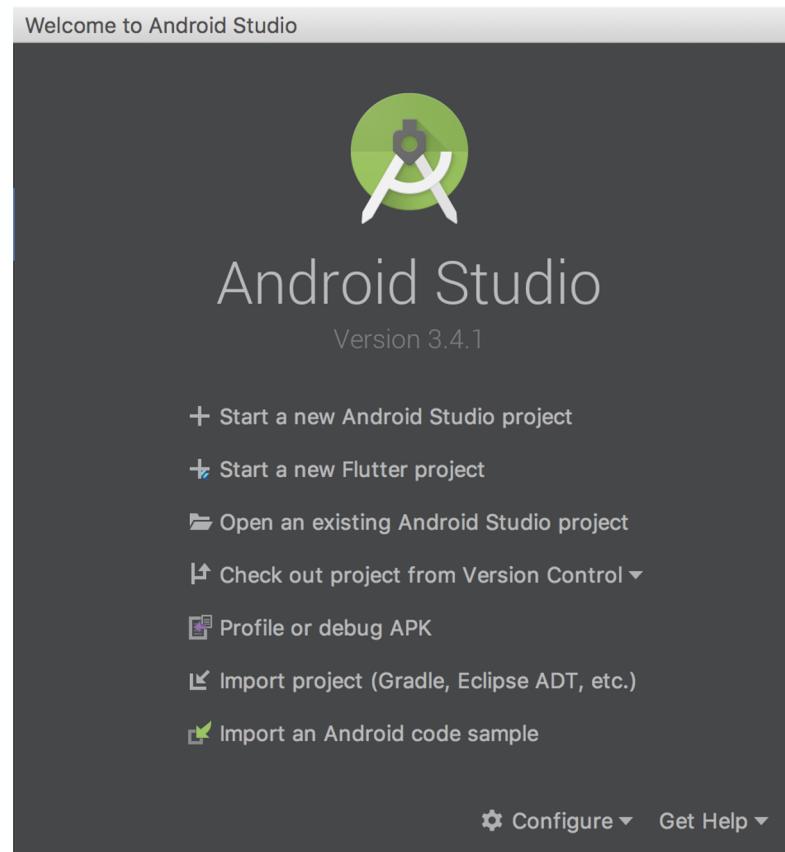
- 下载地址 <https://developer.android.google.cn/studio/>
 - IDE
 - JDK
 - Android SDK
- 配置环境

Android SDK

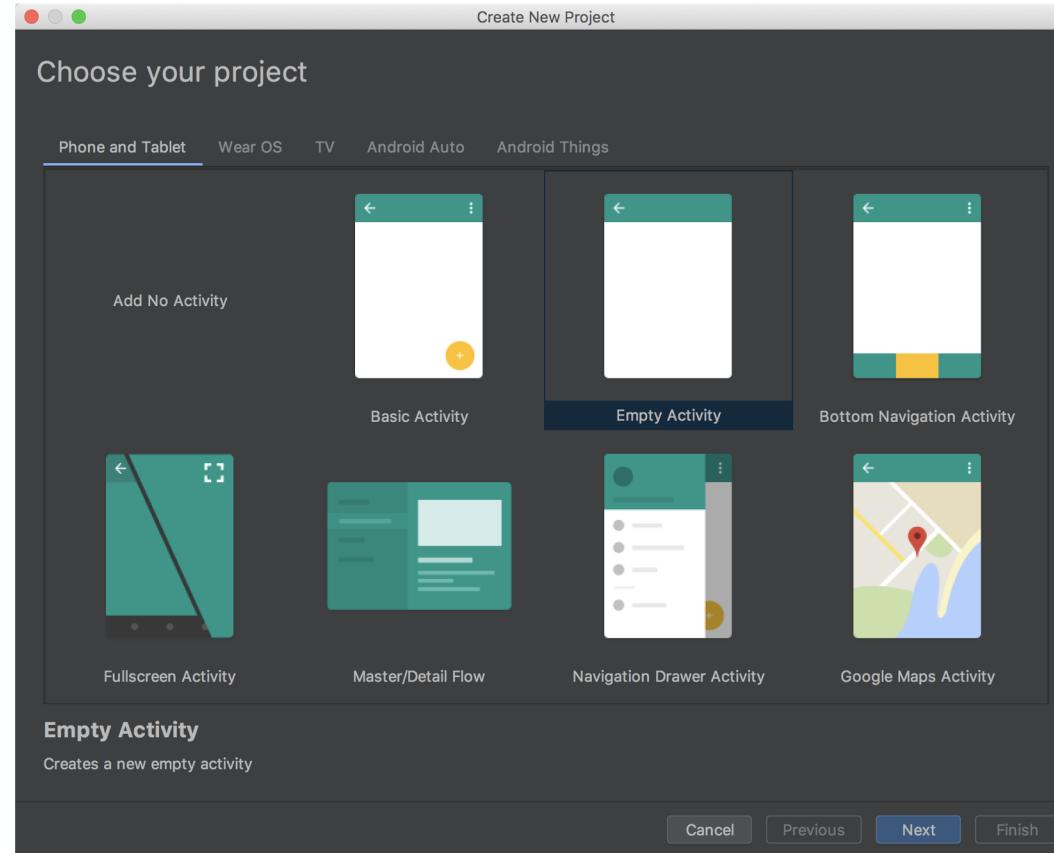
- ❑ SDK Platforms (API)
 - ❑ 有版本区分
- ❑ Build-Tools
 - ❑ Aapt, aidl, d8, apksigner
- ❑ Platform-Tools
 - ❑ Adb, sqlite
- ❑ Emulator



创建第一个Android应用



创建第一个Android应用



Configure your project

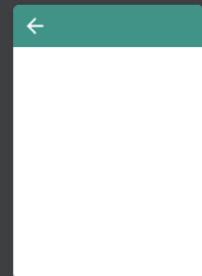
配置Project

Language

Java

Kotlin

Java



Empty Activity

Creates a new empty activity

Name

HelloWorld

Package name

com/bytedance/component/helloworld

Save location

/Users/xulu/xllxtt/camp/Test

Language

Java

Minimum API level

API 14: Android 4.0 (IceCreamSandwich)

Your app will run on approximately **100%** of devices.

