<https://www.geeksforgeeks.org/user-interface-ui/>

<https://www.geeksforgeeks.org/gui-full-form/>

<https://www.geeksforgeeks.org/difference-between-cli-and-gui/>

<https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces>

<https://www.geeksforgeeks.org/android-ui-layouts/>

<https://www.geeksforgeeks.org/layouts-android-ui-design/>

<https://www.geeksforgeeks.org/android-ui-layouts/>

Mobile User Interface Design

**User Interface (UI)**

User Interface (UI) defines the way humans interact with the information systems. In Layman’s term, User Interface (UI) is a series of pages, screens, buttons, forms and other visual elements that are used to interact with the device. Every app and every website has a user interface.

User Interface (UI) Design is the creation of graphics, illustrations, and use of photographic artwork and typography to enhance the display and layout of a digital product within its various device views. Interface elements consist of input controls (buttons, drop-down menus, data fields), navigational components (search fields, slider, icons, tags), informational components (progress bars, notifications, message boxes).

**Types of User Interface (UI):**

* [GUI (Graphical User Interface)](https://en.wikipedia.org/wiki/Graphical_user_interface)
* [Command line interface](https://en.wikipedia.org/wiki/Command-line_interface)
* [Menu Driven Interface](https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces)
* [Form Based Interface](https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces)
* [Natural Language Interface](https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces)
* **Significance of User Interface (UI):**  
  A good User Interface (UI) focuses on making user’s interactions simple and efficient. User would appreciate a website with intuitive user interface that leads them towards their task in most engaging way. User Interface (UI) design focuses on thinking of a user, what they might need to do when they visit website and ensure that the interface has elements that are easy to access and understand. Being a UI designer, one need to understand the goals, skills, preferences and tendencies of the user to make a better interface.

**Why is User Interface (UI) important?**

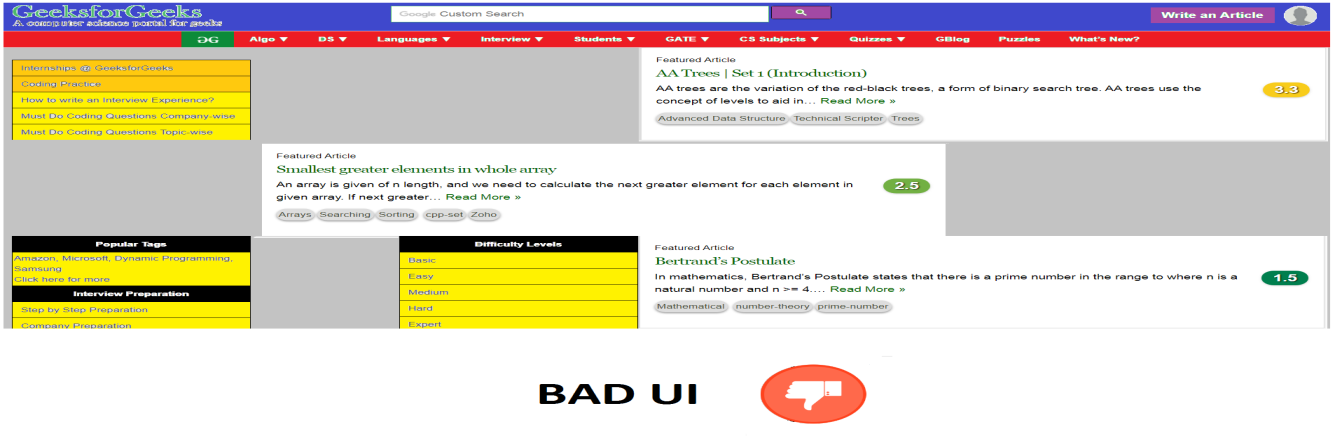
* **How you present your product matters the most.**  
  The presentation (**the interface**) of a badly designed application or website can drive away the incoming users and leave a bad impression on them. Navigation through a site can be made efficient and simple by effective UI design.
* **Great design is great business**  
  The color scheme, layout, graphics, tab and button placement, typography usage and other design elements determine how well the site/app communicates visually with the user.
* **At a given time, finding right thing at right place is essential**  
  Consistent navigation that is equally easy to locate and browse through is basic need of any app/website. If navigation is complex for a user to work, there is no way he would want to work on that site/app.

