Assignment 4

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

A parent div with height: auto. In this div, there are two other divs. One with an icon, one with some text in it. These divs both have height: auto as well so the text mainly decides how large the parent div becomes. Would now like to vertically center the icon next to the text, but the problem is that the div containing the icon isn’t as high as the parent div.

1. What is the difference in using px, em, pt, vh, vw ? Which is the best measuring unit to use ? Explain.

* pt
* CSS inherited the units pt (point).
* CSS there is no reason to use pt, use whichever unit you prefer. But there is a good reason to use neither pt nor any other absolute unit.
* The only place where could use pt (or cm or in) for setting a font size is in style sheets for print, if need to be sure the printed font is exactly a certain size.
* px
* The px, is a often a good unit to use, especially if the style requires alignment of text to images.
* If a property accepts a value in px (margin: 5px) it also accepts a value in inches or centimeters (margin: 1.2in; margin: 0.5cm) and vice-versa.
* em
* The em is simply the font size. In an element with a 2in font, 1em thus means 2in. Expressing sizes, such as margins and paddings, in em means they are related to the font size, and if the user has a big font (e.g., on a big screen) or a small font.
* vh
* The vh is 1/100th of the window's height.
* The vh unit is the relative unit which is 1% of the height of the viewport (size of the browser window).
* In simple terms, it is 1/100th of the height of the viewport. vh is not supported in some old browsers.
* vw
* The vw unit is the relative unit which is 1% of the width of the viewport and the viewport is the size of the browser window. It differs from browsers to browsers.
* In simple terms, it is 1/100th of the width of the viewport.
* This means that vw unit value will keep on changing once you keep changing the width of the browser window.

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

* CSS

HTML and CSS is a rectangle. Every element is governed by a rectangular box model. Images and text are all rectangular and the text flows in rectangular areas.

CSS is capable of making all sorts of shapes. Squares and rectangles are easy, as they are the natural shapes of the web. Add a width and height and 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. Using properties like box-shadow, border-radius and other border properties with a few HTML elements or pseudo-elements, can render different geometric shapes on a page.

* SVG

Using properties like box-shadow, border-radius and other border properties with a few HTML elements or pseudo-elements, can render different geometric shapes on a page.

* SVG shapes are real content (actual elements drawn on screen, instead of empty divs with no content, which leads to..).
* SVG shapes are accessible (by screen readers).
* SVG shapes are semantic (element names and attributes).
* SVG shapes can be edited using graphics editors, while CSS shapes can’t.
* SVG shapes have powerful attributes that give you finer control over the end result (stroke, fill, stroke-width, etc.).