[Help me choose](#) This project will support instant apps Use androidx.* artifacts

Cancel

Previous

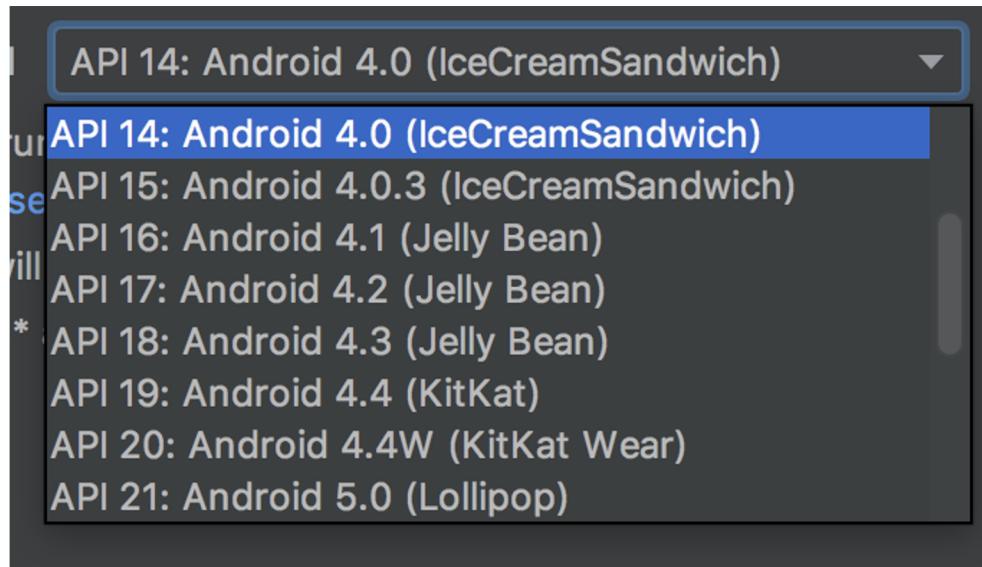
Next

Finish

系统如何区分应用的？

- applicationId (Package Name)
 - apk在系统中的唯一标识
- 签名
 - 验证apk是否来自同一个开发者
- 两个机制同时使用就确保了开发者对于apk的所有权

选择min SDK版本号



关于版本号的三个概念

□ Min SDK version

□ 能跑的最小手机版本号（一般为15左右）

□ Compile SDK version

□ 编译打包用的版本号（一般越新越好）

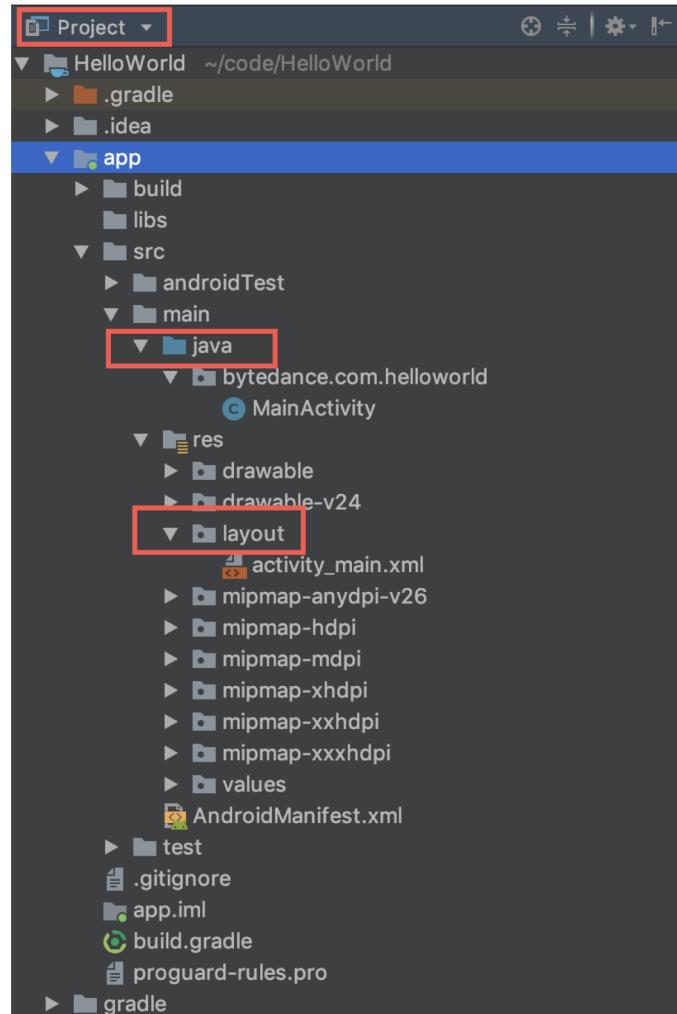
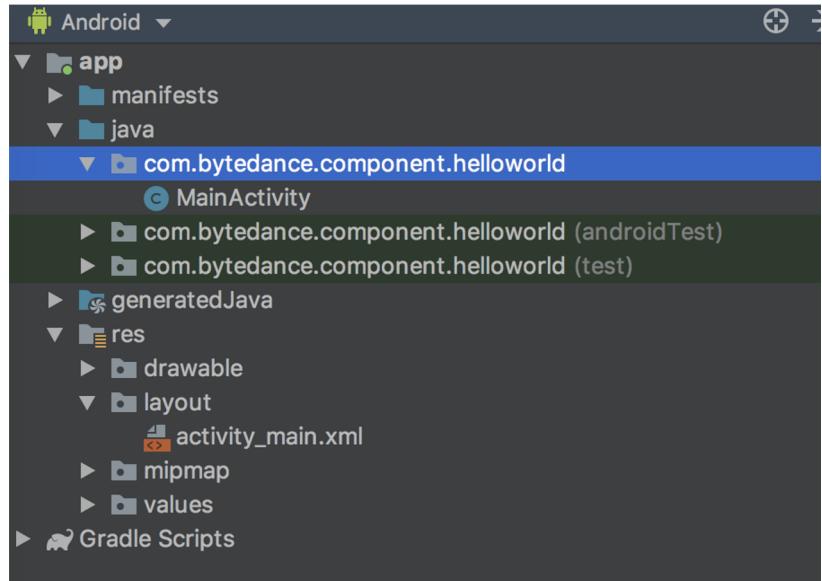
□ Target SDK version