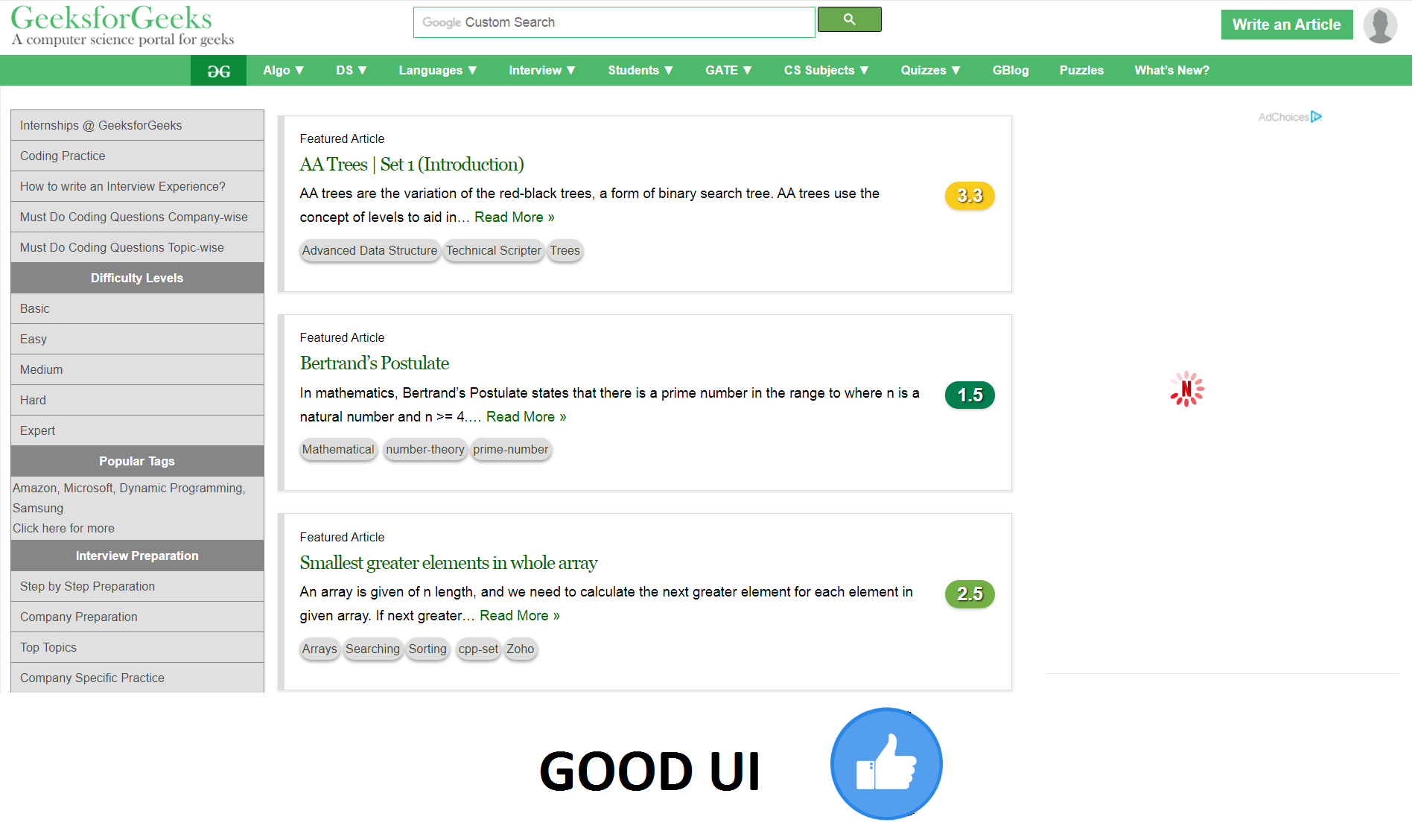
**Advantages of User Interface (UI):**

* No need to learn complex commands/languages for working with UI.
* Easiness for non-technical people. A beginner can navigate through a site with ease if its simple and well informative.
* Usage of blocks and typography makes user experience better.
* Easy setup and ready to start working are awesome. Hiding the complexity of actions from the user and display only the required information is key to good interface.

**Disadvantages of UI :**

* When not properly built, it can be very difficult to work with.
* Takes time to built a Perfect UI.
* **Example:** It shows a Bad UI design and a Good UI design. See the differences and try to understand how important it is to have a good UI design.



