Automation of SVG Animation

A project report

Submitted in partial fulfillment of the requirements for the Degree of Master of Computer Application

of

Department of Computer Science and Engineering

Jadavpur University

by

Neha Kumari

Class Roll No.: 001910503016

Registration No.: 149879 of 2019-2020

Examination Roll No.: MCA226016

Under the Guidance of

Prof.(Dr.) Chintan Kumar Mandal

Dr. Debajyoti Sarkar

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2022

Faculty of Engineering And Technology

Jadavpur University

Certificate of Recommendation

This is to certify that the project report entitled “Automation of SVG Animation” has been carried out by Neha Kumari (University Registration No.:149879 of 2019-2020, Examination Roll No.: MCA226016) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Master of Computer Application of the Department of Computer Science and Engineering, Jadavpur University. The research results presented in the project report have not been included in any other paper submitted for the award of any degree in any other University or Institute.

Prof. (Dr.) Chintan Kumar Mandal (Project Supervisor)
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

Countersigned

Prof. Anupam Sinha
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-32

Prof. Chandan Mazumdar
Dean, Faculty of Engineering and technology,
Faculty of Engineering And Technology
Jadavpur University

Certificate of Approval

This is to certify that the project report entitled "Automation of SVG Animation" is a bona-fide record of work carried out by Neha Kumari in partial fulfillment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University during the period of January 2022 to June 2022. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the project report only for the purpose for which it has been submitted.

____

Signature of Examiner 1

Date:

____

Signature of the Examiner 2

Date:

Faculty of Engineering And Technology

Jadavpur University

Declaration of originality and compliance of academic ethics

I hereby declare that this project report entitled "Automation of SVG Animation" contains a literature survey and original research work by the undersigned candidate, as part of her degree of Master of Computer Application. All information has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully referenced all materials and results that are not original to this work.

Name: Neha Kumari

Registration No.: 149879 of 2019-2020

Exam Roll No.: MCA226016

Project Title: Automation of SVG Animation

Signature with date

Jadavpur University

Abstract

Faculty of Engineering and Technology, Jadavpur University
Computer Science and Engineering

Master of Computer Application

Automation of SVG Animation

by Neha Kumari

Animation is one of the best ways to educate the audience about any topic through entertainment. Animation gives us the ability to transfer emotion efficiently. Nowadays they are being used in almost every field whether it's education,business or entertainment industry.

As the market has become vast and growing rapidly every day, automating the animation is really proving to be a great aid as the animators do not have to draw frame by frame.

With the help of SVG the whole process has become more efficient. The SVG animation is important as SVG looks crisp and pixel- perfect even on high resolution and responsive.

The SVG animation that has been created, is accessible through a web browser that could be running on any operating system. The development process includes a Text Editor notepad++ , inkscape vector graphic editor for this project and a runtime web browser (Google Chrome).

Acknowledgements

The writing of the project report as well as the related work has been a long journey with input from many individuals, right from the first day till the development of the final project. I would like to express my deepest gratitude to my supervisor, Dr. Chintan Kumar Mandal, Professor, department of computer science and engineering, Jadavpur University for giving me the opportunity to do research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the work and to present the works as clearly as possible. I would like to express my sincere, heartfelt gratitude to Dr. Debajyoti Sarakar, his patience, guidance, suggestions and moral support in times of need. It was a great privilege and honour to work and study under their guidance. I am also particularly thankful to Prof. Anupam Sinha, Head of the department of computer science and engineering, Jadavpur University for allowing us to carry out research in the department.

I would also like to thank all the faculty members of the department of computer science and engineering of Jadavpur University for their continuous support. This project report would not have been completed without the inspiration and support of a number of wonderful individuals including my batchmates of Master of Computer Application in Jadavpur University — my thanks and appreciation to all of them for being part of this journey and making this project report possible.

____

Neha Kumari

Examination Roll No.: MCA226016

Registration No.: 149879 of 2019-2020

Class Roll No.: 001910503016

CONTENTS

1. INTRODUCTION

In the era of fast-paced technical development and constantly shifting trends, SVG animations are becoming a necessity.

SVG has been in development within the World Wide Web Consortium (W3C) since 1999 after six competing proposals for vector graphics languages had been submitted to the consortium during 1998.

SVG allows three types of graphic objects: Vector Graphic shapes (such as paths consisting of straight lines and curves), bitmap images, and text.

The Automation of SVG Animation is done as it helps in skipping the process of creating each step in animation movement. We just need to define point A and B and the rest will be done by software.There are various softwares nowadays that are used for the automation of animation.

GSAP can also provide very stunning SVG animations with very little or no knowledge of Javascript. It is one of the most robust JS animation libraries.
It offers a large amount of functionality for dealing with SVG like morphing SVG shapes ,drawing SVG paths,dragging ,dropping etc.

In this Project GSAP is used to create animations that look quite engaging and also interactive. For animation of SVG path Regular expression is used to search and list all the paths involved in the animation and then we are allocating IDs to aloof those paths dynamically.

2. CSS ANIMATIONS

A CSS animation is a method of animating (or changing the style of an element) on a web page.[1][2]
CSS incorporates a range of powerful predefined attributes that make it a more well-known and speedier tool for animations than javaScript.
Javascript can also produce spectacular animations, but it is regarded as a more expensive option due to various factors like code complexity, cross-browser compatibility, and processing speed.

CSS animations can be broken down into three categories:
1)Transformations 2)Transitions 3)Keyframes

2.1. CSS ANIMATIONS-TRANSFORMATIONS

This property enables elements on the web page to resize, rotate, translate, or skew.

Translate:
- translate(value,value): To move an element both along the x and y axes.
- translate(value): The element is translated along both axes with the same value.
- translateX(value): Translates solely along the x-axis.
- translateY(value): Translates along a specified axis.

Scale:
- Scale: To scale an element in both the x and y directions.
- scale(value):The element is scaled along both axes with the same value.
- scaleX(value):Only the x-axis is scaled.
- ScaleY(value):Scaled solely along the y-axis.

Rotate:
- rotate(angle): Rotates an element at a specific angle.
- rotate(angle): Rotates an element along the x-axis at a specified angle.
- rotate(angle): Rotates an element along the y-axis at a specified angle.

Skew:
- skew(x, y): To skew an element on the x-axis and y-axis at an angle of x and y, respectively.
- skewX(x): To skew an element on the x-axis at an angle of x.
- skewY(y): To skew an element on the y-axis at an angle of y.

2.2. CSS ANIMATIONS-TRANSITIONS:

Transitions are the modifications in style of an element from beginning to conclusion that provide the impression that it is moving or being animated.

There are four distinct transition properties:

- **Transition-property:** The value of this attribute determines which property will be used.

- **transition-duration:** It indicates how long a transition will take to complete.
- The **transition-timing-function** specifies the transition's speed.
- **transition-delay:** Defines the amount of time that must pass before the transition can begin.

2.3. CSS ANIMATIONS - KEYFRAMES:

Keyframes are used to connect various animations together. It specifies the name of the animation as well as the timing, location, and subject matter.

Keyframe animation can be described in two ways:

- using from-to Keyword: It is used to specify animation that switches from one supplied value to another.
- We have flexibility while working with intricate animations because of the **percentage assignment**. For instance, 0% means that the animation will have the specified property at the beginning, while 50% means that the mentioned properties will differ at the midpoint of the animation.