□ 应用运行时的行为表现

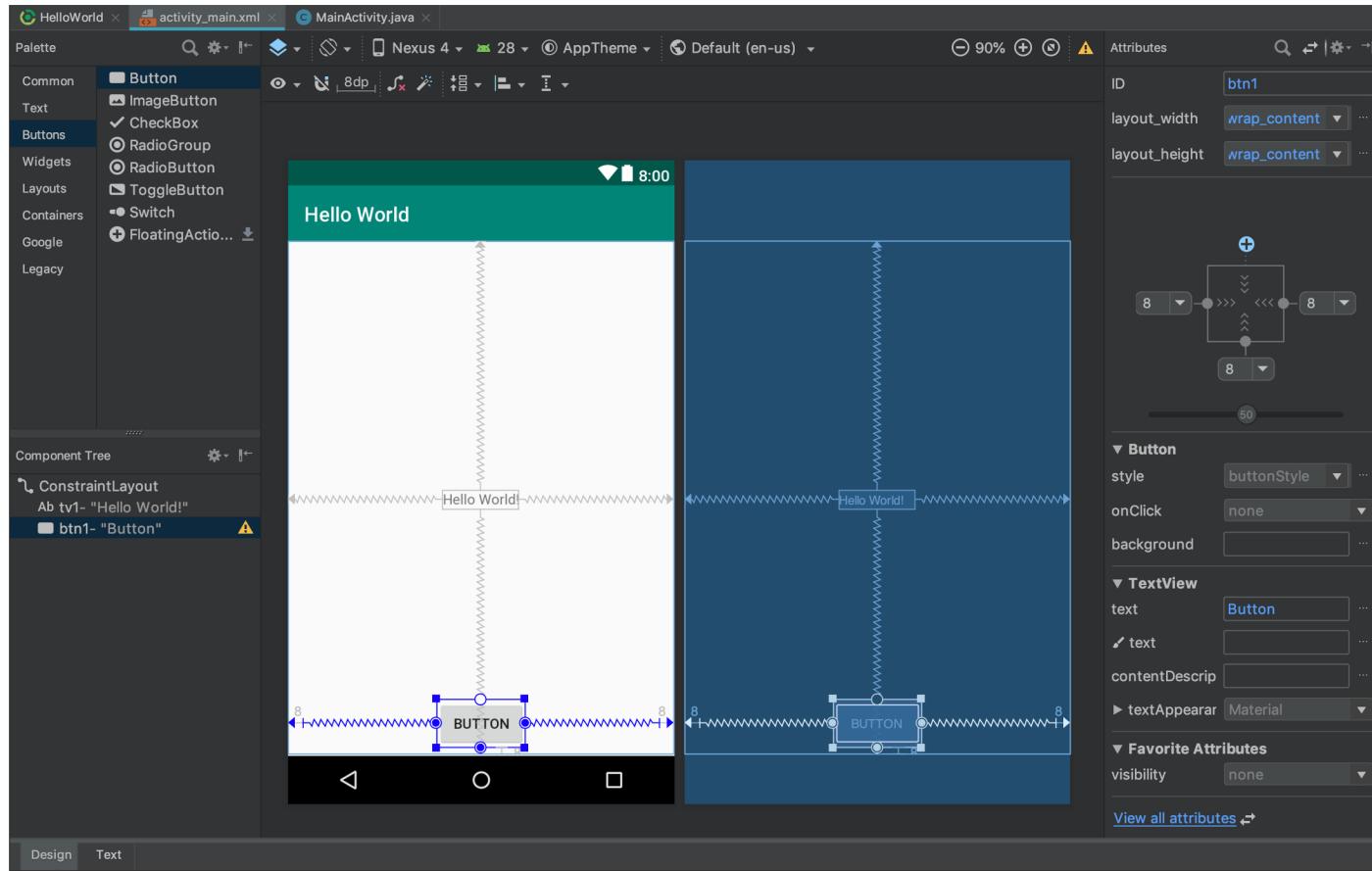
App代码结构

□ Java

□ Resource



页面布局

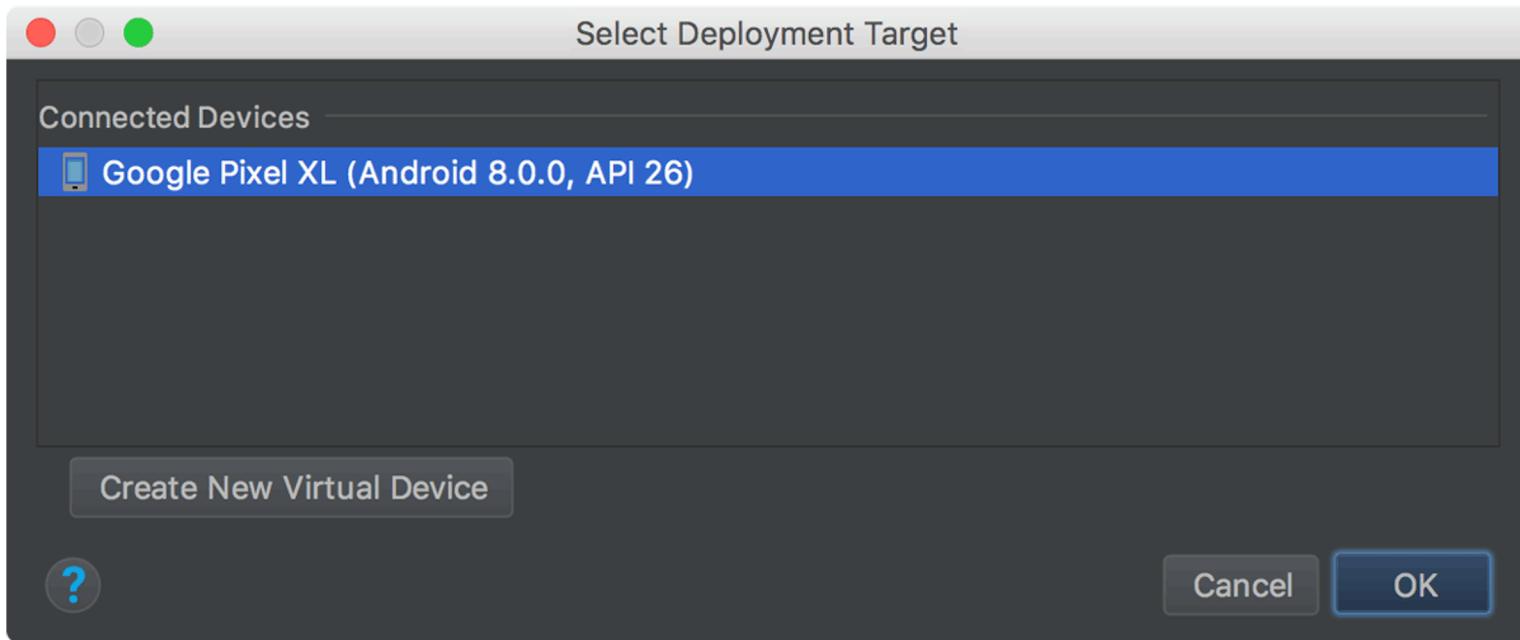


运行App

- 把手机和电脑使用USB线连接
- 安装USB驱动
- 手机上开启“开发者模式”
 - 系统设置->系统->关于手机->build number ->连续点击5次（不同手机系统可能会有少许区别）
 - 系统设置->开发者选项->开启开发者选项 / USB调试
 - 第一次连接手机，会弹出是否允许调试的对话框，点击允许

