

## Vector Graphics:

=====

### Introduction:

-----

HTML5 is evolving to provide better and, standard, richer graphics to help customer experiences. Creates an opportunity for web developers to use standards-based web technologies to create graphically rich interactive sites and applications without having to use specialized technologies or write browser-specific code.

### Vector graphics:

-----

Vector graphics are not a new concept. They are geometric primitives(shapes, points, lines, polygons) that are all based on vectors to represent images. Vector graphics range in complexity from simple to moderate to extremely complex. The following are some basic examples.

Simple - Callout in a document or Illustration

Moderate - Illustrations such as charts, maps and diagrams.

Complex - Documents such as those used for engineering.

Though the above ones are static in nature, vector graphics supports interactivity, a key feature that expands the scenarios significantly. Vector graphics provide for interactive and static for applications on the web, on the desktop, and on the devices.

### HTML5 Graphic Technologies:

-----

Using HTML5, the developer or designer can now create the previous experience using the standards-based technology. This greatly enhances the user experience by eliminating installation of plug-ins, which the installation is lined to 50% site abandonment. Currently graphics are delivered natively by the browser, and in the case of Internet explorer9, leverage the power of Microsoft Windows and hardware accelerated graphics. Next section provides an overview of two new technologies, but two different technologies, how to use them, and individual benefits and limitations of each

----- <canvas>  
<svg>

Almost any vector can be drawn by using either of these technologies, but sometimes there is significantly more work done by the developer or the computer depending upon the task.

### ### SVG:-

Scalable Vector Graphic is used to describe a retained mode of graphics model that persists in an in-memory model that can be manipulated through code results in re-rendering. This is different from immediate mode, which is discussed later. Both are provided for in HTML5.

Similar to HTML5, SVG is built into the document using elements, attributes, and styles. When the <svg> element is first introduced into the document, it behaves much like a <div> and is part of HTMLDocument, but includes additional interface SVGDocument (SVGDocument provides deeper and richer interaction with the vector graphics)

Features of SVG -

- Styling
- Programmability
- Interactivity

### ### Canvas:-

Another approach to provide richer graphical experience for users provided by the <canvas> tag, introduced in HTML5 by Apple for Safari, and other graphical widgets. It exposes a more programmatic experience for drawing immediate mode graphics including the rectangle, path and images, similar to SVG. Immediate mode graphic rendering is a "fire and forget" model that renders graphics directly to the screen and then subsequently has no context as to what was done. In contrast to retained mode, the rendered graphics are not saved; a developer needs to re-invoke all drawing commands required to describe the entire scene each time a new frame is required, regardless of actual changes ( SVG is otherwise known to have a "scene graph")

Features of Canvas:-

- The element
- Programmability of events

### High Level Summary of Canvas & SVG::

-----	
Canvas ::	SVG::
* Pixed based (dynamic .png)	* Shape based
* Single HTML elements elements, which become	* Multiple graphical part of the
DOM	

\* Modified through script only

\* Modified through script and CSS.

\* Event Model/user interaction is granular

\* Event model/user interaction is abstracted

\* Performance is better with small surface/large number of objects

\*

Performance is better with smaller number

of objects (<10K) and a larger surface, or both