Ch 8 CSS3 transforms and transitions

* Transform – lets you translate, rotate, scale, and/or skew any element on page
* Transform functions – can manipulate an element’s appearance
* Translation – allows you to move elements left, right, up, or down
  + Similar to position: relative;
  + Translate(x,y) – move left and move up
  + Transform: translateY(); moves up only
  + Transform: translateX(); moves y only
  + Doesn’t work on inline elements
* Scale(x,y) function – scales element by defined factors horizontally then vertically
* Rotate() function – rotates element around point of origin transform: rotate(10deg);
* Skew(x,y) function – skew along x and y axes both parameters included unless you want it to only occur on x axis
* Transform-origin property – transform-origin: 0 0;
* Transform, depending on the order you take the result will be different. translate and rotate will result in two different outcomes
* Transitions allow the values of CSS properties to change over time, simple animations
* Transition property – first declare original state of element
  + Second – declare final state of transition, ex: hover
  + Third – include transition function in your default style declaration using transition properties
* Animation properties – need 2 declarations for each property
  + Animation-name: appearDisappear;
  + Animation duration: 300ms;
* Animation-timing-function property – determines how the animation will progress over its duration
* There is shorthand with all animation properties
  + .verbose {
  + animation-name: appearDisappear;
  + animation-duration: 300ms;
  + animation-timing-function: ease-in;
  + animation-iteration-count: 1;
  + animation-direction: alternate;
  + animation-delay: 5s;
  + animation-fill-mode: backwards;
  + animation-play-state: running;
  + }
  + /\* shorthand \*/
  + .concise {
  + animation: 300ms ease-in alternate 5s backwards appearDisappear;
  + }

Ch 12 Canvas, SVG, and Drag and Drop

* Canvas API, we can draw anything we can imagine
  + Gives power to manipulate pixels in images and even video
* Add canvas element page
  + <canvas id="myCanvas" class="myCanvas" width="200" height="200"></canvas>
  + Can be used in css as well
    - .myCanvas { border: dotted 2px black;}
* all of these things can be done in the Canvas JavaScript API
  + canvas.html
  + getContext method – passing “2d” string, 2d drawing
    - 2d context is the only wide support
* Combining WebGL and new API – you can draw in 3d
* Just like in real life, you must fill your brush with color
  + strokeStyle, fillStyle, CanvasGradient, CanvasPattern
  + var canvas = document.getElementById("myCanvas");
  + var context = canvas.getContext("2d");
  + context.strokeStyle = "red";
  + context.fillStyle = "blue";
    - can use # colors, and rgb()
* drawing a rectangle
  + var canvas = document.getElementById("myCanvas");
  + var context = canvas.getContext("2d");
  + context.strokeStyle = "red";
  + context.fillStyle = "rgba(0, 0, 255, 0.5)";
  + context.fillRect(10, 10, 100, 100);
  + context.strokeRect(10, 10, 100, 100);
* adding image to your function
  + function drawPattern() {
  + …
  + var img = new Image();
  + img.src = "../images/bg-bike.png";
  + img.onload = function() {
  + var pattern = context.createPattern(img, "repeat");
  + context.fillStyle = pattern;
  + context.fillRect(10, 10, 100, 100);
  + context.strokeRect(10, 10, 100, 100);
  + };
  + }
* Can draw any shape you can imagine
* Saving Canvas drawings – toDataURL
  + function saveDrawing() {
  + var canvas5 = document.getElementById("demo5");
  + window.open(canvas5.toDataURL("image/png"));
  + }
* SVG – Scalable Vector Graphics
  + Allows you to describe vector graphics using XML
* Drawing in SVG
  + <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 400 400">
  + <circle cx="50" cy="50" r="25" fill="red"/>
  + </svg>
* Drag and drop
* API – allows us to specify that certain elements are draggable, then specify what should happen when these draggable elements are dragged over or dropped onto other elements on the page
  + Draggable – attribute used on any HTML
  + Event listener – dragstart on any draggable HTML elements
  + Event listener – dragover and drop events on any elements that you want to have accept dropped items
  + Draggable=”true”
    - Draggable is not a Boolean – you HAVE to use true only