运行App

- 在Android Studio工具栏中点击 Run 

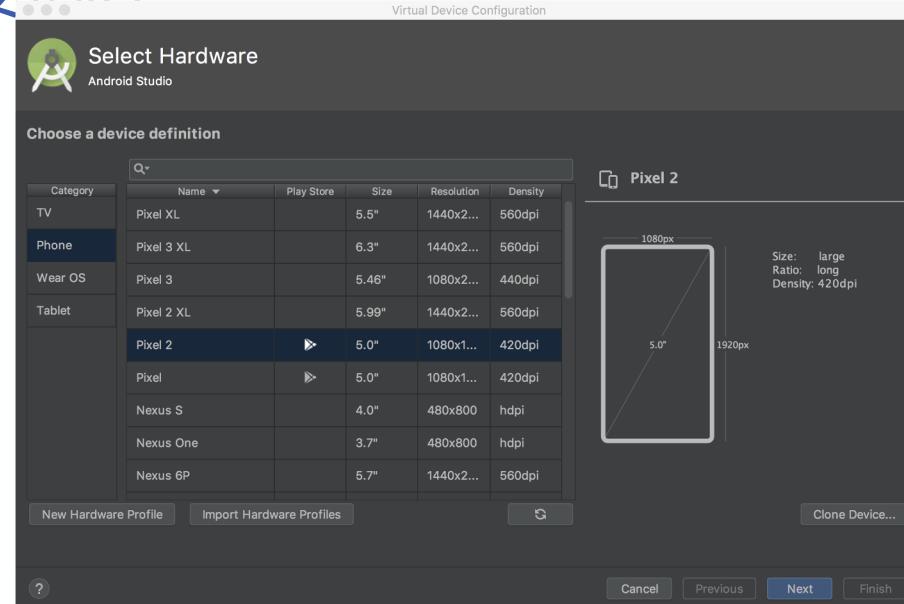


运行虚拟机

□ 如果没有Android手机也不要担心

可以使用虚拟机

□ Android Studio -> Run ->
Create New Virtual Device



下载虚拟机

Virtual Device Configuration

System Image

Android Studio

Select a system image

Recommended	x86 Images	Other Images	
Release Name	API Level ▾	ABI	Target
Q Download	29	x86	Android 9.0 (Google APIs)
Pie Download	28	x86	Android 9.0 (Google APIs)
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	26	x86	Android 8.0 (Google APIs)
Nougat Download	25	x86	Android 7.1.1 (Google APIs)
Nougat Download	24	x86	Android 7.0 (Google APIs)
Marshmallow Download	23	x86	Android 6.0 (Google APIs)
Lollipop Download	22	x86	Android 5.1 (Google APIs)

Oreo

API Level
26

Android
8.0

Google Inc.

System Image
x86

Recommendation

No emulator installed

Install Emulator

We recommend these images because they run the fastest and support Google APIs.

Questions on API level?

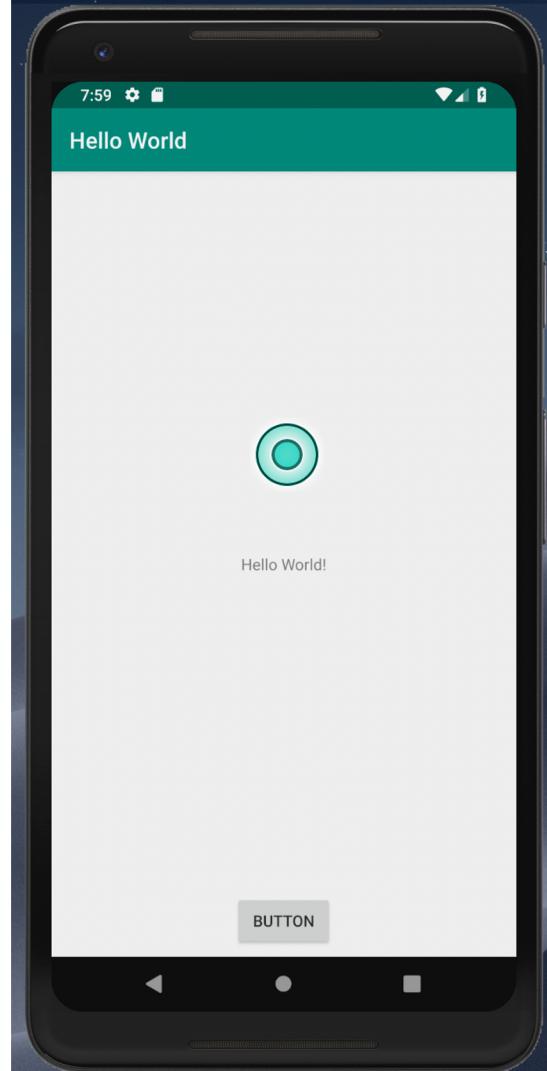
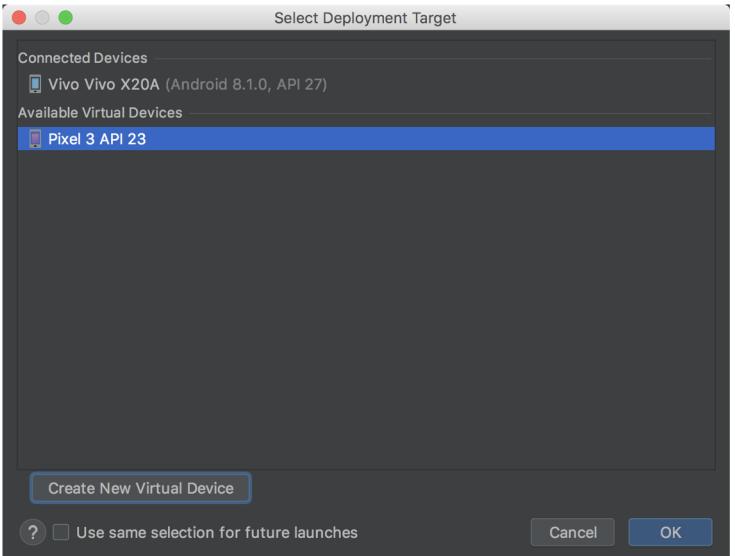
Cancel Previous Next Finish

! A system image must be selected to continue.