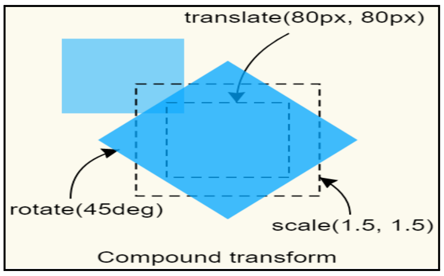


Fig. 2.1. Transformation of an element[3]

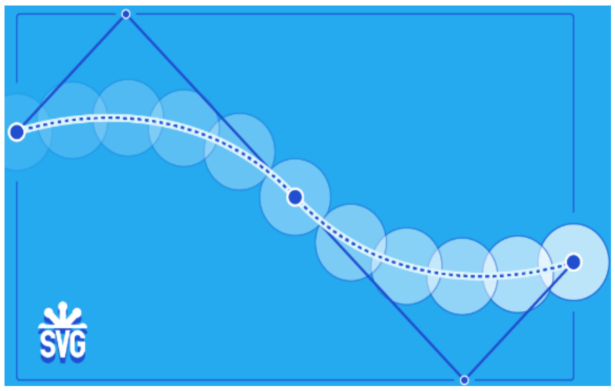


Fig. 2.2. Depicting transition and usage of keyframe[4]

3. POSITIONING OF HTML ELEMENT

Any HTML element can have its location changed by using the **position** attribute.[5] It can take one of six different values: **inherit, relative, sticky, fixed, or absolute**. The position property is combined with the left, right, bottom, and top attributes to adjust the element's positioning. These attributes describe how far away from the viewport the HTML element is.

Placement of element according to user input:

By allocating ID to an element we can manipulate it with the help of HTML DOM methods using javaScript. In this case **document.getElementById()** (It returns the element with specified value) can be used to manipulate the style of the element and then the HTML element can be positioned using left, right, bottom, top properties as per values provided by the user with the help of a JS function.

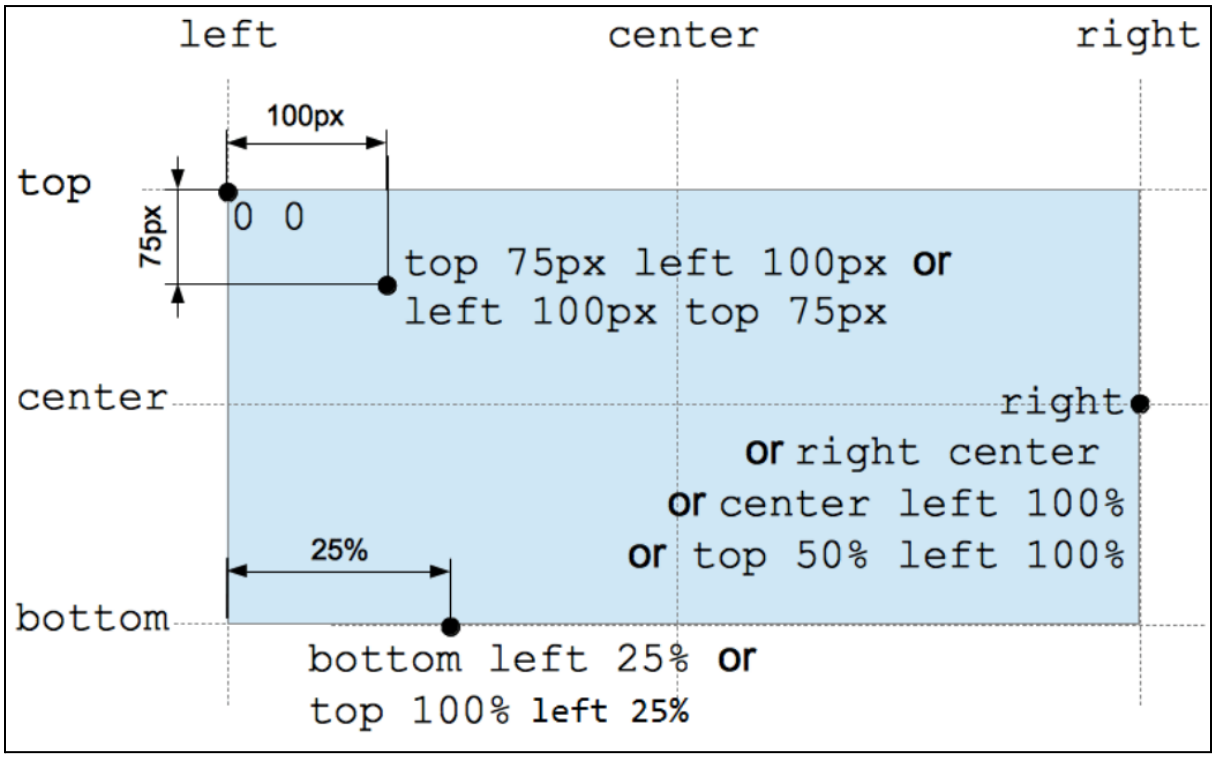


Fig. 3.1. Positioning of an element[6]

4. SVG(SCALABLE VECTOR GRAPHIC)

SVG stands for "Scalable Vector Graphics," it is a vector picture format based on XML.[7]
Any picture, icon, or logo can be displayed in SVG format on a website. They can also be animated using CSS or JavaScript to make them more appealing.
An SVG is a vector picture format, which means it is made up of maths functions that can be drawn on screen after being interpreted.
With the aid of interactivity and animations, numerous types of images are displayed on the web using the lightweight vector image SVG.
When compared to other image formats, SVG has the advantage since svg images can be resized without losing quality and can be shown by any web browser.

4.1. SVG AND OTHER IMAGE FORMATS

On the internet we work with two kinds of images: **raster and vector.**[8]
Raster image file contains the data regarding wherever every component ought to be situated on the screen and what colour that component can have.
Whereas an **SVG** image is created of shapes and path definitions that the PC will use to figure out what the pictures ought to appear as if once rendered on the screen.
Raster and vector images are the two types of images we use on the web.
The information about where each component should be placed on the screen and what colour it can have is contained in the raster image file.

In contrast, an SVG image is made up of shapes and path specifications that the computer will use to determine how the images should look when they are produced on the screen.

- SVG pictures can be stretched indefinitely without losing sharpness at any level because of their **pixel perfect scaling.**
- **More compact files -**unlikely to be bitmap images SVG images have a smaller file size since they don't keep track of the number of colours used in each pixel or other aspect of the image.
- **CSS styling -** SVG pictures may be quickly and simply styled with CSS.
- SVG can be animated since it can be edited directly by a text editor and can also be worked with by javaScript.

