|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| <hitle> | Material Design | <chare> | 1 | <pext> | <starting>Material Design</heading> Android Material Design is a design language created by Google  for use in its Android operating system. It is based on the principles of material design, which are meant to provide  a more intuitive and natural user experience.  Android Material Design was first introduced in 2014  with the release of Android 5.0 Lollipop. </ending> | </end> |
| <hitle> | three-dimensional environment of material design | <chare> | 1 | <pext> | <starting>three-dimensional environment of material design</heading> The three-dimensional environment of material design is created  by using light and shadow to give the illusion of depth. This means that elements on the screen appear to be floating on top  of other elements, and that they cast shadows when appropriate. This effect is achieved through the use of elevation values,  which determine how high an element appears to be in relation  to other elements. </ending> | </end> |
| <hitle> | shadows in material design | <chare> | 1 | <pext> | <starting>shadows in material design</heading> The use of shadows in material design can help  to create a sense of depth and dimensionality, making it easier for users  to understand the layout of a page or screen. Additionally, shadows can help to highlight important elements on a page,  making them more visible and easier to interact with. </ending> | </end> |
| <hitle> | custom animations using material design | <chare> | 1 | <pext> | <starting>custom animations using material design</heading> Yes, it is possible to create custom animations using material design principles. One way to do this is by using the Animation API, which allows you  to create animations that can be applied to Views. Another way to create custom animations is by using the Transition API,  which allows you to create animations that transition between  different states of a View. </ending> | </end> |
| <hitle> | floating action button (FAB) | <chare> | 1 | <pext> | <starting>floating action button (FAB)</heading> A floating action button (FAB) is a circular button  that triggers the primary action in an Android app. It is typically placed in the bottom right corner of a screen. The FAB can be used in a number of ways, but one common use  is as a “call to action” button on a screen. For example, if you have a list of items and you want the user  to be able to add a new item, you could use a FAB for that. The recommended elements or patterns to use with the FAB are  the Snackbar and the Floating Action Button. </ending> | </end> |
| <hitle> | bottom sheets | <chare> | 1 | <pext> | <starting>bottom sheets</heading> Bottom sheets are a type of modal window that can be used to display additional information or controls. They are typically used when there is a need to provide  more information or controls that can be displayed on a single screen,  but without taking up the entire screen like a traditional modal window would. </ending> | </end> |
| <hitle> | snackbar notification | <chare> | 1 | <pext> | <starting>snackbar notification</heading> A snackbar notification is a message  that appears at the bottom of the screen in Android. It is used to provide information to the user,  and can be dismissed by swiping it off the screen. Snackbar notifications are best used for short, timely messages  that do not require immediate user interaction. For example, a snackbar notification might be used to confirm  that a user’s action has been successfully completed,  or to let the user know that a new message has been received. </ending> | </end> |
| <hitle> | navigation drawer | <chare> | 1 | <pext> | <starting>navigation drawer</heading> Some things to keep in mind when designing a navigation drawer  are to make sure that it is easily accessible from anywhere in the app,  that it is easy to use and understand, and that it is visually appealing. </ending> | </end> |
| <hitle> | linear layout vs relative layout | <chare> | 1 | <pext> | <starting>linear layout vs relative layout</heading> Linear layout arranges elements in a single line, either horizontally or vertically. Relative layout arranges elements in relation to each other,  or to the parent layout. Relative layout is more flexible and should be used more often  than linear layout. </ending> | </end> |
| <hitle> | coordinator layout | <chare> | 1 | <pext> | <starting>coordinator layout</heading> A coordinator layout is a layout that allows child views  to coordinate their actions with each other. This can be useful for things like creating floating action buttons  that only appear when the user scrolls to a certain point in the document. </ending> | </end> |
| <hitle> | palette generator tool | <chare> | 1 | <pext> | <starting>palette generator tool</heading> The main components of the palette generator tool are the color picker,  the color swatches, and the color palette. The color picker allows you to select a color from an image or a web page. The color swatches allow you to save and share your favorite colors. The color palette allows you to create and save color schemes. </ending> | </end> |
| <hitle> | Elevation | <chare> | 1 | <pext> | <starting>Elevation</heading> Elevation is one of the key concepts in material design, and refers to the distance between two surfaces. In general, the higher the elevation, the closer the surface is to the user. This concept is used to create a sense of depth and hierarchy in material design applications, and can be used to create shadows and other visual effects. </ending> | </end> |