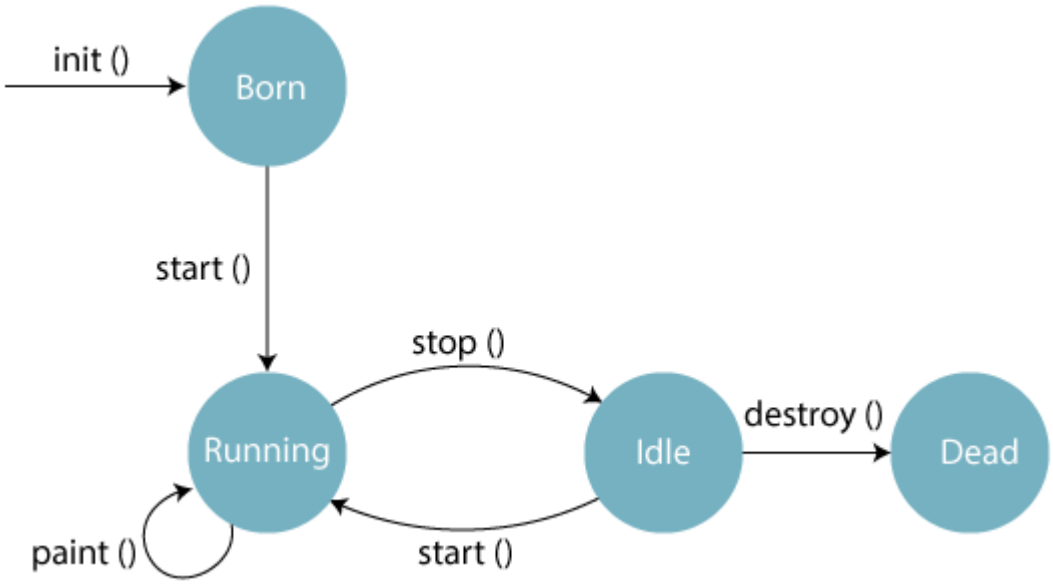
Applet Life Cycle in Java

In Java, an applet is a special type of program embedded in the web page to generate dynamic content. Applet is a class in Java.

The applet life cycle can be defined as the process of how the object is created, started, stopped, and destroyed during the entire execution of its application. It basically has five core methods namely init(), start(), stop(), paint() and destroy().These methods are invoked by the browser to execute.

Along with the browser, the applet also works on the client side, thus having less processing time.

Methods of Applet Life Cycle



There are five methods of an applet life cycle, and they are:

- **init():** The init() method is the first method to run that initializes the applet. It can be invoked only once at the time of initialization. The web browser creates the initialized objects, i.e., the web browser (after checking the security settings) runs the init() method within the applet.
- **start():** The start() method contains the actual code of the applet and starts the applet. It is invoked immediately after the init() method is invoked. Every time the browser is loaded or refreshed, the start() method is invoked. It is also invoked whenever the applet is maximized, restored, or moving from one tab

to another in the browser. It is in an inactive state until the init() method is invoked.

- **stop():** The stop() method stops the execution of the applet. The stop () method is invoked whenever the applet is stopped, minimized, or moving from one tab to another in the browser, the stop() method is invoked. When we go back to that page, the start() method is invoked again.
- **destroy():** The destroy() method destroys the applet after its work is done. It is invoked when the applet window is closed or when the tab containing the webpage is closed. It removes the applet object from memory and is executed only once. We cannot start the applet once it is destroyed.
- **paint():** The paint() method belongs to the Graphics class in Java. It is used to draw shapes like circle, square, trapezium, etc., in the applet. It is executed after the start() method and when the browser or applet windows are resized.

Sequence of method execution when an applet is executed:

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

Sequence of method execution when an applet is executed:

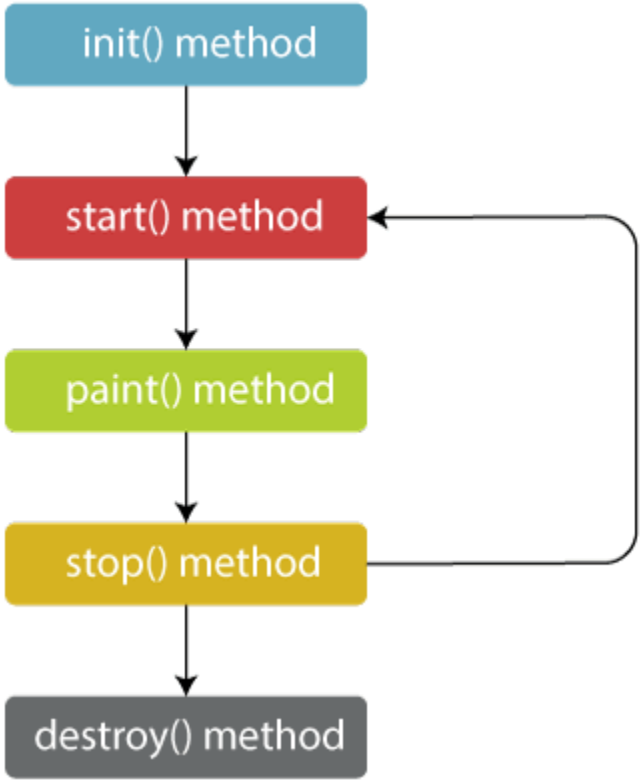
1. stop()
2. destroy()