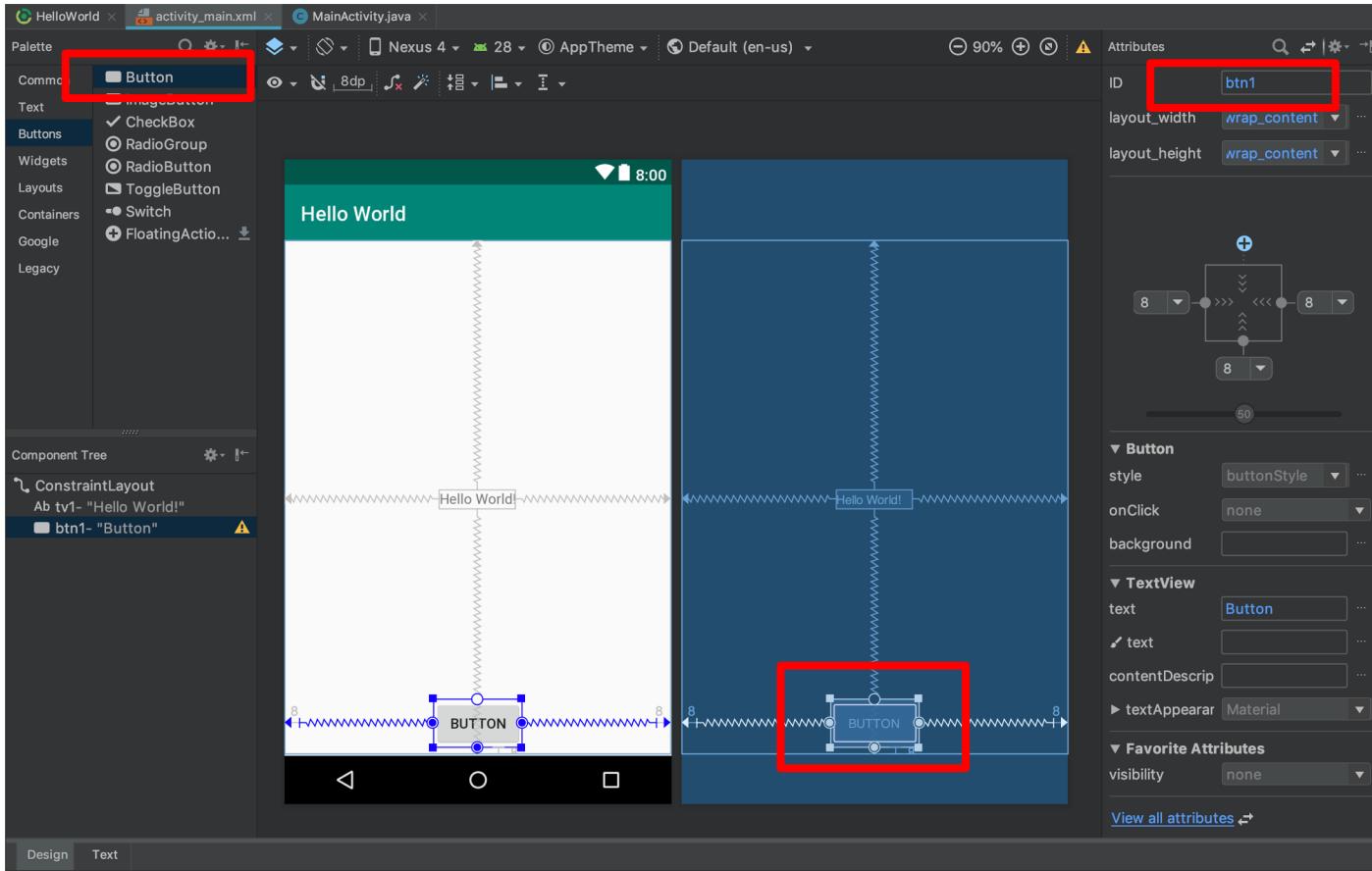
下载虚拟机



运行虚拟机

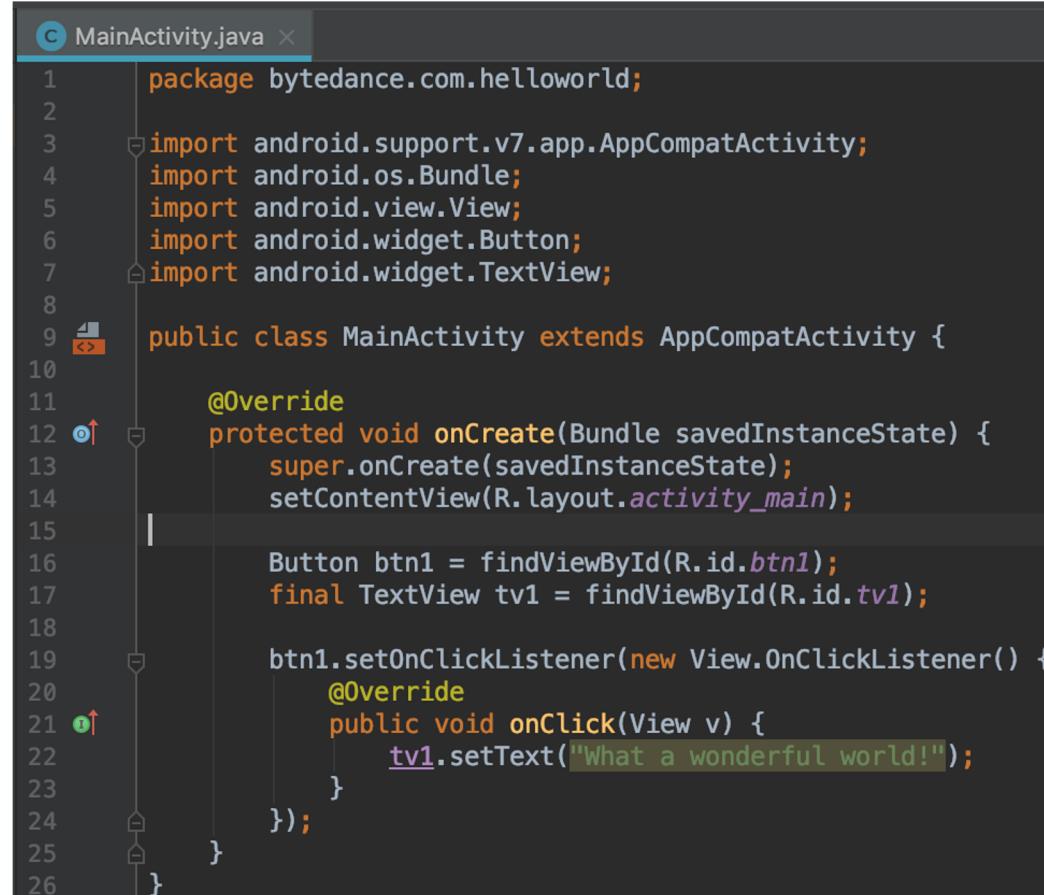


在页面布局中添加控件



在代码中操作控件

- 在代码中引用控件
 - `findViewById()`
- 设置触摸监听器
 - `setOnClickListener()`
- 设置文字
 - `setText()`