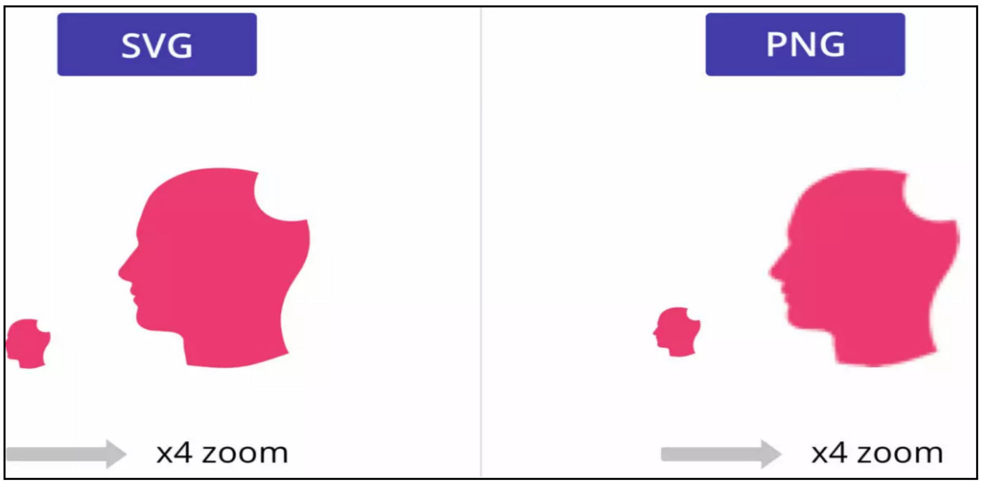


Fig. 4.1. SVG and PNG[8]

4.2. SVG and HTML

In contrast to other image formats that must be downloaded from outside sources, SVG can be integrated in HTML code, which is particularly advantageous because it allows for search, indexing, and caching.[9]
SVG can be inserted into HTML in a variety of ways.
Using the <img> tag is the best and easiest approach to embed SVG into HTML. Other embedded image formats use a similar syntax.
with the <object> tag.The <svg> tag can be used to insert SVG straight into HTML code.

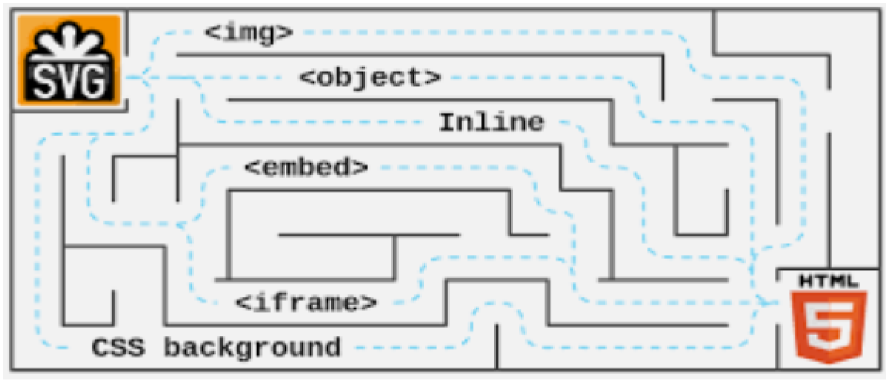


Fig. 4.2. Ways of embedding SVG in HTML[9]

4.3. POSITIONING OF SVG ELEMENT

SVG employs a grid structure for all of its elements, where the element's top left corner is regarded as the origin.[10] After that, positions are measured in pixels, with positive x and y axes pointing to the right and bottom, respectively. The positions of elements in HTML are also the same.

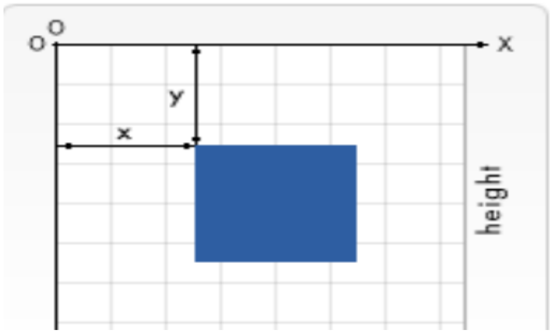


Fig. 4.3. Grid System[10]

Position of SVG elements can be manipulated as per values provided by the user using JavaScript. Javascript can be used to achieve this easily as user input can be taken with the help of javascript function and then the value of x and y coordinates of the SVG image can be set using the setAttribute() of javascript.SVG image can be manipulated using JS by provide a id to <image> (used to embed SVG)tag .

4.4. SVG SHAPES

SVG forms, such as circles, ellipses, and rectangles, are graphic elements defined by the combinations of straight lines or curves. All the fundamental components are fillable, strokeable , and clippable.[11]

- **Rectangle:** <rect> tag is used for creating a rectangle which is axis aligned with the current user system. The six attributes that define the position and shape of a rectangle are:
  x and y determine coordinates of the top left corner of the rectangle. Width and height define the respective properties of rectangle and rx and ry defines the x and y radius of corners.

- **Circle**: <circle> tag is used for creating circles.
  r defines the radius of the circle and cx and cy determines the coordinates of the centre of the circle.

- **Ellipse:** <ellipse> tag used for creating ellipses and the parameters that it takes to determine the shape are:
  rx and ry define the x and y radius of the ellipse respectively.cx and cy determine the x and y value of the centre of the ellipse.
- **Line**: <line> take the positions of two points to draw a straight line.
  X1, y1 are the x, y coordinates of the first point similarly x2, y2 are the x and y coordinates of the second point.

- **Polyline:** <polyline> is a group of connected lines as it may take a number of points to draw a polyline, so all the points are mentioned in the points attributes.
- **Polygon:**A polygon is a closed figure made up of straight lines that is drawn on a screen using the polygon> command; the points property lists all the points required to draw the figure.

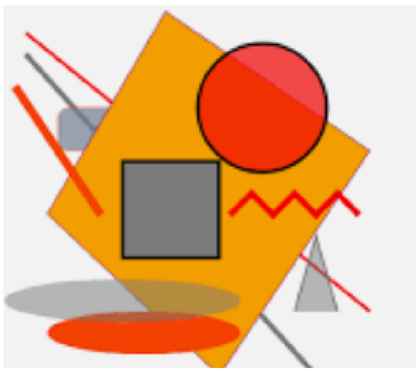


Fig. 4.4 Different SVG Shapes[12]

So,with the basic knowledge of SVG markup you can draw simple images in a text editor. For complex images we need drawing softwares.

4.5. SVG PATH

The most powerful element in the SVG library is the <path> element. SVG <path> can be used to create simple shapes with lines and curves as well as complex images. To create complex images, a solid understanding of SVG <path> is required.[13]

The d attribute is used to define all shapes created by the <path> element. The coordinates in the d attributes are unitless and in the user coordinate system. To draw complex images, the path data contains various instructions or commands such as moveto, lineto, curveto, and so on.

- The **moveto** command (abbreviated as m or M) helps in establishing a new initial point. A path data must begin with a **m** command. **M** defines the absolute coordinates and **m** specifies the relative coordinates.
- The **closepath** (denoted as Z or z) command helps in connecting the end point to its initial point and hence draws a closed figure.
- The **Lineto** (denoted as L) draws a line from the current coordinates to new coordinates and the new point becomes the current one.
- The **H** commands draw a horizontal line from the current point.
- The **V** command draws a vertical line from the current position.

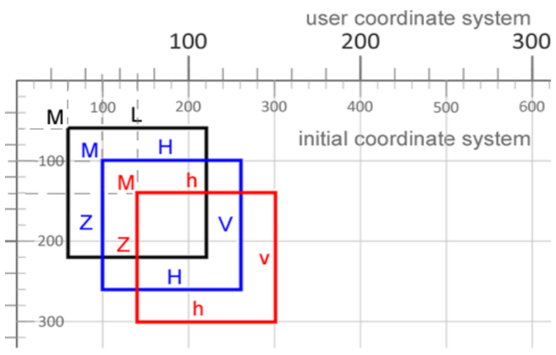


Fig. 4.5. Shapes drawn using different commands[12]

- **Curve commands:** Curve commands of three types are used with the <path> element to generate curves. The curveto creates a cubic Bézier curve from the current point to (x, y) by using (x1, y1) and (x2,y2) as control points at the beginning and end of the curve, respectively. The absolute coordinates are specified by C, while the relative coordinates are specified by c.

