Clipping:

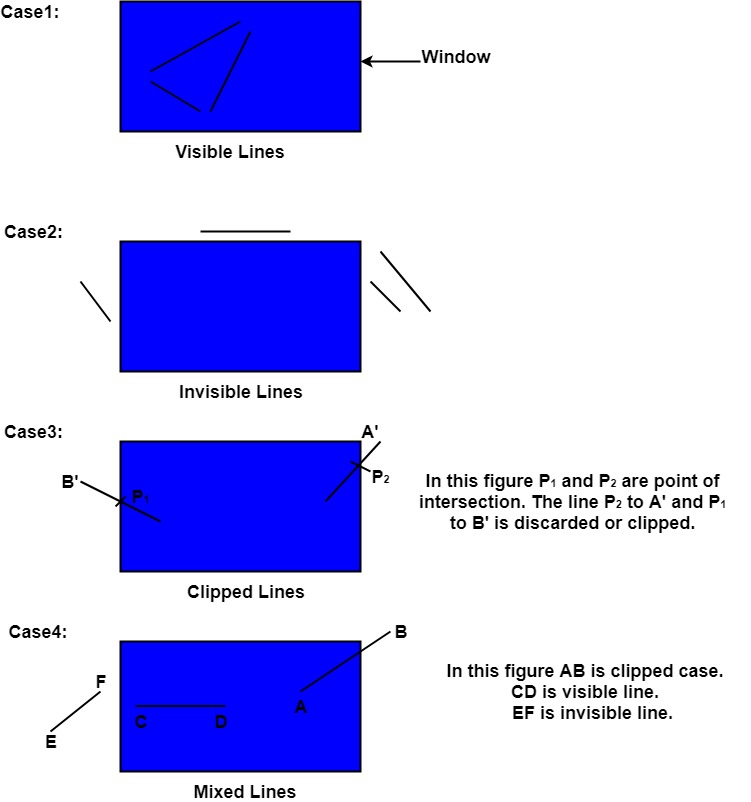
When we have to display a large portion of the picture, then not only scaling & translation is necessary, the visible part of picture is also identified. This process is not easy. Certain parts of the image are inside, while others are partially inside. The lines or elements which are partially visible will be omitted.

For deciding the visible and invisible portion, a particular process called clipping is used. Clipping determines each element into the visible and invisible portion. Visible portion is selected. An invisible portion is discarded.

Types of Lines:

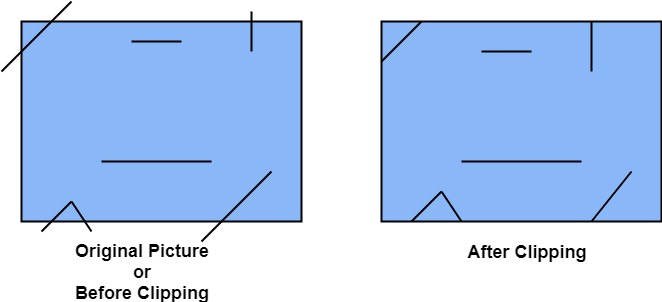
Lines are of three types:

1. **Visible:** A line or lines entirely inside the window is considered visible
2. **Invisible:** A line entirely outside the window is considered invisible
3. **Clipped:** A line partially inside the window and partially outside is clipped. For clipping point of intersection of a line with the window is determined.



Clipping can be applied through hardware as well as software. In some computers, hardware devices automatically do work of clipping. In a system where hardware clipping is not available software clipping applied.

Following figure show before and after clipping



The window against which object is clipped called a clip window. It can be curved or rectangle in shape.

Applications of clipping:

1. It will extract part we desire.
2. For identifying the visible and invisible area in the 3D object.
3. For creating objects using solid modeling.
4. For drawing operations.
5. Operations related to the pointing of an object.
6. For deleting, copying, moving part of an object.

Clipping can be applied to world co-ordinates. The contents inside the window will be mapped to device co-ordinates. Another alternative is a complete world co-ordinates picture is assigned to device co-ordinates, and then clipping of viewport boundaries is done.

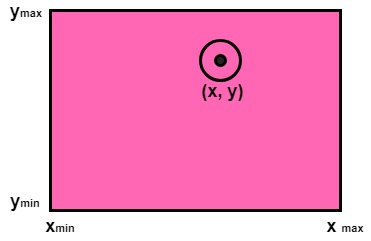
Types of Clipping:

1. Point Clipping
2. Line Clipping
3. Area Clipping (Polygon)
4. Curve Clipping
5. Text Clipping
6. Exterior Clipping

Point Clipping:

Point Clipping is used to determining, whether the point is inside the window or not. For this following conditions are checked.

