

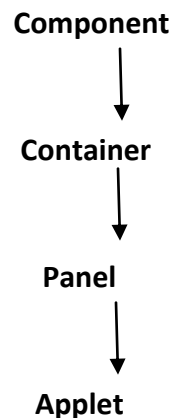
Applet

Java supports two types of programming application and applet. In application programming programs are executed through command prompt. Applets are run through browser applet is a little [application](#).

Applet is an Application designed to transfer over the internet and executed by the any java compatible web browser. It is an intelligent program dynamically acts as action upon that user input. It is an event revert application. Applet is not an console application it is not run in java runtime interpreter. Most of the applets are not executed through that main method.

All the applets classes subclass of Applet class .Applet class contains **java.applet**

Applet class is derived from **Panel** class, panel class is derived from **Container** class , container class derived from **Component** class.



LIFE CYCLE OF APPLET :

- `init()`
- `start()`
- `paint()`
- `stop()`
- `destroy()`

init() : It means Initialization . The first executed is **init** and it runs only one time. It is the function which belongs to Applet class. Any initialization variables are defined in **init** method.

start() : **Start** method is defined for any user defined actions. **Start** method called more than once ,not like as **init()** . **Start** method having actually action of Applet. It immediately called after **init** method. It is the function which belongs to Applet class.

paint() : It is the function which belongs to Component class. It is defined in the Component class. Which is used to display any text (or) Graphics in applets paint method having parameters.

public void paint(Graphics g)

stop() : It is the function which belongs to Applet class. Used to stop that applet action , that means suspending that applet.

destroy() : Termination of applet. Removing entire applet suspending that applet. It is the function which belongs to Applet class. Once the destroy method is called cannot be restarted by using **start()**.

COMPILE & RUNNING OF APPLET:

Applet is running in two different ways. Running in web browser & running in Applet viewer.

Applet code:

```
<applet code="classname.java" width=200 height="300">  
</applet>
```

Running by using applet viewer:

Appletviewer javaname.java

Advantages:

A Java applet can have any or all of the following advantages:

- It is simple to make it work with all operating systems. Applets are supported by most web browsers.
- The same applet can work on "all" installed versions of Java at the same time, rather than just the latest plug-in version only. However, if an applet requires a later version of the Java Runtime Environment (JRE) the client will be forced to wait during the large download.
- Most web browsers cache applets so will be quick to load when returning to a web page. Applets also improve with use: after a first applet is run, the JVM is already running and starts quickly (the JVM will need to restart each time the browser starts afresh).
- It can move the work from the server to the client, making a web solution more scalable with the number of users/clients.
- The applet naturally supports the changing user state, such as figure positions on the chessboard.

- Developers can develop and debug an applet directly simply by creating a main routine and calling `init()` and `start()` on the applet, thus allowing for development in their favorite Java SE development environment. All one has to do after that is re-test the applet in the AppletViewer program or a web browser to ensure it conforms to security restrictions.
- An untrusted applet has no access to the local machine and can only access the server it came from. This makes such an applet much safer to run than a standalone executable that it could replace. However, a signed applet can have full access to the machine it is running on if the user agrees.
- Java applets are fast - and can even have similar performance to native installed software.

Disadvantages:

A Java applet may have any of the following disadvantages:

- It requires the Java plug-in.
- Some browsers, notably mobile browsers running do not run Java applets.
- Some organizations only allow software installed by the administrators. As a result, some users can only view applets that are important enough to justify contacting the administrator to request installation of the Java plug-in.
- Some applets require a specific JRE.
- If an applet requires a newer JRE than available on the system, or a specific JRE, the user running it the first time will need to wait for the large JRE download to complete.
- Java automatic installation or update may fail if a proxy server is used to access the web. This makes applets with specific requirements impossible to run unless Java is manually updated.
- There is no standard to make the content of applets available to screen readers. Therefore, applets can harm the accessibility of a web site to users with special needs.

How to create an applet?

A basic Applet by extending Applet Class. You will need to embed another HTML code to run this program.

Sample program:

```
import java.applet.*;
import java.awt.*;
public class Hello extends Applet
{
    public void paint(Graphics g)
```

Notes by Pritee Parwekar

```
{  
    g.drawString("Welcome in Java Applet.",40,20);  
}  
}
```

To save the file name as class name is Hello

How to compile the program

1. Open command prompt to change the directory to your save program
2. Write javac class name.java
3. The compiler automatically create class file

How to create a html file

```
<html>  
  
<applet code="Hello.class" width=500 height=500>  
  
</applet>  
  
</html>
```

To save program as Hello.html

NOTE:

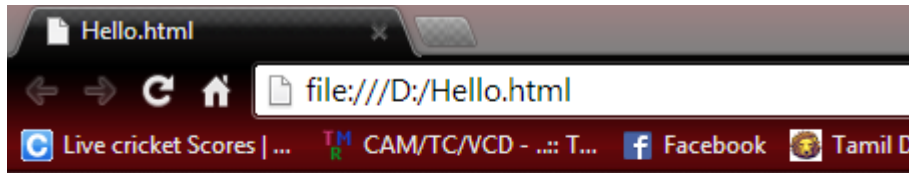
Your program, Html file & class file save same folder.

