# QiSDK Cheat Sheet

## Creating a new project

- Use API level 4
- minSdkVersion : 23 - Marshmallow
- targetSdkVersion : 27 - Oreo

**Your main Activity**

In Java:

```
public class MainActivity extends RobotActivity implements RobotLifecycleCallbacks
```

In your onCreate() callback, register to the QiSDK:

```
QiSDK.register(this, this);
```

… and in onDestroy():

```
QiSDK.unregister(this, this);
```

**Optimizing the build**

For quicker loading, in the “android” section, use splits to build smaller APK files:

| *APK for robot only (smallest)* | *APK for robot or emulator* |
|---|---|
| `splits {`<br>`  abi {`<br>`  enable true`<br>`  reset()`<br>`  include "armeabi-v7a"`<br>`  universalApk false`<br>`  }`<br>`}` | `splits {`<br>`  abi {`<br>`  enable true`<br>`  reset()`<br>`  include "x86", "armeabi-v7a"`<br>`  universalApk false`<br>`  }`<br>`}` |

# Chaining Futures

If you want to chain async calls a and b, use:

| | |
|---|---|
| `a.thenConsume(b)` | if b needs the return value of a, but b doesn't return anything |
| `a.thenApply(b)` | If b needs the return value of a, and returns a value |
| `a.thenCompose(b)` | If b doesn't need the return value of and returns a Future |

Between the .then and the .andThen variants (the same exist apply and compose):

| | |
|---|---|
| `a.thenConsume(b)` | b is always called, and receives a future (that can say whether a failed) |
| `a.andThenConsume(b)` | b is only called if a succeeded, and receives b's return value |

For example, say, then animate (using Java 8's lambdas):

```
Future<Void> returnedOperation = say.async().run().andThenCompose(ignore -> {
   Animate animate = ...;
   Future<Void> animateFuture = animate.async().run();
   return animateFuture;
});
```

# QiChat Syntax

```
concept:(greetings) ^rand[hi hello "hey there"]
concept:(wine) [red white] wine
concept:(alcohol) [beer ~wine]
u:(~greetings) ~greetings
u:(do you have _~alcohol) yes, I have $1
u:(I want to drink something) do you want ~alcohol?
u:(hello {buddy} how are you) hello I am fine
u:(my name is *) nice to meet you
```