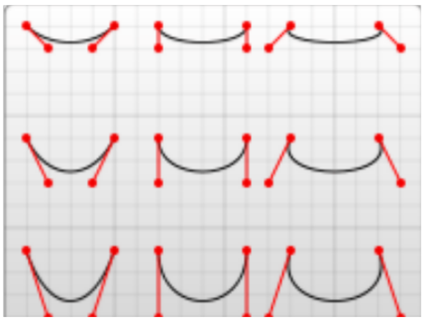


Fig. 4.6. SVG curves[14]

- Several Bézier curves can be combined to form long, smooth shapes. The control point on one side of a point is usually a reflection of the control point on the other side, which keeps the slope constant. In this case, a shorthand curveto command, denoted as, is used (s or S).

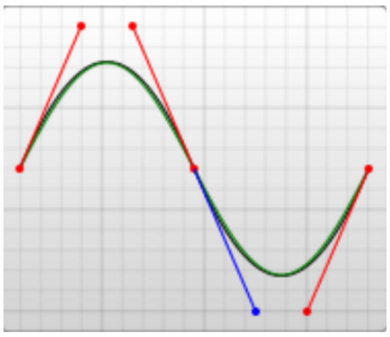


Fig. 4.7. Bezier curve[14]

- The **quadratic bézier** curve draws a quadratic bézier curve from the current point to (x, y) using (x1,y1) as the control point.

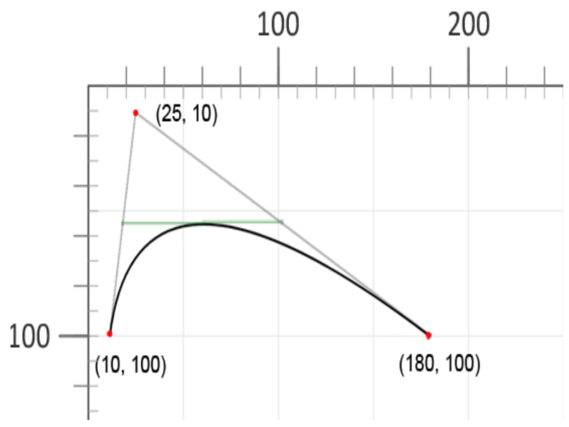


Fig. 4.8. Quadratic Bezier curve[12]

- The elliptical curve command, denoted by A, creates an elliptical arc from the current point to (x, y). The ellipse's size and orientation are defined by two radii (rx, ry) and an x-axis rotation, which determines how the ellipse will rotate relative to the current coordinate system.

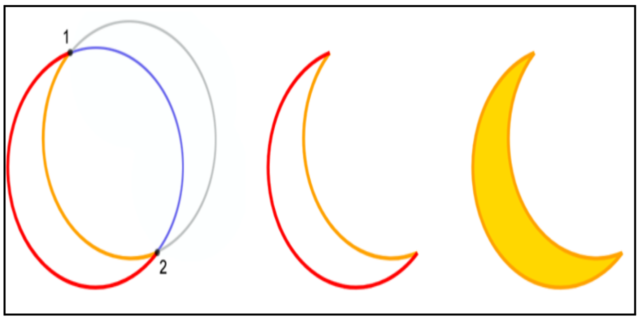


Fig. 4.9. Elliptical curve[12]

4.6. SVG ANIMATIONS WITH CSS

Similar to how CSS can be used to style and animate HTML components, SVG elements can also be animated and interactive using JS and CSS.However, combining CSS with SVG allows us the ability to create our own animations without the need for third-party libraries.
The viewable portion of the SVG is known as the **VIEWPORT** and is controlled by the height and width attributes.[15]
You can describe how the graphic will extend to fit the container element using VIEWBOX.

The min-x, min-y, width, and height attributes are used to set it. If the viewbox and viewport don't have the same width to height ratio, **PreserveAspectRatio** instructs the browser to display an SVG image.

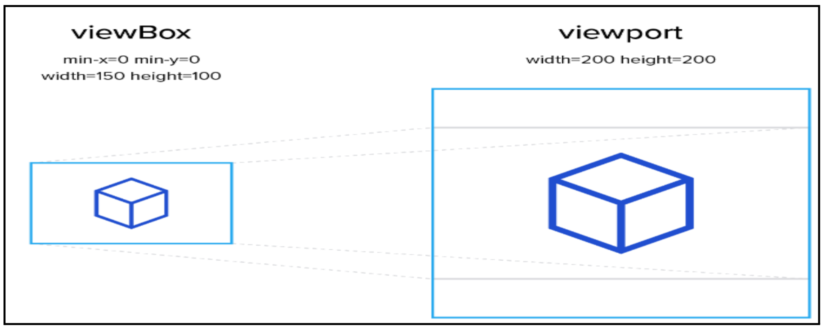
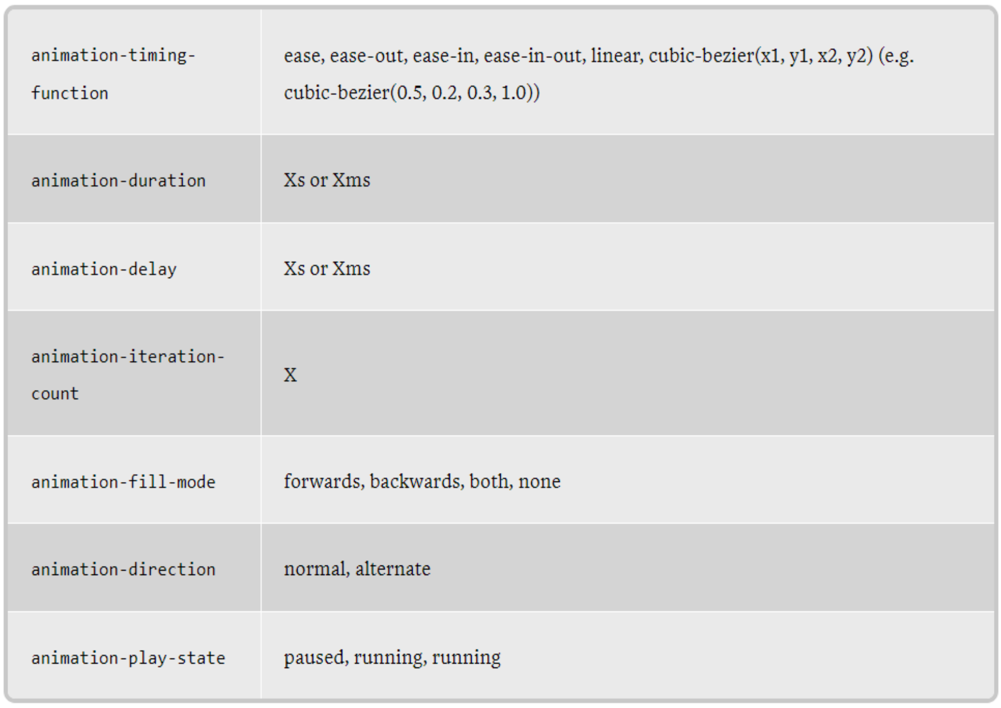


Fig. 4.10. Viewport and Viewbox[16]

The two most crucial elements of animation are rate and duration. The CSS transition property enables us to define these attributes so that elements move gradually rather than abruptly from beginning to conclusion.

