Assignment 4

1) How to automatically set the height of div to take the height of parent?

Ans: To set the height of div to take the height of parent, we can use

* display: block and height: auto
* display: inline-block and height: auto
* display: flex
* display: table

2) What is the difference in using px, em, pt, vh, vw? Which is the best

measuring unit to use? Explain.

Ans**: Pixel:** Before the advent of responsive design, pixel was widely used as

THE font sizing for everything from typography to width, padding

and height. However, pixels are fixed-size units and do not change

based on the size of the viewport or are scalable. If you want to all

build a responsive website using pixels, it can get very difficult to

keep track of the pixel-based values throughout the CSS style we

sheet, including the values within media queries. So, What font

property can use that will allow us to make minimal changes to

our CSS stylesheet but still be scalable? That’s where em, rem,

percent and viewport units come into the picture.

**EM:** Em is a flexible, scalable unit that is converted by the browser into

pixel values. If the default font size in chrome is 16 px, 1 em = 16 pixel.

One big misconception about em is that it is relative to the font size

of the parent element. As per the W3 spec, em is relative to the font

size of the element on which they are used.

**VW/VH:** Viewport width and height properties are relative to

the width/height of the current viewport size.

1vw = 1% of viewport width  
 1vh = 1% of viewport height

Viewport here means the browser’s window size. If you want to

size your element based on the viewport width and not the parent

need to element/root element, then viewport units are the font sizing

unit you use.

**PT:** The only place where you could use **pt** for setting a font size is in style

sheets for print, if you need to be sure the printed font is exactly a certain

size.

3) How to draw geometrical shapes using CSS and SVG?

Ans: **SVG:** SVG, which stands for *Scalable Vector Graphics*, is an XML-based

vector image format for the Web. Unlike GIF, PNG and JPEG image file

formats, SVG is scalable, which means that no matter how you scale

or enlarge the image file, the quality will still look good.

**Why we use SVG:**

* Can be created and edited with any text editor
* Can be printed with high quality resolution
* Can be used for animation
* Is a W3C recommendation
* Works with other W3C standards like DOM
* Looks great for retina displays
* Scales to any size without looking stretch out

SVG can be easily embedded into HTML using SVG tags. SVG has its own tags. SVG must have a width and height so as to contain its element. The following are the elements can be used to draw inside its canvas.

* Circle
* Rectangle
* Ellipse
* Line
* Polyline
* Polygon
* Path
* Text

**CSS:** CSS is capable of making all sorts of shapes. Squares and a

rectangles are easy, as they are the natural shapes of the web.

Add width and height and you have the exact size rectangle

you need. Add border-radius and you can round that shape,

and enough of it you can turn those rectangles into circles and ovals.