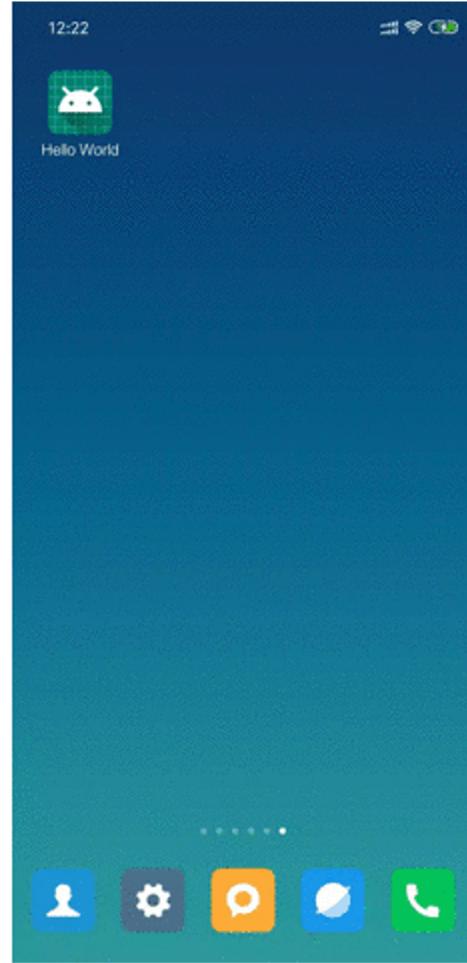


The screenshot shows the code for `MainActivity.java` in an Android project. The code is as follows:

```
1 package bytedance.com.helloworld;
2
3 import android.support.v7.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.TextView;
8
9 public class MainActivity extends AppCompatActivity {
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         Button btn1 = findViewById(R.id.btn1);
17         final TextView tv1 = findViewById(R.id.tv1);
18
19         btn1.setOnClickListener(new View.OnClickListener() {
20             @Override
21             public void onClick(View v) {
22                 tv1.setText("What a wonderful world!");
23             }
24         });
25     }
26 }
```

The line `tv1.setText("What a wonderful world!");` is highlighted in green, indicating it is selected or being edited.