We may lengthen the animation's runtime using only the transition attribute, but we are unable to include more keyframes. With CSS animation, we may use endless loops and many keyframes. Every keyframe aids in explaining the values of one or more CSS attributes to the desired element at that certain moment in time. It accepts functions for delay, duration, and ease.



`animation-timing-function`	ease, ease-out, ease-in, ease-in-out, linear, cubic-bezier(x1, y1, x2, y2) (e.g. cubic-bezier(0.5, 0.2, 0.3, 1.0))
`animation-duration`	Xs or Xms
`animation-delay`	Xs or Xms
`animation-iteration-count`	X
`animation-fill-mode`	forwards, backwards, both, none
`animation-direction`	normal, alternate
`animation-play-state`	paused, running, running

Fig. 4.11. List of CSS animation property[17]

The SVG elements can be scaled, positioned, and oriented with the aid of CSS transform.The orientation or size of the desired element can be adjusted with the aid of the translateX, translateY, and rotate functions.

Modern CSS animation techniques allow us to create a wide array of engaging and polished animations in a simple, cross-browser-compatible way.

4.7. SVG PATH ANIMATION

An **SVG path** is a continuous line defined by coordinates contained in it. SVG path animation is used to create a self drawing effect. The ‘drawing’ is nothing but an optical trick created by manipulating SVG paths.[18]

Due to its vector and path capabilities, SVG format is a wonderful fit for our purposes because it allows for the creation of delicate line art. And, it turns out, not just our technological needs, but also our aesthetic needs.

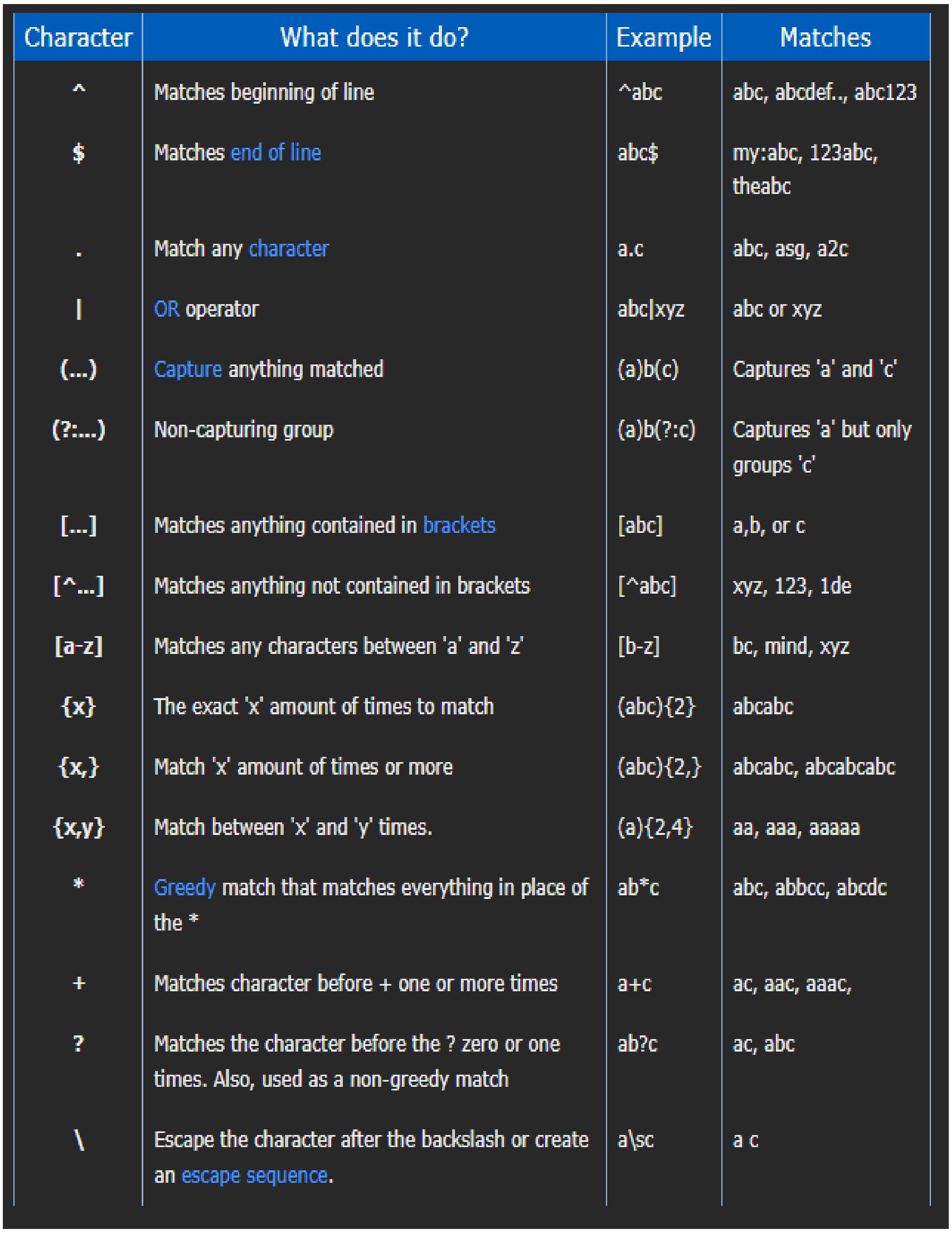
The ‘animation’ of path is an illusion created by adjusting stroke dasharray and dashoffset.

Stroke-dasharray: It is a comma delimited list that controls the gaps that make a path.
For example, if the path's stroke is set to "10 10," the path's stroke alternates between a dash of 10 and a gap of 10.The first "dash" is equal in length to the path itself if the path's overall length is 10.
This means that a path with no stroke-dasharray set and a path with a stroke-dasharray defined where the dash is equal to the path's whole length have the same visual appearance.

Stroke-dashoffset: It regulates the offset of the path's gaps and dashes. Here, offset refers to the starting point of the quick dash produced by dasharray. Since 1 is its default setting, it means that it will begin right away.
We can influence these two characteristics to achieve the outcomes we want.

