

What is Applet?

An applet is a Java program that can be embedded into a web page. It runs inside the web browser and works at client side. An applet is embedded in an HTML page using the APPLET or OBJECT tag and hosted on a web server.

1. All applets are sub-classes (either directly or indirectly) of [*java.applet.Applet*](#) class.
2. Applets are not stand-alone programs. Instead, they run within either a web browser or an applet viewer. JDK provides a standard applet viewer tool called applet viewer.
3. In general, execution of an applet does not begin at `main()` method.
4. Output of an applet window is not performed by `System.out.println()`. Rather it is handled with various AWT methods, such as `drawString()`.
5. **Life cycle of an applet :**

Following are the steps of Applet Life Cycle :-

Init() ----→ **start()** ----→ **paint()**
|
Stop()
Destroy().

It is important to understand the order in which the various methods shown in the above image are called. When an applet begins, the following methods are called, in this sequence:

1. `init()`
2. `start()`
3. `paint()`

When an applet is terminated, the following sequence of method calls takes place:

1. `stop()`
2. `destroy()`

Let's look more closely at these methods.

1. **`init()`** : The **`init()`** method is the first method to be called. This is where you should initialize variables. This method is called **only once** during the run time of your applet.
2. **`start()`** : The **`start()`** method is called after **`init()`**. It is also called to restart an applet after it has been stopped. Note that **`init()`** is called once i.e. when the first time an applet is loaded whereas **`start()`** is

called each time an applet's HTML document is displayed onscreen. So, if a user leaves a web page and comes back, the applet resumes execution at **start()**.

3. **paint()** : The **paint()** method is called each time an AWT-based applet's output must be redrawn. This situation can occur for several reasons. For example, the window in which the applet is running may be overwritten by another window and then uncovered. Or the applet window may be minimized and then restored.

paint() is also called when the applet begins execution. Whatever the cause, whenever the applet must redraw its output, **paint()** is called.

The **paint()** method has one parameter of type [Graphics](#). This parameter will contain the graphics context, which describes the graphics environment in which the applet is running. This context is used whenever output to the applet is required.

Note: This is the only method among all the method mention above, which is parametrised. It's prototype is **public void paint(Graphics g)**

where g is an object reference of class Graphic.

The prototype of paint() method,

Whenever we pass object reference in arguments then the object will be provided by its caller itself. In this case the caller of paint() method is browser, so it will provide an object. The same thing happens when we create a very basic program in normal Java programs.

For Example:

```
public static void main(String []args) {}
```

Here we have created an object reference without creating its object but it still runs because it's caller, i.e JVM will provide it with an object.

- **stop()** : The **stop()** method is called when a web browser leaves the HTML document containing the applet—when it goes to another page, for example. When **stop()** is called, the applet is probably running. You should use **stop()** to suspend threads that don't need to run when the applet is not visible. You can restart them when **start()** is called if the user returns to the page.
- **destroy()** : The **destroy()** method is called when the environment determines that your applet needs to be removed completely from memory. At this point, you should free up any resources the applet may be using.

The **stop()** method is always called before **destroy()**.