# Day：触摸事件分发机制分析

## Activity的分发

1、事件先分发给PhoneWindow，PhoneWindow不消费则传给Activity的onTouchEvent方法。

2、PhoneWindow传给DecorView，DecorView传给根布局ViewGroup。 ## ViewGroup的分发 1、首先调用onInterCeptTouchEvent方法，判断ViewGroup自己是否需要，需要则拦截，拦截后就不会传递给childView。

2、拦截后，调用ViewGroup自己的onTouchEvent方法。

3、不拦截则传递给childView，按视图层次结构依次传递下去。

4、最终的那个childView也不需要时，事件会进行回传，依次调用前面每一层的onTouchEvent方法。 ## View的分发 1、View会依次调用onTouchListener、onTouchEvent、onLongClickListener、onClickListener。

2、在onTouchListener、onTouchEvent中可决定是否消费事件，不消费，事件则开始回传。

3、注册了onLongClickListener、onClickListener及设置了Clickable等，View就会消费事件。 ## 伪代码 1、ViewGroup分发的伪代码

// 返回值代表是否将事件消费掉  
public boolean dispatchTouchEvent(MotionEvent event) {  
 // 默认状态为未消费过  
 boolean result = false;  
  
 // 如果没有拦截（ViewGroup一般不会进行拦截，可点击的ViewGroup除外）  
 if (!onInterceptTouchEvent(event)) {  
 // 则交给childView  
 result = child.dispatchTouchEvent(event);  
 }  
  
 // 如果事件没有被childView消费  
 if (!result) {  
 // 则调用自身 onTouchEvent()  
 result = onTouchEvent(event);  
 }  
  
 // 返回事件消费状态  
 return result;  
}

2、View分发的伪代码

// 返回值代表是否将事件消费掉  
public boolean dispatchTouchEvent(MotionEvent event) {  
 if(mOnTouchListener.onTouch(this, event)) {  
 return true;  
 } else if (onTouchEvent(event)) {  
 return true;  
 }  
 return false;  
}

## 注意事项

* 拦截事件指事件不再往下传递，使用事件指从事件中获取想要的信息，消费事件指把事件吃了
* 只要给 View 注册了 onClickListener、onLongClickListener、OnContextClickListener 其中的任何一个监听器或者设置了 android:clickable=”true” 就代表这个 View 是可点击的，可点击的 View 就会消费事件
* ViewGroup 和 ChildView 同时注册了点击事件监听器时，事件优先给 ChildView 消费
* 同一次点击事件只能被一个 View 消费，防止事件混乱