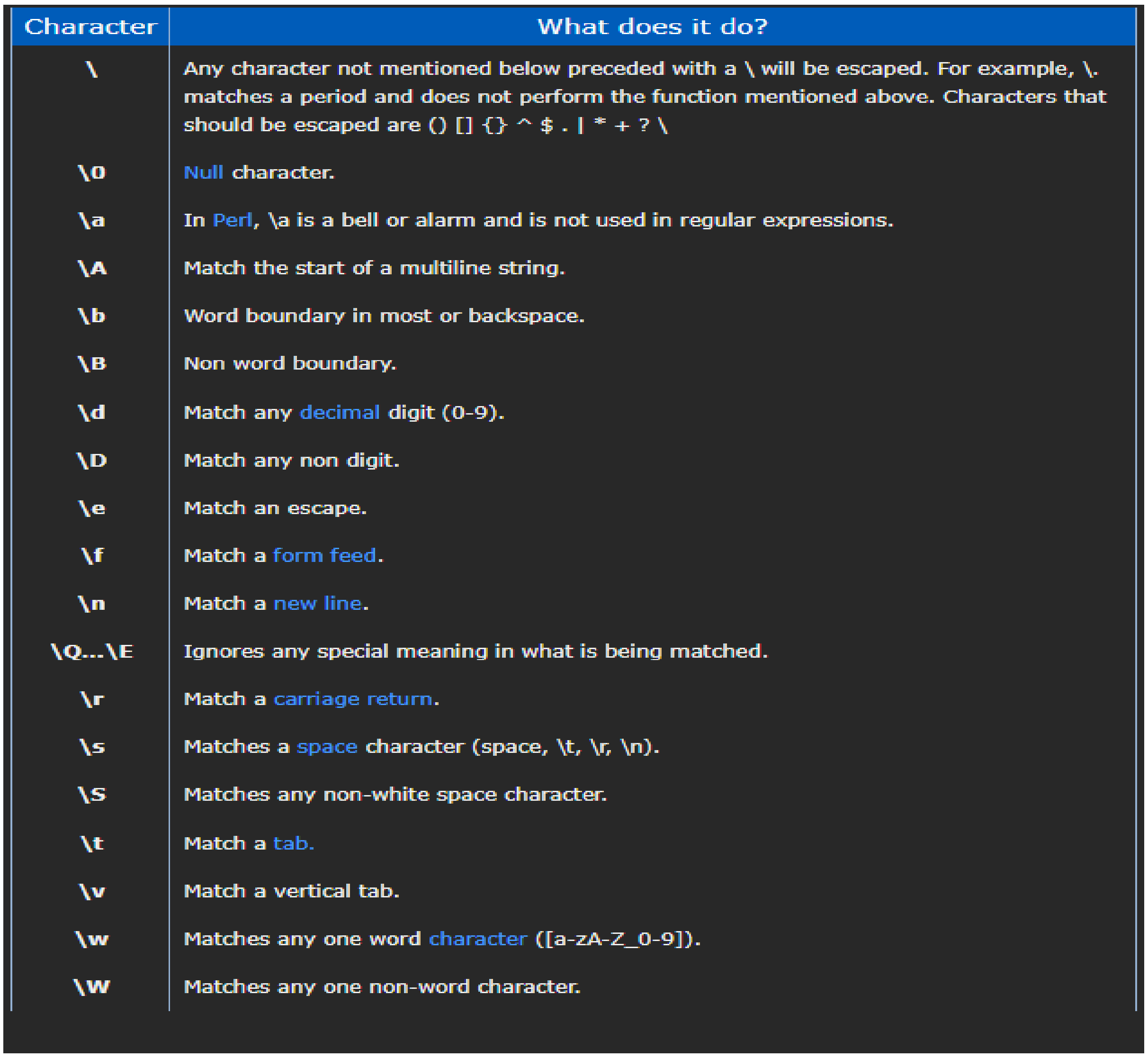
5. REGULAR EXPRESSIONS

For text searching, pattern matching, managing text, etc., regular expression is a very effective and valuable tool.[19]
Regex, which stands for regular expression, can be used from text editors or the command line.The table below shows which character has what role.



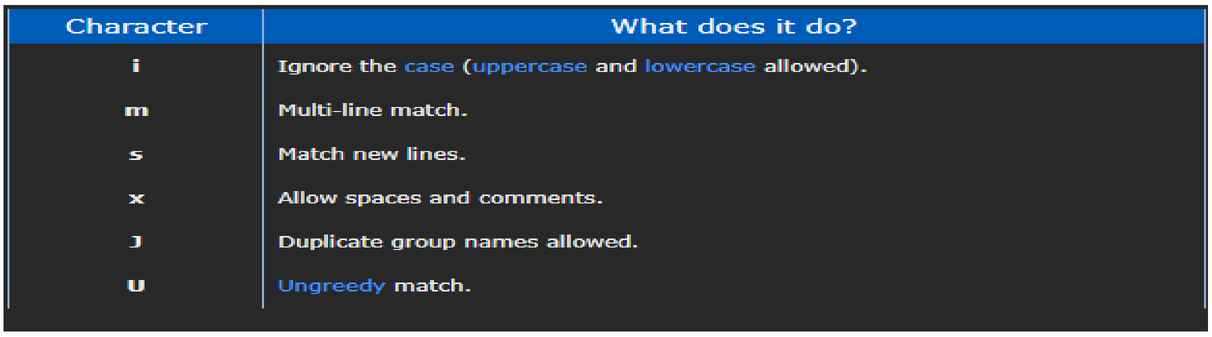
Character	What does it do?	Example	Matches
^	Matches beginning of line	^abc	abc, abcdef., abc123
$	Matches end of line	abc$	my:abc, 123abc, theabc
.	Match any character	a.c	abc, asg, a2c
|	OR operator	abc|xyz	abc or xyz
(...)	Capture anything matched	(a)b(c)	Captures 'a' and 'c'
(?:...)	Non-capturing group	(a)b(?:c)	Captures 'a' but only groups 'c'
[...]	Matches anything contained in brackets	[abc]	a,b, or c
[^...]	Matches anything not contained in brackets	[^abc]	xyz, 123, 1de
[a-z]	Matches any characters between 'a' and 'z'	[b-z]	bc, mind, xyz
{x}	The exact 'x' amount of times to match	(abc){2}	abcabc
{x,}	Match 'x' amount of times or more	(abc){2,}	abcabc, abcabcabc
{x,y}	Match between 'x' and 'y' times.	(a){2,4}	aa, aaa, aaaaa
*	Greedy match that matches everything in place of the *	ab*c	abc, abbcc, abcdc
+	Matches character before + one or more times	a+c	ac, aac, aaac,
?	Matches the character before the ? zero or one times. Also, used as a non-greedy match	ab?c	ac, abc
\	Escape the character after the backslash or create an escape sequence.	a\sc	a c

Table 5.1.[19]



Character	What does it do?
\	Any character not mentioned below preceded with a \ will be escaped. For example, \. matches a period and does not perform the function mentioned above. Characters that should be escaped are () [] {} ^ $. | * + ? \
\0	Null character.
\a	In Perl, \a is a bell or alarm and is not used in regular expressions.
\A	Match the start of a multiline string.
\b	Word boundary in most or backspace.
\B	Non word boundary.
\d	Match any decimal digit (0-9).
\D	Match any non digit.
\e	Match an escape.
\f	Match a form feed.
\n	Match a new line.
\Q...\E	Ignores any special meaning in what is being matched.
\r	Match a carriage return.
\s	Matches a space character (space, \t, \r, \n).
\S	Matches any non-white space character.
\t	Match a tab.
\v	Match a vertical tab.
\w	Matches any one word character ([a-zA-Z_0-9]).
\W	Matches any one non-word character.