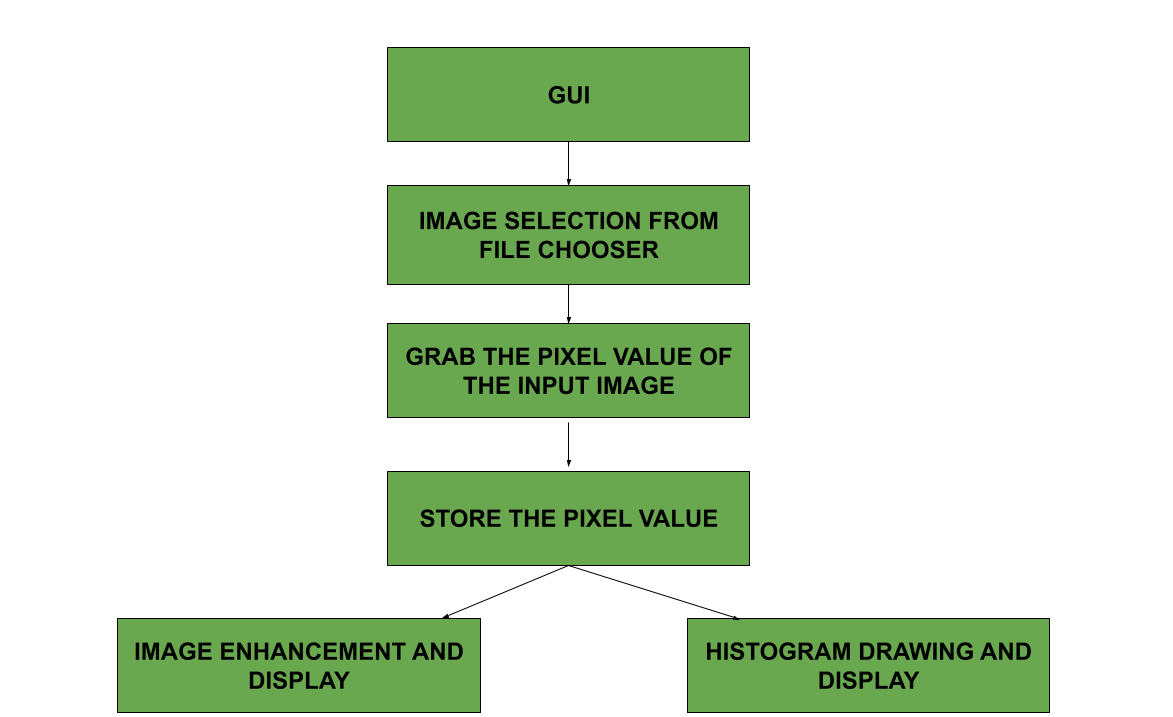


As the User Interface can make or break the incoming users, it’s important to take care of below points when designing a UI:

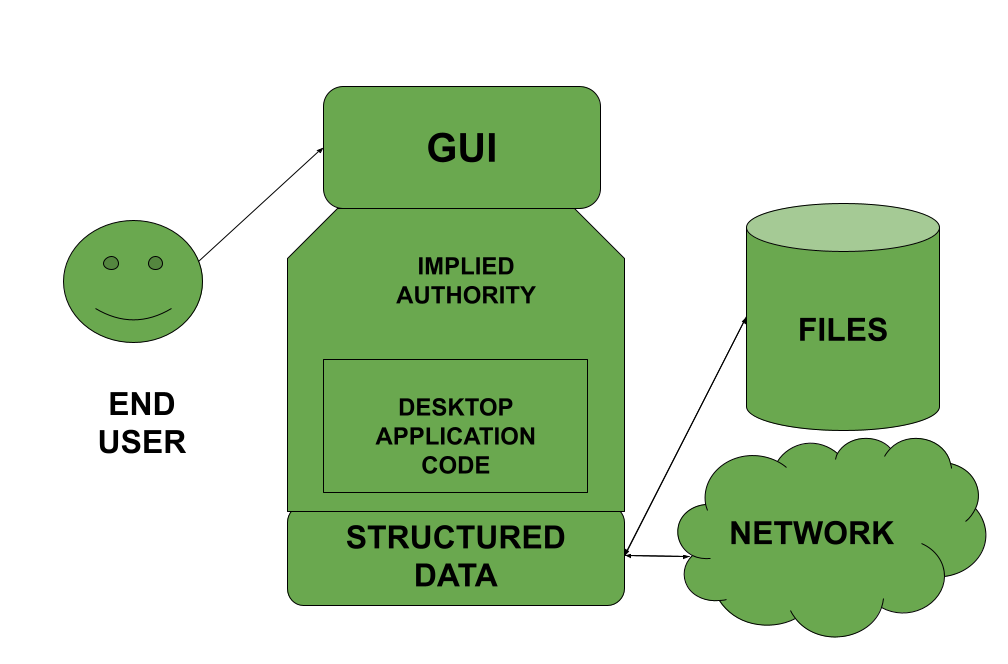
* **Keep the interface simple:** Clear and simple interface are best. Avoid unnecessary elements. Best interfaces are invisible to user.
* **Be consistent and use common UI elements:** Using common elements, users feel more comfortable and are able to get things done more quickly. Create pattern to facilitate efficiency.
* **Placement of items:** To draw attention to most important pieces of information careful placement of items is necessary. This can improve users readability and engage them.
* **Use of right color:** To direct attention towards something take advantage of color, light, shade, contrast and texture. It’s important top make use of good color combination as a bad color combination can easily distract or irritate a user.
* **Anticipate:** Make the user to work less by having pre-chosen fields, reduce the burden on the user, anticipate the goals of the users who come to your site. The things which can be mostly searched by the users are presented so that the users need not to work to search for it.