1. x ≤ xmax
2. x ≥ xmin
3. y ≤ ymax
4. y ≥ ymin



The (x, y) is coordinate of the point. If anyone from the above inequalities is false, then the point will fall outside the window and will not be considered to be visible.

Line Clipping:

It is performed by using the line clipping algorithm. The line clipping algorithms are:

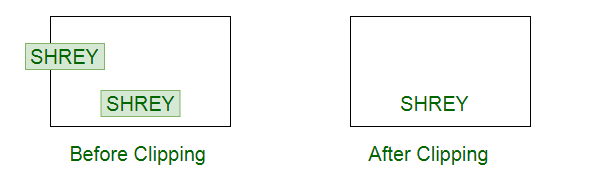
1. Cohen Sutherland Line Clipping Algorithm
2. Midpoint Subdivision Line Clipping Algorithm
3. Liang-Barsky Line Clipping Algorithm

**Methods for Text Clipping in Computer Graphics**

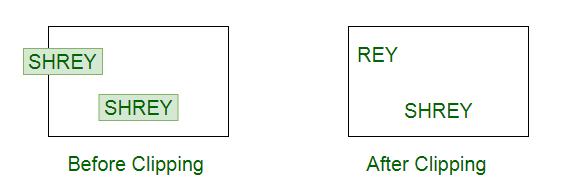
**Text Clipping** is a process of clipping the string. In this process, we clip the whole character or only some part of it depending upon the requirement of the application.

**Text clipping Methods :**

1. **All or None String Clipping method –**  
   In this method, if the whole string is inside the clip window then we consider it. Otherwise, the string is completely removed. Text pattern is considered under a bounding rectangle. The boundary positions of the rectangle are then compared to the window boundaries. String is rejected if there is any overlapping between the string and the window. This method produces the fastest text clipping.



1. **All or None Character Clipping method –**  
   In this method, we keep the characters of the string which lies inside clip window and remove all the characters which lie outside the clip window. The boundary limits of individual characters are compared to the window. In case of overlapping of character with the clip window, we remove the character.



1. **Text Clipping method –**  
   In this method, we keep the characters of the string which lies inside the clip window and remove all the characters which lie outside the clip window. If a character overlaps the window boundary then we keep that part of the character which lies inside the window and discard that part which lies outside the clip window.



Curve Clipping:

In computer graphics, we often need to draw different types of objects onto the screen. Objects are not flat all time and we need to draw curves many times to draw an object.

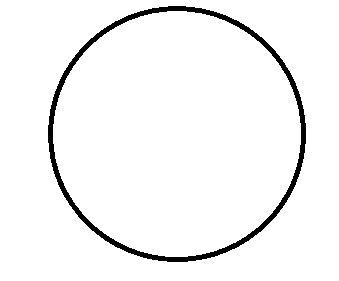
## Types of Curves:

The curve is an infinitely large set of points. Each point has two neighbors except endpoints.

1. Implicit curves
2. Explicit curves
3. Parametric curves
4. Bezier curves
5. B-spline curves

#### Implicit Curves:

An implicit curve or surface is the set of zeros of a function of 2 or 3 variables. We use implicit curve functions to define lines and planes. Provides no control over tangents at connection points when joining several implicit functions. Implicit functions are hard to find for many shapes. Use a function that states which points are on and off the curves.



All lines: Ax+By+C=0

In three dimensions f(X,Y, Z) defines a surface.

* Any plane Ax+By+ +D=0, with constants a,b,c, and d.
* A sphere centered at the origin with a radius:

Curves in 3D are not so easily represented in implicit form. In general, we cannot solve for points that satisfy the implicit form.

* Implicit function form – f(x,y)=0
* The implicit representation for the circle is:  X^2+Y^2-R^2=0

#### Explicit curves:

* Do not allow for multiple values for a given argument
* Cannot describe vertical tangents, as infinite slopes are hard to represent.
* Cannot represent all curves (vertical lines, circles)

Where y is the dependent variable, and x is the independent variable.