Table 5.2.[19]



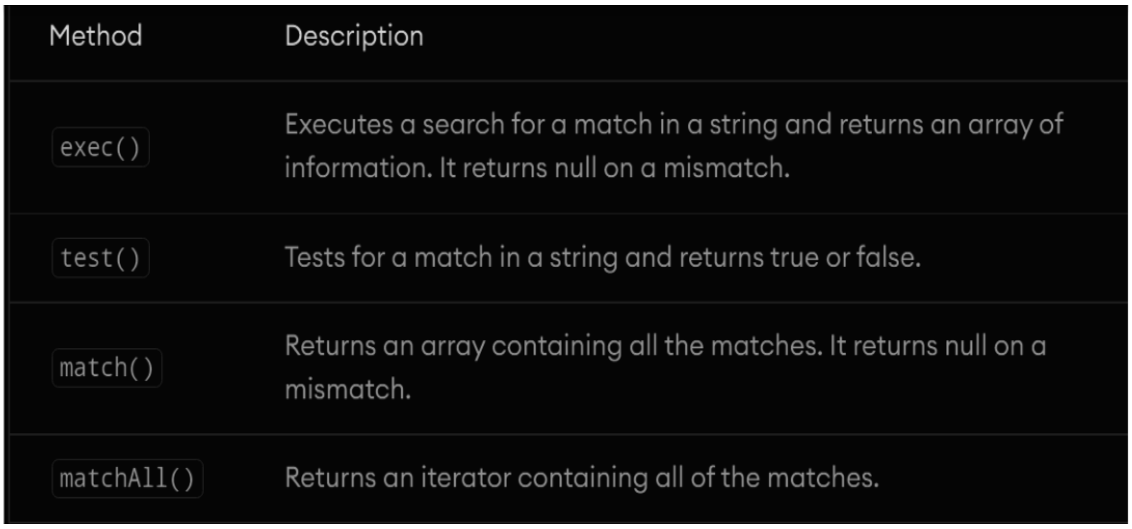
Character	What does it do?
i	Ignore the case (uppercase and lowercase allowed).
m	Multi-line match.
s	Match new lines.
x	Allow spaces and comments.
J	Duplicate group names allowed.
U	Ungreedy match.

Table 5.3.[19]

There are two ways to create Regex:

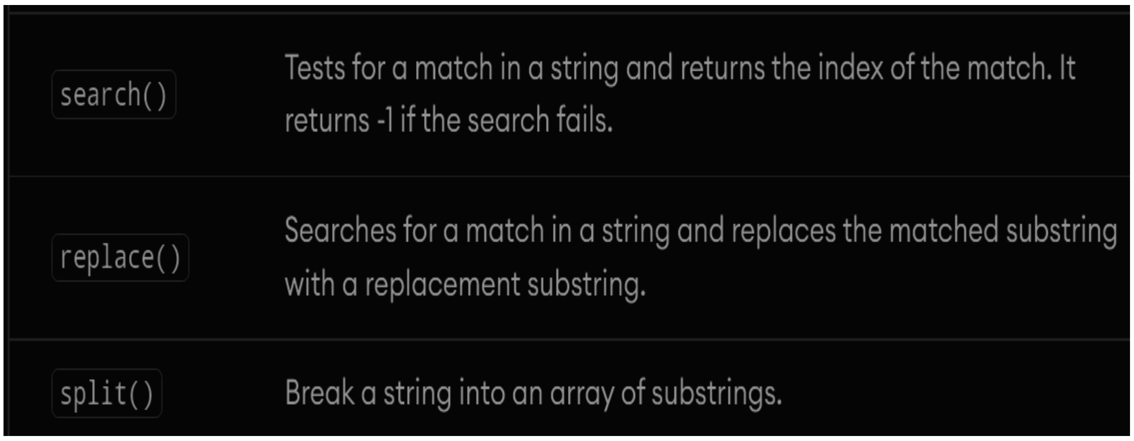
- The regular expression consists of a pattern between the slashes.
- Using the RegExp() constructor function.

There are various string methods that allow us to perform different operations on our test string.The table below mentions their functions:



Method	Description
`exec()`	Executes a search for a match in a string and returns an array of information. It returns null on a mismatch.
`test()`	Tests for a match in a string and returns true or false.
`match()`	Returns an array containing all the matches. It returns null on a mismatch.
`matchAll()`	Returns an iterator containing all of the matches.

Table 5.4.[20]



`search()`	Tests for a match in a string and returns the index of the match. It returns -1 if the search fails.
`replace()`	Searches for a match in a string and replaces the matched substring with a replacement substring.
`split()`	Break a string into an array of substrings.

Table 5.5.[20]

5.1. ALLOCATING ID TO SVG PATH WITH REGEX:

We very well can use regular expressions for dynamic id allocation to SVG paths. After constructing a suitable regular expression the list of all the paths can be searched with the help of exec().We can simply count the number of times exec() does not return null and that specifies the number of paths in SVG then with the help of a loop(number of iterations equal to the number of paths) keep allocating ID to each path at every iteration.

With the help of **regex** we allocated ID to svg paths, which can be used to animate all the paths. With the help of javaScript we can achieve this by using HTML DOM methods(say document.getElementById) and animate each path one by one at every iteration of the loop.

6. WHITEBOARD ANIMATION

Whiteboard animation is a particular kind of presentation where the author or creator sketches and records the illustrated narrative. The hand-drawn picture is made more lively in this type of animation approach by using stop-motion and time-lapse drawing techniques.[21]

6.1. PROCESS

Choosing a topic and composing a screenplay are the first steps in creating a whiteboard animation, which is then followed by a number of further procedures. So, basically, making these videos, which are frequently referred to as "video doodling," follows the procedures listed below.

- Make the material
- voiceover recording
- Make animated movies
- Put the animations in order.
- Make manuals
- Record the video and sync the sound and image
- Add music
- Share and export

6.2. APPLICATIONS

These animations may be produced for a very low price and are memorable and powerful.
Whiteboard cartoons are now widely used in ads.
It is utilised by online video platforms like YouTube and Vimeo.
Languages, chapters, and summaries are all explained using whiteboard animations in the classroom.

6.3. ADVANTAGES-

- **Memorable**-A whiteboard animation, according to research, helps viewers recall 15% more information than a regular video.
- **versatile**-Whiteboard animations are incredibly flexible and can be used for any subject.
- **Capable Of holding attention**- All of our senses are stimulated by the combination of sound, movement, and sights, and the manner that animations are drawn makes us curious about what will happen next.
- **They're fun-** People learn better when they are in a good mood, and whiteboard animations make the material more enjoyable rather than just another piece of homework.

7. GSAP(GREENSOCK ANIMATION PLATFORM)

GSAP is a javascript library, designers with little experience in javaScript may produce interesting SVG animations. The fact that GSAP is portable and simple to use is its strongest feature.[22]

Before starting to write the code, the GSAP library needs to be added to the HTML file, get the CDN link and insert it in a script tag. CDN stands for content delivery network , it means that hosting javascript files on your network some outside sources will host them for you.
GSAP offers many free features such as TweenLite and TimelineLite that allows it to manipulate SVG very efficiently.
Tweens are fundamental animation functions; to animate any object, we first call it and specify all of its characteristics. TweenLite informs JavaScript that we need to use GSAP to animate things.
Timelines are nothing more than a collection of animations that play out sequentially or simultaneously.

On SVG elements, we can apply any transform operation that we apply to DOM elements.
We can generate drag and drop effects on svg elements using GSAP draggable.

DrawSVGPlugin makes it easier to gradually show or hide an SVG's stroke. We can also use it to animate the beginning and end positions of strokes.

There are many features of GSAP that can help in creating rich SVG animations with minimal time and effort and with little or no knowledge of javaScript.