How to run a program?

1. First u can go your program location
2. To open html file in u r location

Output:

Notes by Pritee Parwekar



Welcome in Java Applet.

Program to create a calculator in java using Applet

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class cal extends Applet implements ActionListener
{
    int a,b,c;
    TextField t1;
    Button

    b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16;
    String s,s1,s2,s3,s4;
    public void init()
    {
        setLayout(null);
        t1=new TextField(10);
        t1.setBounds(80,200,260,40);
        add(t1);
        b1=new Button("0");
        b2=new Button("1");
        b3=new Button("2");
        b4=new Button("3");
        b5=new Button("4");
        b6=new Button("5");
```

Notes by Pritee Parwekar

```
b7=new Button("6");
b8=new Button("7");
b9=new Button("8");
b10=new Button("9");
b11=new Button("+");
b12=new Button("-");
b13=new Button("*");
b14=new Button("/");
b15=new Button("=");
b16=new Button("CLR");
b1.setBounds(90,260,40,30);
add(b1);
b2.setBounds(140,260,40,30);
add(b2);
b3.setBounds(190,260,40,30);
add(b3);
b4.setBounds(240,260,40,30);
add(b4);
b5.setBounds(290,260,40,30);
add(b5);
b6.setBounds(90,300,40,30);
add(b6);
b7.setBounds(140,300,40,30);
add(b7);
b8.setBounds(190,300,40,30);
add(b8);
b9.setBounds(240,300,40,30);
add(b9);
b10.setBounds(290,300,40,30);
add(b10);
b11.setBounds(90,340,40,30);
add(b11);
b12.setBounds(140,340,40,30);
add(b12);
b13.setBounds(190,340,40,30);
add(b13);
b14.setBounds(240,340,40,30);
add(b14);
b15.setBounds(290,340,40,30);
add(b15);
```

Notes by Pritee Parwekar

```
b16.setBounds(90,380,70,20);
add(b16);
b1.addActionListener(this);
b2.addActionListener(this);
b3.addActionListener(this);
b4.addActionListener(this);
b5.addActionListener(this);
b6.addActionListener(this);
b7.addActionListener(this);
b8.addActionListener(this);
b9.addActionListener(this);
b10.addActionListener(this);
b11.addActionListener(this);
b12.addActionListener(this);
b13.addActionListener(this);
b14.addActionListener(this);
b15.addActionListener(this);
b16.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
repaint();
s=ae.getActionCommand();
if(s.equals("0") || s.equals("1") || s.equals("2") ||
s.equals("3") || s.equals("4") || s.equals("5") || s.equals
("6") || s.equals("7") || s.equals("8") || s.equals("9"))
{
s1=t1.getText()+s;
t1.setText(s1);
}
if(s.equals("+"))
{
s2=t1.getText();
t1.setText("");
s3="+";
}
}
```

```
if(s.equals("-"))
{
s2=t1.getText();
t1.setText("");
s3="-";
}
```

```
if(s.equals("*"))
{
s2=t1.getText();
t1.setText("");
s3="*";
}
```

```
if(s.equals("/"))
{
s2=t1.getText();
t1.setText("");
s3="/";
}
```

```
if(s.equals("="))
{
s4=t1.getText();
a=Integer.parseInt(s2);
b=Integer.parseInt(s4);
if(s3.equals("+"))
c=a+b;
```

```
if(s3.equals("-"))
c=a-b;
```

```
if(s3.equals("*"))
c=a*b;
if(s3.equals("/"))
c=a/b;
t1.setText(String.valueOf(c));
}
```

```
if(s.equals("CLR"))
```


Notes by Pritee Parwekar

```
{  
  
t1.setText("");  
  