# Graphical user interface

# GUI is a user-friendly interface used to communicate with the help of electronic devices. It displays all the contents whether a text file or an object or pictures or videos and all the things that a user wants to visualize. It interacts well and can be used everywhere whether a mobile phone, tablet, laptops, Personal Computer and all the other electronic devices. It can be used best in the gaming side where the resolution is being considered. The basic GUI format is represented in the form of the diagram:



The basic structure of GUI using implied authority is given by the following diagram



**Characteristics**

* As mentioned above they are user-friendly i.e very easy to use.
* A GUI consists of different characteristics such as Menu, Tabs, Pointers and many more kinds of stuff
* The icons represent on the user interface represents the software or the file or some application required on the screen.

**Advantages**

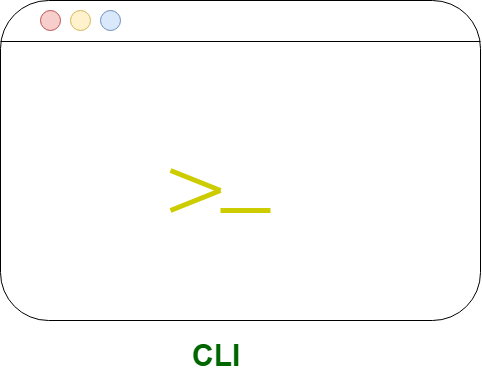
* User Interface is pretty simple and convenient for the beginners to understand.
* It is very intuitive and user friendly and can be used by anyone.
* End-user need not memorize commands to perform actions in the application.

**Disadvantages**

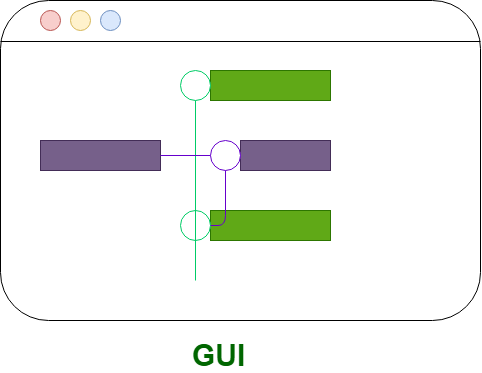
* A bad GUI always create a problem for the gamer’s as it creates a bad impact for them
* A GUI which is not user-friendly can mislead the user and the efficiency of completing the work reduces.

# Difference between CLI and GUI

# **CLI** is the word form used for **Command Line Interface.** CLI permits users to put in writing commands associate degree exceedingly in terminal or console window to interact with an operating system. CLI is a platform or medium wherever users answer a visible prompt by writing a command and get the response from the system, for this users have to be compelled to kind command or train of command for performing the task. CLI is suitable for pricey computing wherever input exactitude is the priority.



**GUI** stands for **Graphical User Interface.** GUI permits users to use the graphics to interact with an operating system. In the graphical user interface, menus are provided such as windows, scrollbars, buttons, wizards, painting pictures, alternative icons, etc. It’s intuitive, simple to find out, and reduces psychological feature load. In GUI, the information is shown or presented to the user in any form such as: plain text, videos, images, etc.



