

Benvenuti a questa seconda lezione del corso di Android per Accademia di Android. In questa lezione vedremo che cos'è una attività, come essa viene implementata, ovvero parleremo del mattone fondamentale di ogni applicazione Android. Questo mattone consente di compiere delle precise operazioni con cui l'utente può interagire con l'app stessa; è importante fin da subito comprendere che per ogni applicazione in esecuzione vi è una sola attività in esecuzione in un preciso istante (vedi **mutua esclusione**). In questa lezione vi spiegherò inoltre il **ciclo di vita** di una attività, vedremo quindi che quando noi lanciamo un'applicazione il sistema operativo Android va a chiamare delle funzioni native che appartengono a una classe padre ereditata (**AppCompatActivity**) che l'SDK di Android ci mette a disposizione quando installiamo Android studio.

Un tablet è diverso dal nostro computer di casa infatti abbiamo un display più piccolo, abbiamo poca memoria RAM e soprattutto abbiamo un'autonomia della batteria ridotta, quindi Android è strutturato in maniera tale che in ogni singolo momento in esecuzione ci sia una e una sola attività riferita a una app. Il passaggio da una attività all'altra può accadere nei seguenti due casi:

1) Intenzionalmente quando decidiamo di aprire un'altra applicazione perché dobbiamo fare qualcos'altro e quindi android interrompe la nostra applicazione ibernandola in RAM salvando tutto quello che abbiamo fatto fin a quel momento e successivamente passa il controllo alla nuova attività della seconda applicazione. Una volta finito di utilizzare la seconda applicazione possiamo riaprire la prima applicazione e Android scongelerà l'attività precedentemente ibernata ripristinandola dal preciso momento in cui la si era sospesa.

2) Interruzione esterna:

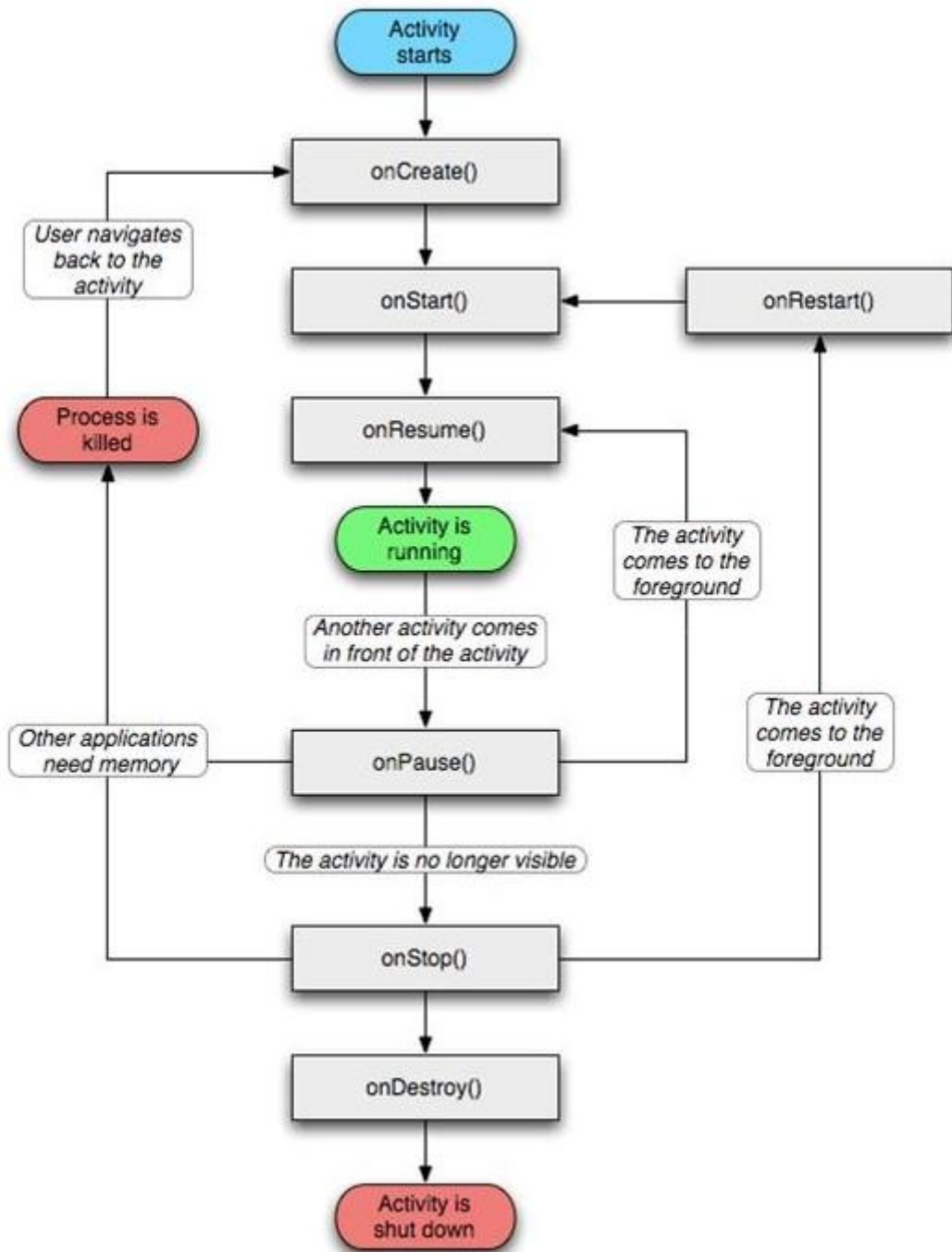
Avviene quando noi riceviamo per esempio una telefonata: Android iberna la nostra app in RAM e passa il controllo alla attività del gestore software delle chiamate. Una volta terminata la telefonata, Android ripristinerà l'app che stavate usando poco tempo prima.