}  
  
}  
public void paint(Graphics g)  
{  
setBackground(Color.pink);  
g.drawRect(80,200,260,200);  
showStatus("ASHUSOFTECH");  
g.drawString("CALCULATOR",200,50);  
}  
}
```

Compile the program

Javac cal.java

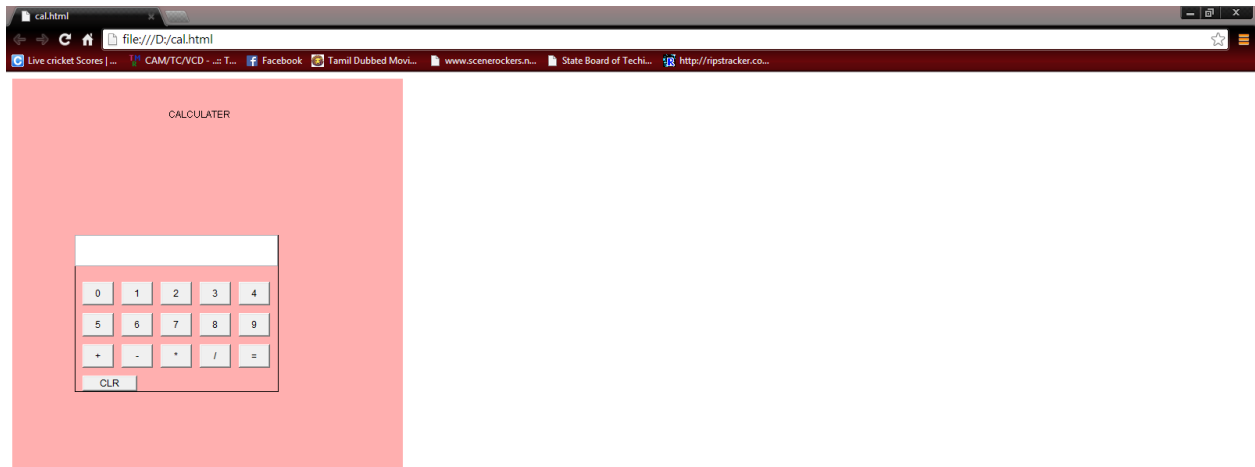
To create a html page

```
<html>  
  
<applet code="cal.class" width=500 height=500>  
  
</applet>  
  
</html>
```

To save a program as cal.html

Output:

Notes by Pritee Parwekar



Program to draw Rectangle , Square , Ellipse , Circle using applet

```
/*<applet code="pro.java" width =1200 height=1300>
```

```
</applet>*/
```

```
import java.awt.*;
```

```
import java.applet.*;
```

```
public class pro extends Applet
```

```
{
```

```
public void paint(Graphics g)
```

```
{
```

```
g.drawRect(100,100,60,60);
```

```
g.drawRect(300,300,60,100);
```

```
g.drawOval(400,400,120,120);
```

```
g.drawOval(500,500,120,80);
```

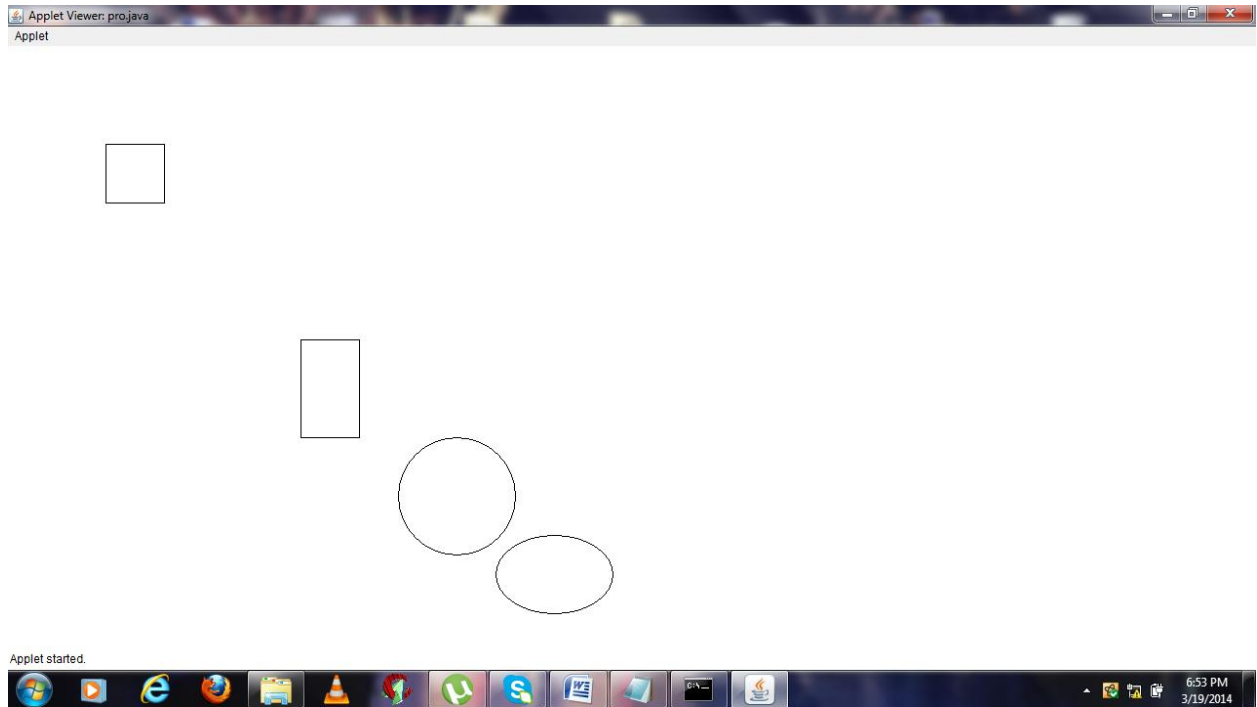
```
}
```

```
}
```

Notes by Pritee Parwekar

Output:

Output might be like this.



Notes by Pritee Parwekar