Let’s see the difference between GUI and CLI:

|  |  |  |
| --- | --- | --- |
| S.no | CLI | GUI |
| 1 | CLI is difficult to use. | Whereas it is easy to use. |
| 2 | It consumes low memory. | While consuming more memory. |
| 3 | In CLI we can obtain high precision. | While in it, low precision is obtained. |
| 4 | CLI is faster than GUI. | The speed of GUI is slower than CLI.  GUI |
| 5 | CLI operating system needs only a keyboard. | While GUI operating system needs both a mouse and keyboard. |
| 6 | CLI’s appearance cannot be modified or changed. | While its appearance can be modified or changed. |
| 7 | In CLI, input is entered only at a command prompt. | While in GUI, the input can be entered anywhere on the screen. |
| 8 | In CLI, the information is shown or presented to the user in plain text and files. | While in GUI, the information is shown or presented to the user in any form such as: plain text, videos, images, etc. |
| 9 | In CLI, there are no menus provided. | While in GUI, menus are provided. |
| 10 | There are no graphics in CLI. | While in GUI, graphics are used. |
| 11 | CLI do not use any pointing devices. | While it uses pointing devices for selecting and choosing items. |
| 12 | In CLI, spelling mistakes and typing errors are not avoided. | Whereas in GUI, spelling mistakes and typing errors are avoided. |
| 13 | Some command-line environments provide multitasking but it is complicated to see several things on one screen. | |  |  |  | | --- | --- | --- | |  |  | GUI enables a user to easily observe and operate various things at once. | |
| 14 | CLI enables a user to simply script a series of instructions to carry out a task or execute a program. | GUI does not provide the facility to script a sequence of commands. |

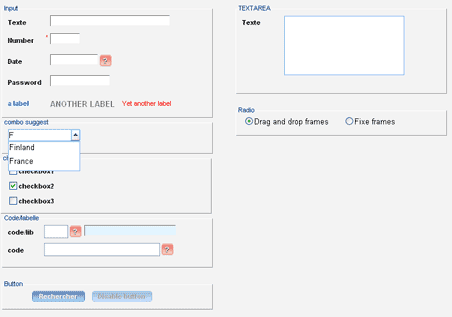
[Menu Driven Interface](https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces)

A menu-driven interface is commonly used on cash machines (also known as automated teller machines, or ATMs), ticket machines and information kiosks (for example in a museum). Menu-driven interfaces provide a simple and an easy to use interface composed of a series of menus and sub-menus which the user accesses by pressing buttons, often on a touch-screen device. Preferably, if one has knowledge on UML modeling, it can be a good example when designing the architecture of the machine.

[](https://commons.wikimedia.org/wiki/File:Bankomat_050421.jpg)

[Form Based Interface](https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces)

* This is a method of enabling you to interact with an application.
* The form normally provides limited choices as to the use.
* For example, a form interface for setting text characteristics in application software might offer the choices of selecting font size, colour, style.
* A form interface which will allow you to interact with the system software might offer choices such as selecting your screen resolution, default language, keyboard style etc.
* A form interface can also be used to enter data into a system, for example a database system will usually allow you to create a form to enter data into tables.



[Natural Language Interface](https://en.wikibooks.org/wiki/A-level_Computing/CIE/Computer_systems,_communications_and_software/System_software/User_interfaces)

A natural language interface is a spoken interface where the user interacts with the computer by talking to it. Sometimes referred to as a 'conversational interface', this interface simulates having a conversation with a computer. Made famous by science fiction (such as in [Star Trek](https://en.wikipedia.org/wiki/Star_Trek)), natural language systems are not yet advanced enough to be in wide-spread use. Commonly used by telephone systems as an alternative to the user pressing numbered buttons the user can speak their responses instead. An Example of this type of interface is Voice Recognition.

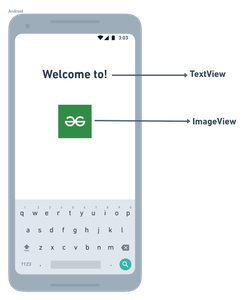
This is the kind of interface used by the popular [iPhone](https://en.wikipedia.org/wiki/iPhone) application called [Siri](https://en.wikipedia.org/wiki/Siri" \o "w:Siri) and [Cortana](https://en.wikipedia.org/wiki/Cortana" \o "w:Cortana) in [Windows](https://en.wikipedia.org/wiki/Windows).