E' utile sottolineare che una attività puo' terminare quando:

- 1) Il sistema decide che non è più utile e perciò la distrugge.
- 2) Il sistema ha quasi esaurito la memoria RAM e deve recuperare spazio iniziando ad uccidere brutalmente le attività in background.

Il **ciclo di vita di una activity** parte quando noi lanciamo l'applicazione dal nostro tablet/smartphone ed è il sistema operativo Android che va a eseguire l'attività settata come primaria e di conseguenza eseguendo delle funzioni una di seguito all'altra in base a determinate situazioni come vedremo più' avanti.

Faccio notare che una applicazione può creare internamente (come vedremo poi nell'esempio di codice che vedremo fra poco) più' attività, ma anche qui la regola è la stessa: in ogni istante solo 1 attività può andare in esecuzione.



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
}

@Override
protected void onRestart() {
    super.onRestart();
}

@Override
protected void onStart() {
    super.onStart();
}

@Override
protected void onResume() {
    super.onResume();
}

@Override
protected void onPause() {
    super.onPause();
}

@Override
protected void onStop() {
    super.onStop();
}

@Override
protected void onDestroy() {
    super.onDestroy();
}

```

Le funzioni principali di una activity sono:

Questa funzione è la prima che viene invocata quando si lancia l'app con l'obiettivo di creare l'attività. In poche parole si tratta della porta di accesso del nostro programma. Il parametro savedInstanceState permette di passare eventualmente una informazione in merito a uno stato di una attività precedentemente terminata. La prima volta savedInstanceState = null.

Questa funzione è invocata quando si riprende l'esecuzione di una attività precedentemente sospesa.

Questa funzione è invocata quando l'app diventa visibile sullo schermo.

Questa funzione è invocata poco prima che si inizi a interagire graficamente (la schermata) con l'utente.

Quando si mette in sospensione l'app e l'utente non interagisce piu' con l'app stessa.

Quando l'app non è piu' visibile sullo schermo dopo onPause.

Quando è stato dato l'ordine di terminare l'applicazione.

Ora implementerò una app che notificherà tramite logcat quando e dove vengono chiamati queste funzioni mostrate qui sopra.