Android Slides + Documentation Notes

dimens.xml	storing dimensional values as dp, sp, pt, px, mm, in.
Manifest.xml	Every app project must have an AndroidManifest.xml file (with precisely that name) at the root of the project source set. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play. Contents: ● The app's package name ● The components of the app, which includes all activities, services, broadcast receivers, and content providers. ● The permissions that the app needs in order to access protected parts of the system or other apps. ● The hardware and software features the app requires, which affects which devices can install the app from Google Play.
xmlns:android="..."	Instead of calling android:id, the xml will use http://schemas.android.com/apk/res/android:id to be unique. Generally this page doesn't exist (it's a URI, not a URL), but sometimes it is a URL that explains the used namespace.
Context	Interface to global information about an application environment. This is an abstract class whose implementation is provided by the Android system. It allows access to application-specific resources and classes, as well as up-calls for application-level operations such as launching activities, broadcasting and receiving intents.
onPause vs. onStop	If you can still see any part of it (Activity coming to foreground either doesn't occupy the whole screen, or it is somewhat transparent), onPause() will be called. If you cannot see any part of it, onStop() will be called.
URI vs. URL	Your name could be a URI because it identifies you, but it couldn't be a URL because it doesn't help anyone find your location. On the other hand, your address is both a URI and a URL because it both identifies you *and* it provides a location for you.

getContentResolver().query()	`cursor = getContentResolver().query(`
	`    UserDictionary.Words.CONTENT_URI,   // The content URI`
	`    projection,                        // Selected columns`
	`    selectionClause,                   // Selection criteria`
	`    selectionArgs,                     // Selection criteria`
	`    sortOrder);                        // The sort order`

finishActivity(requestCode: Int)	The request code of the activity that you had given to startActivityForResult(). If there are multiple activities started with this request code, they will all be finished.

Explicit intents	specify which application will satisfy the intent, by supplying either the target app's package name or a fully-qualified component class name.
Implicit intents	do not name a specific component, but instead declare a general action to perform, which allows a component from another app to handle it. When you use an implicit intent, the Android system finds the appropriate component to start by comparing the contents of the intent to the *intent filters* declared in the manifest file of other apps on the device. If the intent matches an intent filter, the system starts that component and delivers it the Intent object. If multiple intent filters are compatible, the system displays a dialog so the user can pick which app to use.
intent-filter	Specifies the types of intents that an activity, service, or broadcast receiver can respond to. An intent filter declares the capabilities of its parent component. Most of the contents of the filter are described by its <action>, <category>, and <data> subelements.
Intent.putExtra(key,value)	You can add extra data with various putExtra() methods, each accepting two parameters: the key name and the value.
savedInstaceState	If you save the state of the application in a bundle (typically non-persistent, dynamic data in onSaveInstanceState), it can be passed back to onCreate if the activity needs to be recreated (e.g., orientation change) so that you don't lose this prior information. If no data was supplied, savedInstanceState is null.
Measured width/height	These dimensions define how big a view wants to be within its parent.
(Drawing) width/height	These dimensions define the actual size of the view on screen, at drawing time and after layout. These values may, but do not have to, be different from the measured width and height.
`ViewGroup.LayoutParams`	Every ViewGroup class implements a nested class that extends this class. This subclass contains property types that define the size and position for each child view, as appropriate for the view group.
ContactsContract	ContactsContract defines an extensible database of contact-related information. Contact information is stored in a three-tier data model: ● A row in the Data table can store any kind of personal data, such as a phone number or email addresses. The set of data kinds that can be stored in this table is open-ended. There is a predefined set of common kinds, but any application can add its own data kinds. ● A row in the RawContacts table represents a set of data describing a person and associated with a single account (for example, one of the user's Gmail accounts). ● A row in the Contacts table represents an aggregate of one or more RawContacts presumably describing the same person. When data in or associated with the RawContacts table is changed, the affected aggregate contacts are updated as necessary.