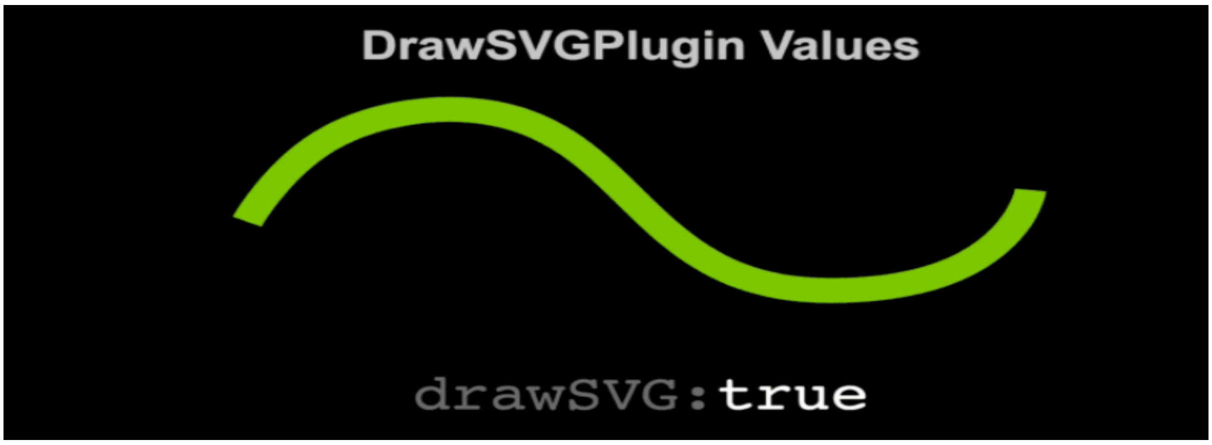


Fig. 7.1. Depicting GSAP animation[23]

8. CONCLUSION

In this work we have created interactive animation using GSAP,CSS and javaScript. Positioning of elements is one of the fundamental requirements of animation; the elements are positioned as per user requirement as it’s also a basic need for animating SVG elements.
We have attempted to automate the animation with the help of regex(Regular animation).
The Automation of SVG animation is very advantageous today and due to powerful combinations with GSAP, CSS or JavaScript it will remain relevant in future also.

BIBLIOGRAPHY

[1]Rajora, Harish. "The Complete CSS Animations Tutorial [With Examples]." *LambdaTest*, www.lambdatest.com, 8 July 2021, https://www.lambdatest.com/blog/css-animations-tutorial/

[2]JS. "CSS-Animations." *CSS-Animations*, javascript.info, 23 Apr. 2022,https://javascript.info/css-animations.

[3]"CSS Transforms Module Level 1." *CSS Transforms Module Level 1*, www.w3.org, 14 Feb. 2019, https://www.w3.org/TR/css-transforms-1/.

[4]"An Introduction to SVG Animation | Toptal." *Toptal Engineering Blog*, www.toptal.com, https://www.toptal.com/front-end/svg-animation-guide. Accessed 23 June 2022.

[5]Coyler, Chris. "Position | CSS-Tricks." *CSS-Tricks*, css-tricks.com, 20 Mar. 2012, https://css-tricks.com/almanac/properties/p/position/.

[6]MDN Web Docs. "- CSS: Cascading Style Sheets | MDN." - *CSS: Cascading Style Sheets | MDN*, developer.mozilla.org, 18 Mar. 2022, https://developer.mozilla.org/en-US/docs/Web/CSS/position_value.

[7]Warren, Lewis. "What Is an SVG File Used For and Why Developers Should Be Using Them | Delicious Brains." *Delicious Brains*, deliciousbrains.com, 19 Jan. 2021, https://deliciousbrains.com/svg-advantages-developers/.

[8]MDN Web Docs. "Adding Vector Graphics to the Web - Learn Web Development | MDN." *Adding Vector Graphics to the Web - Learn Web Development | MDN*, developer.mozilla.org, 2 May 2022, https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Adding_vector_graphics_to_the_Web.

[9]Yip, Thomas. "The Best Way to Embed SVG on HTML (2021)." *The Best Way to Embed SVG on HTML (2021)*, vecta.io, 25 May 2018, https://vecta.io/blog/best-way-to-embed-svg.

[10]MDN Web Docs. "Positions - SVG: Scalable Vector Graphics | MDN." *Positions - SVG: Scalable Vector Graphics | MDN*, developer.mozilla.org, 28 Jan. 2022, https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Positions.

[11]Mdn web docs. "Basic Shapes - SVG: Scalable Vector Graphics | MDN." *Basic Shapes - SVG: Scalable Vector Graphics | MDN*, developer.mozilla.org, 4 Feb. 2022, https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Basic_Shapes.

[12]Aspose. "SVG Path Data |Documentation." *Aspose Documentation*, docs.aspose.com, https://docs.aspose.com/svg/net/drawing-basics/svg-path-data/. Accessed 21 June 2022.

[13]W3C Editor's Draft. "SVG Paths." *SVG Paths*, svgwg.org, https://svgwg.org/specs/paths/. Accessed 21 June 2022.

[14]Mdn web docs. “Paths - SVG: Scalable Vector Graphics | MDN.” *Paths - SVG: Scalable Vector Graphics | MDN*, developer.mozilla.org, 8 June 2022, https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths.

[15]Rinaldi, Brian. “Understanding The SVG ViewBox And Viewport | Modern Web.” *Modern Web*, modernweb.com, 19 June 2014, https://modernweb.com/understanding-the-svg-viewbox-and-viewport/.

[16]Defar, Filip. “How to Approach SVG Animations: A CSS Tutorial | Toptal.” *Toptal Engineering Blog*, www.toptal.com, https://www.toptal.com/css/svg-animation-css-tutorial. Accessed 21 June 2022.

[17] Coyler, Chris. “Animation | CSS-Tricks.” *CSS-Tricks*, css-tricks.com, 20 Aug. 2011, https://css-tricks.com/almanac/properties/a/animation/.

[18]Codrops, Team, and Louis Hoebregts. “Animate Anything Along an SVG Path - Codrops.” *Codrops*, tympanus.net, 19 Jan. 2022, https://tympanus.net/codrops/2022/01/19/animate-anything-along-an-svg-path/

[19]Computer Hope. “What Is a Regex (Regular Expression)?” *What Is a Regex (Regular Expression)?*, www.computerhope.com, https://www.computerhope.com/jargon/r/regex.htm. Accessed 21 June 2022.

[20]Programiz. “JavaScript Regex.” *JavaScript Regex*, www.programiz.com, https://www.programiz.com/javascript/regex.

[21]“Whiteboard Animation - Wikipedia.” *Whiteboard Animation - Wikipedia*, en.wikipedia.org, 14 Mar. 2022,https://en.wikipedia.org/wiki/Whiteboard_animation

[22]FreeCodeCamp(A). “The Beginner’s Guide to the GreenSock Animation Platform.” *freeCodeCamp.Org*, www.freecodecamp.org, 20 Feb. 2018, https://www.freecodecamp.org/news/the-beginners-guide-to-the-greensock-animation-platform-7dc9fd9eb826/.

[23]GreenSock. “DrawSVGPlugin - Plugins - GreenSock.” *GreenSock*, greensock.com, 3 Dec. 2014, https://greensock.com/drawsvg/.