Applet Life Cycle Working

- The Java plug-in software is responsible for managing the life cycle of an applet.
- An applet is a Java application executed in any web browser and works on the client-side. It doesn't have the main() method because it runs in the browser. It is thus created to be placed on an HTML page.
- The init(), start(), stop() and destroy() methods belongs to the **applet.Applet** class.
- The paint() method belongs to the **awt.Component** class.
- In Java, if we want to make a class an Applet class, we need to extend the **Applet**
- Whenever we create an applet, we are creating the instance of the existing Applet class. And thus, we can use all the methods of that class.

Flow of Applet Life Cycle:

These methods are invoked by the browser automatically. There is no need to call them explicitly.



Syntax of entire Applet Life Cycle in Java

```
1. class TestAppletLifeCycle extends Applet {
2. public void init() {
3. // initialized objects
4. }
5. public void start() {
6. // code to start the applet
7. }
8. public void paint(Graphics graphics) {
9. // draw the shapes
10. }
11. public void stop() {
12. // code to stop the applet
```

```
13. }
14. public void destroy() {
15. // code to destroy the applet
16. }
17. }
```

Difference Between a Java Application and a Java Applet

Parameters	Java Application	Java Applet
Meaning and Basics	A Java Application is a type of program that can get independently executed on a computer.	A Java Applet is a small program that makes use of another application program so that we can execute it.
Main() Method	The execution of the Java application begins with the main() method. The usage of the main() is a prerequisite here.	The Java applet initializes through the init(). It does not require the usage of any main() method.
Execution	It cannot run alone, but it requires JRE for its execution.	It cannot run independently but requires APIs for its execution (Ex. APIs like Web API).
Installation	One needs to install a Java application priorly and explicitly on a local computer.	A Java applet does not require any prior installation.
Communication among other Servers	It is possible to establish communication with the other servers.	It cannot really establish communication with the other servers.
Read and Write Operations	The Java applications are capable of performing the read and write operations on various files present in a local computer.	A Java applet cannot perform these applications on any local computer.
Restrictions	These can easily access the file or data present in a computer system or device.	These cannot access the file or data available on any system or local computers.
Security	Java applications are pretty trusted, and thus, come with no security concerns.	Java applets are not very trusted. Thus, they require security.

JAVA APPLET Program to Find Sum of Two Numbers

At the time creation of a data base, we must embed the applet into the HTML page then only it works

Step 1: Create the java program with " *filename.java* ".

Step 2: Create the html program with "*filename.html*".

Step 3: Compile the java program "*javac filename.java*".

Step 4: View applet using "*appletviewer filename.html*".

Note: Both program must have the same name and Both program must be in same folder.

Java Program

```
1. import java.awt.*;
2. import java.awt.event.*;
3. import java.applet.*;
4. public class Q2 extends Applet implements ActionListener
5. {
6.     TextField t1 = new TextField(10);
7.     TextField t2 = new TextField(10);
8.     TextField t3 = new TextField(10);
9.     Label l1 = new Label("FIRST NO=:");
10.      Label l2 = new Label("SECOND NO:");
11.      Label l3 = new Label("SUM:");
12.      Button b = new Button("ADD");
13.      public void init()
14.      {
15.          t1.setForeground(Color = Red);
16.          add(l1);
17.          add(t1);
18.          add(l2);
19.          add(t2);
20.          add(l3);
21.          add(t3);
22.          add(b);
23.          b.addActionListener(this);
24.      }
25.      public void actionPerformed(ActionEvent e)
26.      {
```

```
27.          if (e.getSource() == b)
28.          {
29.              int n1 = Integer.parseInt(t1.getText());
30.              int n2 = Integer.parseInt(t2.getText());
31.              t3.setText(" " + (n1 + n2));
32.          }
33.      }
34. }
```

HTML Code

```
1. <HTML>
2.
3.     <HEAD>
4.         <TITLE>WELCOME TO JAVA APPLET</TITLE>
5.     </HEAD>
6.
7.     <BODY>
8.         <CENTER>
9.             <H1>WELCOME TO THE APPLET</H1> </CENTER>
10.          <BR>
11.          <APPLET CODE=Q2.class WIDTH=400 HEIGHT=400> </APP
   LET>
12.      </BODY>
13.
14. </HTML>
```

OUTPUT


Applet Viewer: Q2.class
Applet
FIRST NO=:
10
SECOND NO:
20
SUM:
30
ADD
Applet started.