# Android UI Layouts

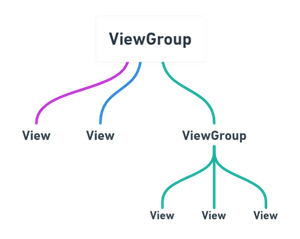
Android **Layout** is used to define the user interface that holds the UI controls or widgets that will appear on the screen of an android application or activity screen. Generally, every application is a combination of View and ViewGroup. As we know, an android application contains a large number of activities and we can say each activity is one page of the application. So, each activity contains multiple user interface components and those components are the instances of the View and ViewGroup. All the elements in a layout are built using a hierarchy of **View**and **ViewGroup**objects.

### View

A **View** is defined as the user interface which is used to create interactive UI components such as [TextView](https://www.geeksforgeeks.org/textview-widget-in-android-using-java-with-examples/), [ImageView](https://www.geeksforgeeks.org/imageview-in-android-with-example/), [EditText](https://www.geeksforgeeks.org/edittext-widget-in-android-using-java-with-examples/), [RadioButton](https://www.geeksforgeeks.org/radiobutton-in-kotlin/), etc., and is responsible for event handling and drawing. They are Generally Called Widgets.



A **ViewGroup** act as a base class for layouts and layouts parameters that hold other Views or ViewGroups and to define the layout properties. They are Generally Called layouts.



The Android framework will allow us to use UI elements or widgets in two ways:

* Use UI elements in the XML file
* Create elements in the Kotlin file dynamically

### Types of Android Layout

* **Android Linear Layout:**LinearLayout is a ViewGroup subclass, used to provide child View elements one by one either in a particular direction either horizontally or vertically based on the orientation property.
* **Android Relative Layout:**RelativeLayout is a ViewGroup subclass, used to specify the position of child View elements relative to each other like (A to the right of B) or relative to the parent (fix to the top of the parent).
* **Android Constraint Layout:**ConstraintLayout is a ViewGroup subclass, used to specify the position of layout constraints for every child View relative to other views present. A ConstraintLayout is similar to a RelativeLayout, but having more power.
* **Android Frame Layout:**FrameLayout is a ViewGroup subclass, used to specify the position of View elements it contains on the top of each other to display only a single View inside the FrameLayout.
* **Android Table Layout:**TableLayout is a ViewGroup subclass, used to display the child View elements in rows and columns.
* **Android Web View:**WebView is a browser that is used to display the web pages in our activity layout.
* **Android ListView:**ListView is a ViewGroup, used to display scrollable lists of items in a single column.
* **Android Grid View:**GridView is a ViewGroup that is used to display a scrollable list of items in a grid view of rows and columns.

# Layouts in Android UI Design

Layout Managers (or simply layouts) are said to be extensions of the ViewGroup class. They are used to set the position of child Views within the UI we are building. We can nest the layouts, and therefore we can create arbitrarily complex UIs using a combination of layouts.

There is a number of layout classes in the Android SDK. They can be used, modified or can create your own to make the UI for your Views, Fragments and Activities. You can display your contents effectively by using the right combination of layouts.

# The most commonly used layout classes that are found in Android SDK are:

# **FrameLayout-** It is the simplest of the Layout Managers that pins each child view within its frame. By default the position is the top-left corner, though the gravity attribute can be used to alter its locations. You can add multiple children stacks each new child on top of the one before, with each new View potentially obscuring the previous ones.

# https://media.geeksforgeeks.org/wp-content/uploads/androidLayouts3-1.png

# **LinearLayout-** A LinearLayout aligns each of the child View in either a vertical or a horizontal line. A vertical layout has a column of Views, whereas in a horizontal layout there is a row of Views. It supports a weight attribute for each child View that can control the relative size of each child View within the available space.

# Lightbox

# **RelativeLayout-** It is flexible than other native layouts as it lets us to define the position of each child View relative to the other views and the dimensions of the screen.

# Lightbox

* **GridLayout-** It was introduced in Android 4.0 (API level 14), the Grid Layout used a rectangular grid of infinitely thin lines to lay out Views in a series of rows and columns. The Grid Layout is incredibly flexible and can be used to greatly simplify layouts and reduce or eliminate the complex nesting often required to construct UIs using the layouts described before.

Each of these layouts is designed to scale to suit the screen size of the host device by avoiding the used of absolute co- ordinates of the positions or predetermined pixel values. This makes the app suitable for the diverse set of Android devices.

