Android Debug Bridge
Cheat Sheet

Selecting a device

`adb devices`	List of devices by serial number.
`adb devices -l`	List of devices by product/model.
`adb -s <serial> ...`	Command line selection.
`export ANDROID_SERIAL=<serial>`	Env. variable selection.

If a command starts with $ it has to be run from the Android shell or via `adb shell <command>`, or even better `adb shell <command> | less`.

Package installation

`adb install <apk>`	Installs app.
`$ pm install <path>`	Install app from phone path.
`$ pm install -r <path>`	Reinstall app from phone path.
`$ pm uninstall <name>`	Remove the app.
`$ pm get-install-location`	Install location: 0 - Auto 1 - Internal 2 - External

Package info

`$ pm list packages`	List package names.
`$ pm list packages -f`	As above + path to apks.
`$ pm list packages -3`	Only third party packages.
`$ pm list packages -s`	Only system packages.
`$ pm list packages -u`	Also uninstalled packages.
`$ dumpsys package packages`	List info on all apps.
`$ pm dump <name>`	List info on one package.
`$ pm path <package>`	Path to the apk file.

Permissions

`$ pm permission groups`	Permission groups definitions.
`$ pm list permissions -g -f`	List permissions details.

File operations

`adb push <local> <remote>`	Copy file/dir to device.
`adb pull <remote> [<local>]`	Copy file/dir from device.
`adb backup -f <file> [<packages...>]`	Backup the phone.

If you want to access the private package files just use `run-as <package> cat <file>`.

Paths

`/data/data/<package>`	App data, as described below.
`databases/`	App databases.
`shared_prefs/`	Shared preferences.
`/data/app`	APK files installed by user.
`/system/app`	Pre-installed APK files.
`/mnt/asec`	Encrypted apps (App2SD).
`/mnt/emmc`	Internal SD Card.
`/mnt/sdcard`	External/Internal SD Card.
`/mnt/sdcard/external_sd`	External SD Card.

Phone info

`$ sqlite3 /data/data/`
`com.android.providers.settings/`
`databases/settings.db .dump`
	Dump phone settings.

`$ getprop`	Get properties (e.g. model).
`$ dumpsys iphonesubinfo`	Get the IMEI.
`adb get-serialno`	Get the serial number.
`$ dumpsys battery`	Battery status.
`$ pm list users`	Lists phone users (4.1+).
`$ pm list features`	List phone features.

Services & activities

`$ service list`	List all services.
`$ dumpsys activity <package>/<activity>`	Activity info.

Activity Manager usage:

```
$ am start|startservice|broadcast <INTENT>
                                  [<COMPONENT>]
```

where `<INTENT>` is specified with following options:

`-a <ACTION>`	e.g. `android.intent.action.VIEW`
`-c <CATEGORY>`	e.g. `android.intent.category.LAUNCHER`

Common actions

To open the URL:

```
$ am start -a android.intent.action.VIEW -d URL
```

To launch camera:

```
$ am start -a android.media.action.IMAGE_CAPTURE
```

To launch Device Adminstrator Settings:

```
$ am start -S "'com.android.settings.Settings$
DeviceAdminSettingsActivity'"
```

Logs

All logs are accessed by using either

```
$ logcat [options] [filter] [filter] ...
```

or

```
adb logcat [options] [filter] [filter] ...
```

Useful options are:

`-d`	Only dump logs (do not block).
`-c`	Flush the buffers.
`-b <buffer>`	Buffer to display (default: system, main).
`<tag>[:priority]`	`filter` spec at the end of command.

Available priorities are:

`V`	Verbose
`D`	Debug
`I`	Info
`W`	Warn
`E`	Error
`F`	Fatal
`S`	Silent (suppress all output)

Other useful log information:

`$ dumpstate`	Dump current phone state.
`$ dumpsys`	Dump all system data.

Miscellaneous

`$ screencap -p <path>.png`	Screenshot (saved on device).
`$ screenrecord <path>.mp4`	Screen capture (path on device).

ADB daemon

`adbd` runs on TCP/5037.

`adb kill-server`	Kill the server if it is running.
`adb start-server`	Ensure that there is a server running.
`adb root`	Restarts the `adbd` with root permissions.

v 0.2 by @maldr0id
based on LaTeX cheat sheet by Winston Chang
http://www.stdout.org/∼winston/latex/