WDOS - Tutorial – Web Media Tutorial

Objectives: Getting to know on handling media in the web

Instructions

- You are required to style the given web page.
- Download the given **web media tute** folder in LMS, unzip and open it in VS code.
- Add styles to the respective style sheets.
- Refresh the browser and check each time you apply a new style to the web page.
- Make sure you understand each part, but if you are unsure, ask your tutor.

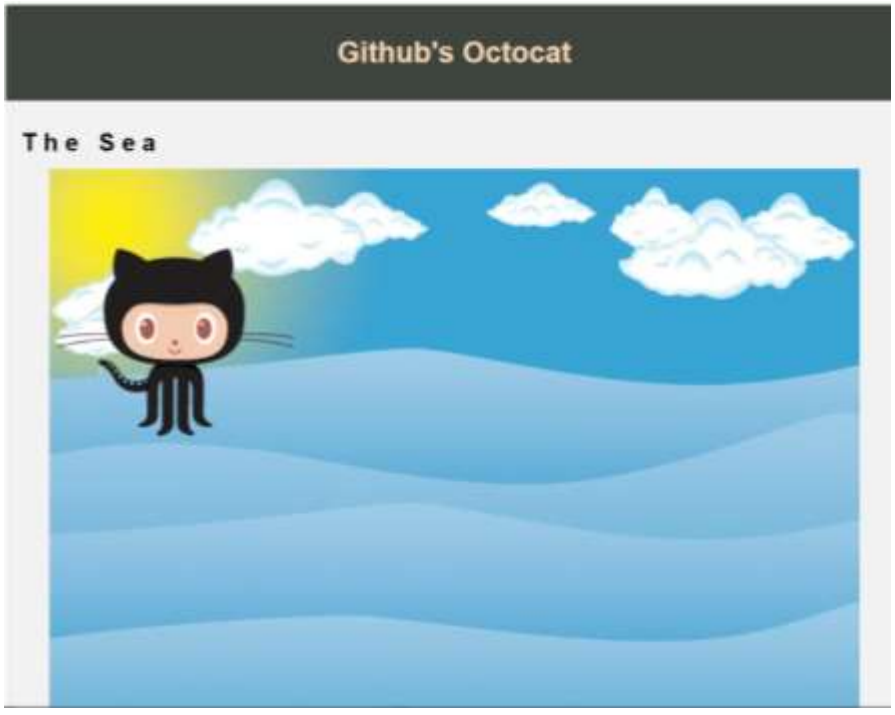
Task 1 – Picture Tag

1. Take the dog image and save it for art direction for desktop, tablet and mobile
2. Create a new web page with a picture tag on it with
    a. Different images (webP) for mobile, tablet and desktop
    b. Different srcset values for 1x, 2x and 3x resolution
    c. A fall back image for browsers that don't support webP or picture tag

Refer the lecture slides and the given links in the lecture.

Task 2 – Transformations

1. Your animation page should look something like this



First, we will trigger transformations using the :hover so that we don't have to add any JavaScript. Use the animation.css we discussed yesterday and try the following.

2. Add in the code to the `#animal:hover`

```
transform: translate(50px,100px);
-webkit-transform: translate(50px,100px); /* older webkit browsers, some
mobile browsers*/
```

3. Test by hovering over the cat
4. Change the transformation to the following one by one, testing for each one (comment out each one as you do it as you will need this again in a later step).
    a. translateX(50px)
    b. translateY(50px)
    c. Move across 250px and down 60px
    d. rotate(30deg);
    e. rotate by 270 degrees
    f. scale(2,2);
    g. scale by half in both directions
    h. scale in the x direction by 2 and the y direction by 3
    i. scale in the x direction by 4 and the y by half.
    j. skew(50deg,-20deg);
    k. skew by the x direction by 20 degrees
    l. skew in the y direction by 60 degrees
    m. matrix(1, 0, 0, 1, 0, 0); - this should show the same - default
    n. matrix(1, .25, 0, 1, 0, 0); - this should skew the object
    o. matrix(1, .25, -.25, 1, 0, 0); - this should rotate the object
    p. matrix(1, .25, -.25, 1, 50, 50); - This should apply a translate (x and y) by the last two numbers
    q. matrix(2, .25, -.25, 2, 50, 50); - The 1st and 4th digit is to zoom. This should zoom the object
    r. Create a matrix that makes the object 3 times bigger in each direction, rotates by a quarter rotation, and moves it by 50px in both x and y directions.
5. Change the origin to `transform-origin:0% 0%;`
6. Test each of the transformations in step 7 again and note the difference

Task 3 – Transitions

1. Comment out any transformations you have so far
2. Add the following

```
#animal:hover {
opacity:0;
}
```

3. Test the site. You should notice the Octocat disappears like a light switch. It has two states – there and not there
4. Now change the style of the animal (initial state) to the following

```
#animal {
padding-top:80px;
height:192px;
width: 250px;
background:url(images/octocat.png) no-repeat center center;
transition-property:opacity;
transition-duration:2s;
transition-delay:1s;
}
```

5. Save and test the page and you should see a one second delay and then a fade over two seconds
6. Change the transition lines to the DRY short-code version

```
transition:opacity 2s ease-in 1s;
```

7. Test the page again and see the difference.
8. Remove the 1s delay and change the timing function to each of these in turn (you can do this in the dev tools in F12 in Chrome). Try each one and note the difference
    a. ease
    b. linear
    c. ease-in
    d. ease-out
    e. ease-in-out
    f. cubic-bezier(.12,.95,.2,.14)
9. Take a look at http://cubic-bezier.com/ and try out some other curves

Task 4 – Simple Animations

1. Comment out any styles you have on animal and animal hover
2. Add the following code to animate Octocat

```
#animal {
height:192px;
width: 250px;
background:url(images/octocat.png) no-repeat center center;
animation-name: squarePath;
animation-duration: 2s;
-webkit-animation-name: squarePath;
-webkit-animation-duration: 2s;
}

 @keyframes squarePath {
    0%, 100% {
      transform: translate(0, 0);
    }
    25% {
      transform: translate(200px, 0);
    }
    50% {
      transform: translate(200px, 200px);
    }
    75% {
      transform: translate(0, 200px);
    }
  }

@-webkit-keyframes squarePath {
    0%, 100% {
      -webkit-transform: translate(0, 0);
    }
    25% {
       -webkit-transform: translate(200px, 0);
    }
    50% {
       -webkit-transform: translate(200px, 200px);
    }
    75% {
       -webkit-transform: translate(0, 200px);
    }
  }
```

3. Test the page in chrome, firefox and edge.
4. Change the animation using percentages to move octocat in a different shape with 6 points in the keyframe – call it myAnimation
5. Save and show the lecturer the result

6. Change the animation properties to the following lines

```
animation: myAnimation 5s linear 0.5s infinite alternate;
-webkit-animation: myAnimation 5s linear 0.5s infinite alternate;
```

7. Test the page again and see what each of the properties do
8. Change the delay so that there is a 2 second delay at the beginning
9. Change the iteration so it just runs 4 times only
10. Test the animation in edge, firefox and chrome and show the final version to the lecturer