Student: Jorge Alvarez Grana (#249179033)
Class: WDD330
Term: Fall 2021
Subject: W08 Readings

Ch8: Transforms and Transitions

This chapter was rather interesting because giving some transformations and transitions might well be a form of giving a website a more professional touch. Of course, we have to be aware that all of these won't get in the middle of the user experience, given the fact that I read the other day that users won't last longer than 4 seconds to find the information they are looking for. Even if a website takes more than 5 seconds to load, 90% of users will leave.[1]

Now I understand better what are those *-webkit-transforms* I used to see at times when checking code in the Internet. As the text said "Transforms require vendor preUxing for IE9, Android up to 4.4.3, iOS8, and Blackberry 10." Trying to play a bit with some code I made a moving header-text to make the effect of loading the website when getting into a URL (here showing just the left side, but there was a right move as well to give some buncing effect):

```
section header h1 {
  color: #fff;
  text-transform: uppercase;
  font-size: 60px;
  font-weight: 400;
  letter-spacing: 35px;
  -webkit-animation-name: LeftMove;
          animation-name: LeftMove;
  -webkit-animation-duration: 5s;
          animation-duration: 5s;
  /*animation-delay: 1s;*/
  -webkit-animation-timing-function: ease-out;
          animation-timing-function: ease-out;
}
```

```
@keyframes LeftMove {
  0% {
    opacity: 0;
    -webkit-transform: translateX(-100px);
            transform: translateX(-100px);
  }
  80% {
    -webkit-transform: translateX(10px);
            transform: translateX(10px);
  }
  100% {
    opacity: 1;
    -webkit-transform: translate(0);
            transform: translate(0);
  }
}
```

Rotation effects where rather funny although I guess we should be careful using them because if not the site might be seen as not very serious. And the translate worked quite similiar (in syntax) as transform above mentioned. But it all seemed very impresive that those things could be done using just CSS and not needing JavaScript or any other programming language.

---

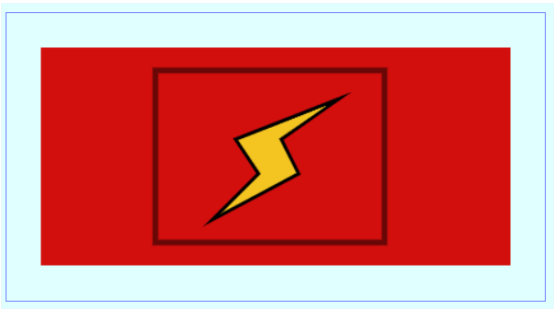
1 How fast should a website load in 2021? https://www.hobo-web.co.uk/your-website-design-should-load-in-4-seconds/

Ch12: Canvas, SVG, and Drag and Drop

I've never heard about Canvas, so I gave it a try and using the following code in JavaScript I got the following figure:

```
const canvas = document.querySelector("#canvas");
const ctx = canvas.getContext("2d");
ctx.fillStyle= "#d30f0e"
ctx.fillRect(30, 30, 410, 190)
ctx.strokeStyle = 'rgba(0,0,0,0.5)'
ctx.lineWidth = 5
ctx.strokeRect(130, 50, 200, 150)

ctx.strokeStyle = 'black'
ctx.lineWidth = 3
ctx.lineTo(200, 110)
ctx.lineTo(220, 140)
ctx.lineTo(180, 180)
ctx.lineTo(255, 140)
ctx.lineTo(240, 110)
ctx.lineTo(290, 75)
ctx.closePath()
ctx.fillStyle = "#f4c521"
ctx.fill()
ctx.stroke()
```



Concerning SVG I was surprised that it doesn't pixelate so images and graphs will always have a great display in the website. I had some exposure before to this format so I tried more Canvas that was the new item here. On which to use? Well, I assume that both have their place and the good thing on compiting technologies is that they will give designers some more choice in case one goes wrong or doesn't get to the point we are aiming. I guess SVG will be more helpful because you can use it in software like Illustrator that is a software all designers talk much about. I guess that in a real life project programmers would be doing one thing and designers will do another. I don't see designers using JS but I can see some sort of middle-ground cooperation between the two where many a little bit of help from coding Canvas can help us both arrive to the desired end.