**Mathematical function:**

*f = y(x) can be plotted on curve*

*eg: y = 2X^5+3X^4*

*y=mx+c*

#### Parametric curves:

Curves have a parametric form called parametric curves. A curve in the plane is said to be parameterized if the set of coordinates on the curves (x,y,z) is represented as a function of a variable t. The variable t is called a parameter and the relations between x,y,z, and t are called a parametric equation

The parametric form of a curve is a function that assigns a position to values of the free parameters. That the parametric function is a vector-valued function. This example is a 2D curve, so the output of the function is a 2-D vector, in 3D it would be a 3 vector. It is simple and flexible

The parametric form is suitable for representing closed and multivalued curves. In parametric curves, each coordinate of a point on a curve is represented as a function of a single parameter. There are many curves that we cannot write down as a single equation in terms of x and y. The position vector of a point on the curve is fixed by the value of the parameter. Since a point on a parametric curve is specified by a single value of the parameter, the parametric form is axis-dependent. The function of each coordinate can be defined independently

Eg: x=acost; y=asint

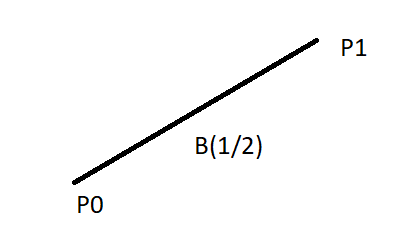
#### Bezier curves:

A bezier curve is particularly a kind of spline generated from a set of control points by forming a set of polynomial functions. Discovered by the french engineer Pierre bezier. These functions are computed from the coordinates of the control points. These curves can be generated under the control of other points. Tangents by using control points are used to generate curves.

It is an approximate spline curve. A bezier curve is defined by the defining polygon. It has no properties that make them highly useful and convenient for curve and surface design.

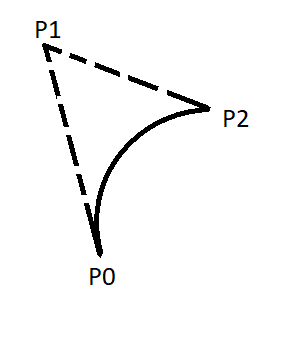
Different types of curves are Simple, Quadratic, and Cubic.

1. Simple curve: Simple bezier curve is a straight line from the point.



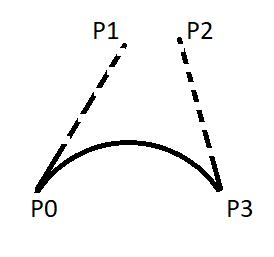
*Simple*

      2. Quadratic curve: Quadratic bezier curve is determined by three control points.



*Quadratic*

    3. Cubic curve: The cubic bezier curve is determined by four control points.



*cubic*

### Properties of Bezier Curve:

1. Bezier curves are widely available and used in various CAD systems, in general graphics packages such as GL
2. The slope at beginning of the curve is along the line joining the first two control points and the slope at the end of the curve is along the line joining the last two points
3. Bezier curve always passes through the first and last points i.e p(o)=po, p(1,=pnlie)
4. The curves lies entirely within the convex hall formed by the four control points
5. The slope at the beginning of the curve is along the line joining the first two control points and the slope at the end of the curve is along the line joining the last two points.
6. The degree of polynomial defining the curve segment is one less than the no of defining the polygon.

**Sutherland-Hodgeman Polygon Clipping:**

Sutherland-Hodgeman Polygon Clipping Algorithm :

* Read coordinates of all vertices of the polygon.
* Read coordinates of the clipping window
* Consider the left edge of the window
* Compare the vertices of each edge of the polygon, individually with the clipping plane
* Save the resulting intersections and vertices in the new list of vertices according to four possible relationships between the edge and the clipping boundary discussed earlier.
* Repeat the steps 4 and 5 for remaining edges of the clipping window. Each time the resultant list of vertices is successively passed to process the next edge of the clipping window.
* Stop.

It is performed by processing the boundary of polygon against each window corner or edge. First of all entire polygon is clipped against one edge, then resulting polygon is considered, then the polygon is considered against the second edge, so on for all four edges.

**Four possible situations while processing**

1. If the first vertex is an outside the window, the second vertex is inside the window. Then second vertex is added to the output list. The point of intersection of window boundary and polygon side (edge) is also added to the output line.
2. If both vertexes are inside window boundary. Then only second vertex is added to the output list.
3. If the first vertex is inside the window and second is an outside window. The edge which intersects with window is added to output list.
4. If both vertices are the outside window, then nothing is added to output list.

Following figures shows original polygon and clipping of polygon against four windows.