运行效果

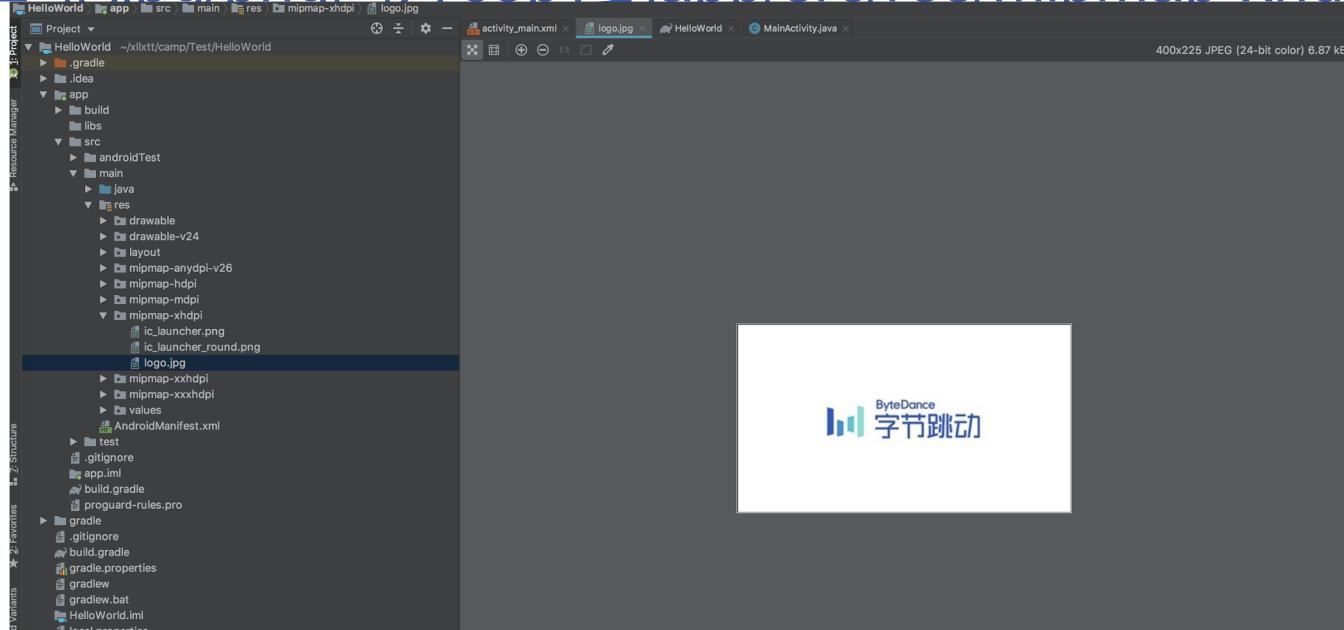




恭喜你，第一个App成功运行啦！

为App中添加图片资源

- 把ipa/png图片copy到app/src/res/mipmap-xhdpi目录



为App中添加图片资源

- 在layout中添加 ImageView控件
- 选择Image Resource

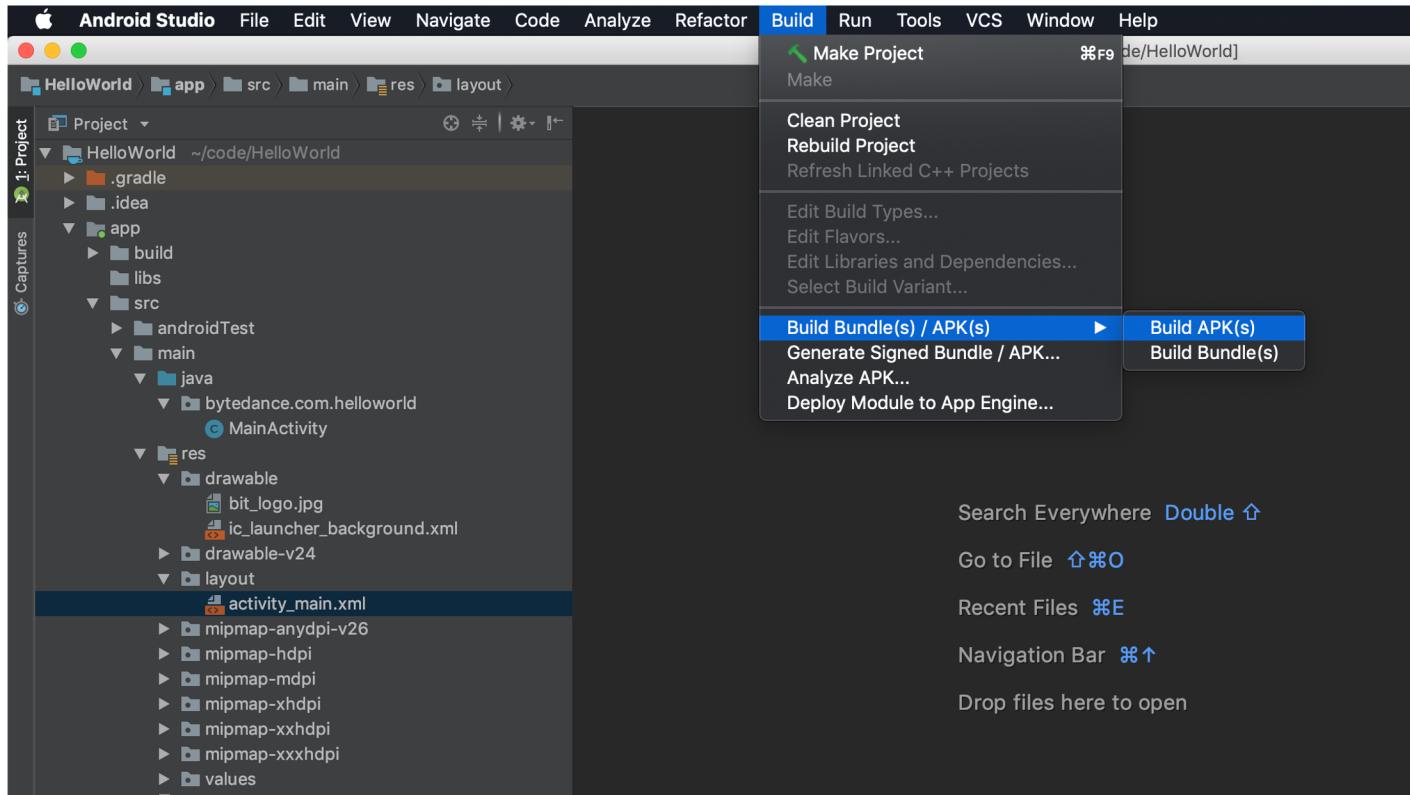
```
<ImageView  
    android:id="@+id/imageView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="8dp"  
    android:layout_marginLeft="8dp"  
    android:layout_marginTop="8dp"  
    android:layout_marginEnd="8dp"  
    android:layout_marginRight="8dp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@mipmap/logo" />
```

显示效果



ByteDance字节跳动

打包生成apk文件



ce字节跳动

必备技能

必备技能 - 日志

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** Shows the project directory structure with files like activity_main.xml, AndroidManifest.xml, test, .gitignore, app.iml, build.gradle, proguard-rules.pro, and gradle.
- Code Editor:** Displays the MainActivity.java code. It contains an onCreate() method that sets the content view to activity_main, finds a button and text view, sets the button's click listener, and logs "hello" and "world".

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button btn1 = findViewById(R.id.btn1);
        final TextView tv1 = findViewById(R.id.tv1);

        btn1.setOnClickListener(v -> {
            tv1.setText("What a wonderful world!");
            Log.d( tag: "MainActivity", msg: "world");
        });

        Log.d( tag: "MainActivity", msg: "hello");
    }
}
```
- Logcat:** Shows log entries from the Emulator Pixel_2_XL_API. The logcat interface includes filters for device (Emulator Pixel_2_XL_API), application (bytedance.com.helloworld), and log level (Verbose). A red box highlights the application filter field. Other visible fields include Regex and Show only selected application.

Date	Time	Thread	Log Level	Message
2019-01-17	00:22:09.774	21426-21426	bytedance.com.helloworld D/MainActivity:	hello
2019-01-17	00:22:37.167	21426-21426	bytedance.com.helloworld D/MainActivity:	world
2019-01-17	00:22:38.682	21426-21426	bytedance.com.helloworld D/MainActivity:	world
2019-01-17	00:22:39.421	21426-21426	bytedance.com.helloworld D/MainActivity:	world
2019-01-17	00:22:40.373	21426-21426	bytedance.com.helloworld D/MainActivity:	world
2019-01-17	00:22:41.957	21426-21426	bytedance.com.helloworld D/MainActivity:	world
2019-01-17	00:22:43.248	21426-21426	bytedance.com.helloworld D/MainActivity:	world
- Bottom Navigation Bar:** Includes icons for Run, TODO, Logcat (highlighted with a red box), Profiler, Terminal, Build, and Event Log.

必备技能 - adb

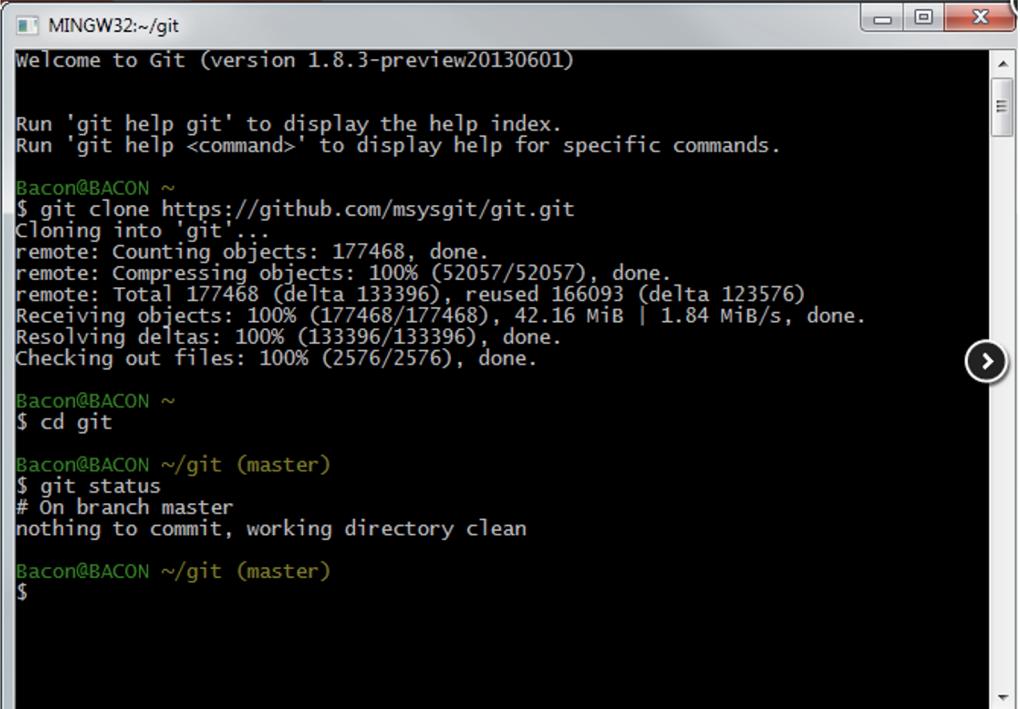
□ Android Debug Bridge

- 一种调试android的工具
- 把Android设备和开发主机连接起来
- 支持USB连接或者网络连接
- 可以提供console，执行shell命令
- 可以提供文件传输能力

