Basic of regular expressions:

A regular expression is a sequence of characters that forms a pattern.

Recommended docs:

References code:

typedef void (\* Function\_pointer)(const int data);

static const C\_table[]{

&function,

&function,

NULL

}

static void s\_init\_hw(const int t){

for(){

if(table[i] != NULL){

C\_Table[i]

}

}

}

Android process:

Is basically a linux process.

When launching an android app, android os creates a new process for it.

Process is the execution environment for application’s code.

The memory of a process can’t be accessed from other processes (“sandboxing”).

Process lifecycle:

Created when the application is launched.

Destroyed when the application is terminated (not just sent to background).

Can be destroyed by android os in special circumstances.

Android class:

Android.app.Application

Instantiated when android application is launched (when a process is created).

Only one instance of application per process.

Destroyed only when the process is destroyed.

File: AndroidManifest.xml

Code :

Val applicaton = this.application

Val application = this.applicationContext as Application

Object in class abstracts from android.app.Application will live entire lifetime of android application.

Global object:

Objects which are referred from the application object will live for as long as the app is alive. We’ll call these objects global objects.

Global objects are somewhat like singletons, but don’t involve static calls.

Code:

onCreate:

Timber.plant(Timer.DebugTree())

ProcessLifecycleOwner.get().lifecycle.addObserver(MyAppLifecycleObserver()){

}

Class MyAppLifecycleObserver : DefaultLifecycleObserver {

Override fun onStart(owner : LifecycleOwner){

Super.onStart(owner)

Timber.i(“”)

Override fun onStop(owner: LifecycleOwner){

Super.onStop(owner)

Timer.i(“”)

}

Application onCreate:

Will be called once when android application is launched.

The place to perform application-wide, global initializations.

Conclusion of lifecycle:

Specification of object’s creation and destruction times, as well as a set of special callback methods which the system invokes automatically in response to specific events.

When launching android application, android os creates a new process for it.

After android application terminates, the system kills it process and releases all the associated resources.

Sending applications to background does not necessarily terminate them.

A single instance of android.app.Application class is created when the app is launched.

After instantiation, system invokes application’s onCreate callback.

Application object will remain alive for entire lifetime of the application.

User can provide subclass of Application.

Objects referenced from application object (include transitively) will live for as long as the app is alive and won’t be garbage collected.

We call these objects: global objects.

Global objects can be used to share state between different components inside the app.