ContactsContract.CommonDataKinds	Container for definitions of common data types stored in the ContactsContract.Data table.
LayoutInflater	Instantiates a layout XML file into its corresponding View objects. It is never used directly. Instead, use Activity.getLayoutInflater() or Context#getSystemService to retrieve a standard LayoutInflater instance that is already hooked up to the current context and correctly configured for the device you are running on.
Fragment.onAttach()	the system calls when adding the fragment to the activity
Sensor Categories	Motion, Position, Environment
Motion Sensor	These sensors measure acceleration forces and rotational forces along three axes. This category includes accelerometers, gravity sensors, gyroscopes, and rotational vector sensors.
Env Sensor	These sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination, and humidity. This category includes barometers, photometers, and thermometers.
Position Sensor	These sensors measure the physical position of a device. This category includes orientation sensors and magnetometers.
`SensorManager`	You can use this class to create an instance of the sensor service. This class provides various methods for accessing and listing sensors, registering and unregistering sensor event listeners, and acquiring orientation information. This class also provides several sensor constants that are used to report sensor accuracy, set data acquisition rates, and calibrate sensors.
`Sensor`	You can use this class to create an instance of a specific sensor. This class provides various methods that let you determine a sensor's capabilities.
`SensorEvent`	The system uses this class to create a sensor event object, which provides information about a sensor event. A sensor event object includes the following information: the raw sensor data, the type of sensor that generated the event, the accuracy of the data, and the timestamp for the event.
`SensorEventListener`	You can use this interface to create two callback methods that receive notifications (sensor events) when sensor values change or when sensor accuracy changes.
getDefaultSensor()	If a device has more than one sensor of a given type, one of the sensors must be designated as the default sensor.
i-android-sensors page 9	It's also important to note that this example uses the onResume() and onPause() callback methods to register and unregister the sensor event listener. As a best practice you should always disable sensors you don't need, especially when your activity is paused. Failing to do so can drain the battery in just a few hours because some sensors have substantial power requirements and can use up battery power quickly. The system will not disable sensors automatically when the screen turns off.

Sensor Coordinate System	• The most important point to understand about this coordinate system is that the axes are not swapped when the device's screen orientation changes. • Another point to understand is that your application must not assume that a device's natural (default) orientation is portrait. The natural orientation for many tablet devices is landscape.
Bound Service	A bound service runs only as long as another application component is bound to it. Multiple components can bind to the service at once, but when all of them unbind, the service is destroyed.
Network security best practices	• Minimize the amount of sensitive or personal user data that you transmit over the network. • Send all network traffic from your app over SSL. • Consider creating a network security configuration, which allows your app to trust custom CAs or restrict the set of system CAs that it trusts for secure communication
DBAL	A database abstraction layer is a simplified representation of a database in the form of a written description or a diagram.
SqliteDB.query params	`query(boolean distinct, String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy, String limit)`
db.update	`update(String table, ContentValues values, String whereClause, String[] whereArgs)`
db.insert	`insert(String table, String nullColumnHack, ContentValues values)`
db.insert.nullColumnHack	By default sqlite won't let you add empty values. So this is the name of the columns which are nullable.
db.delete	`delete(String table, String whereClause, String[] whereArgs)`
`execSQL(String sql)`	Execute a single SQL statement that is NOT a SELECT or any other SQL statement that returns data.
`rawQuery(String sql, String[] selectionArgs)`	Runs the provided SQL and returns a `Cursor` over the result set.
PendingIntent	A PendingIntent is a token that you give to a foreign application (e.g. NotificationManager, AlarmManager, Home Screen AppWidgetManager, or other 3rd party applications), which allows the foreign application to use your application's permissions to execute a predefined piece of code.
AlarmManager.`RTC_WAKEUP`	Alarm time in `System#currentTimeMillis` (wall clock time in UTC), which will wake up the device when it goes off.
Alarm set	`set(int type, long triggerAtMillis, PendingIntent operation)`

`Uri withAppendedId (Uri contentUri,long id)`	Appends the given ID to the end of the path. a new URI with the given ID appended to the end of the path This value will never be `null`.
`notifyChange(Uri uri, ContentObserver observer)`	Notify registered observers that a row was updated and attempt to sync changes to the network.
Firefox OS Apps	Hosted, Packaged, Privileged
objectStore.getAll()	There is a performance cost associated with looking at the value property of a cursor, because the object is created lazily. When you use getAll() for example, the browser must create all the objects at once. If you're just interested in looking at each of the keys, for instance, it is much more efficient to use a cursor than to use getAll(). If you're trying to get an array of all the objects in an object store, though, use getAll().
objectStore.openCursor(a,"prev")	Sometimes you may want to iterate in descending order rather than in ascending order (the default direction for all cursors). Switching direction is accomplished by passing prev to the openCursor() function as the second argument