```
→ HelloWorld adb shell
root@generic_x86:/ # ls
acct
cache
charger
config
d
data
default.prop
dev
etc
file_contexts
fstab.goldfish
fstab.ranchu
fstab.ranchu-encrypt
init
init.environ.rc
init.goldfish.rc
init.ranchu-common.rc
init.ranchu-encrypt.rc
init.ranchu.rc
init.rc
```

必备技能 - git - 安装

- Git 下载官网
- 同步安装 git-bash
- 如果需要 Client



```
MINGW32:~/git
Welcome to Git (version 1.8.3-preview20130601)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

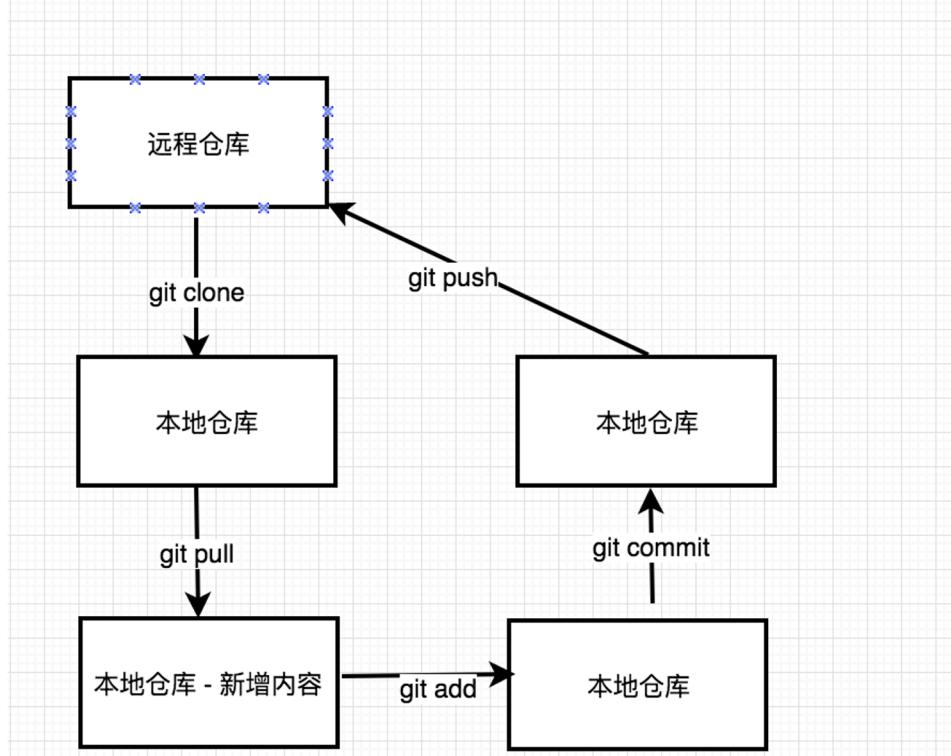
Bacon@BACON ~
$ git clone https://github.com/msysgit/git.git
Cloning into 'git'...
remote: Counting objects: 177468, done.
remote: Compressing objects: 100% (52057/52057), done.
remote: Total 177468 (delta 133396), reused 166093 (delta 123576)
Receiving objects: 100% (177468/177468), 42.16 MiB | 1.84 MiB/s, done.
Resolving deltas: 100% (133396/133396), done.
Checking out files: 100% (2576/2576), done.

Bacon@BACON ~
$ cd git

Bacon@BACON ~/git (master)
$ git status
# On branch master
nothing to commit, working directory clean

Bacon@BACON ~/git (master)
$
```

必备技能 - git - 命令



必备技能 - git

- git clone <url> 克隆一个仓库项目到本地
- git pull 从仓库更新代码
- git checkout <branch name> 切换分支
- git add <filename> 添加文件
- git commit 创建提交纪录
- git push origin <remote branch> 推送提交纪录到仓库

资料速查手册

□ Google Android官方中文站点

- <https://developer.android.google.cn/>

□ Android API速查

- <https://developer.android.google.cn/reference/packages>

□ Android View Api

- <https://developer.android.com/reference/android/view/View>

HomeWork



现场练习

- 配置运行环境，从仓库 clone 到本地（包含 demo 和讲义）
- 将 demo 运行起来，了解 demo 配置的环境
- 仓库地址：

<https://github.com/zjubytedancer/bytedance-android-camp-zju-2021/tree/main/chapter-1>

现场练习

- 创建一个Android app工程，包含一个activity（环境可以参考 demo）
- 使用5种以上的View并实现一些简单交互
 - ImageView, Button, TextView, RadioButton, CheckBox, EditText, ProgressBar, SeekBar, Switch 等等不限
- 将一些交互结果输出log
- 打包生成apk

现场练习

- 注册一个github账号（已有就可以跳过此步）
- 创建一个仓库
- 把练习代码上传到github仓库里
- 当场检验 or 发邮件
 - 给 zhouyihang.cmmsz@bytedance.com
 - 标题“浙大android课程”
 - 内容写上你的姓名，学号和git仓库地址

QA





THANKS

