[Lesson 13]

Roi Yehoshua 2018

[DAN.IT]
EDUCATION

What we learnt last time?

- CSS Grid Layout

[Our targets for today]

- How to change property values smoothly (from one value to another), with a given duration
- How to gradually change element’s style using animation
- How to create repeated animation
- How to create complex animation with multiple objects
- How to create simple parallax effect with pure CSS
- Make 2 blocks with parallax on our landing page
- Create parallax with different speed of background layers

[Transition]

- CSS transitions allow you to change property values smoothly (from one value to another), over a given duration.
- To create a transition effect, you must specify two things:
  - the CSS property you want to add an effect to
  - the duration of the effect
- example - transition: width 2s;
- to change several property values smoothly use comma:

  transition: width 2s, height 4s;

[Transition]

→ The transition property is a shorthand of four CSS properties: transition-property, transition-duration, transition-timing-function, transition-delay:

```
.selector {
    transition: property duration transition-timing-function delay;
}
```

[Transition]

- The transition property is a shorthand of four CSS properties: transition-property, transition-duration, transition-timing-function, transition-delay:
  - transition-property refers to the CSS property you wish to transition. It is required in the transition shorthand.
  - transition-duration refers to the duration of the transition. This value is written in seconds with the s suffix (like 3s). Also required.
  - transition-delay refers to how long you want to wait before starting the transition. This value is written in seconds with the s suffix (e.g. 3s). Optional.
  - the transition-timing-function governs how a transition occurs

[transition-timing-function - part 1]

- All transitions have a value of linear for transition-timing-function by default, which means the property changes evenly until the end of the transition. Other options:
  - Imagine yourself throwing a tennis ball into an open field. The ball leaves your hand with the maximum speed. As it moves, it loses energy, it decelerates and eventually comes to a halt. This is called ease-out.

[transition-timing-function - part 2]

- Now imagine you're in a car. It's not moving right now. When you move the car, it accelerates and goes toward its top speed. This is called ease-in.
- ease-in-out - specifies a transition effect with a slow start and end
- cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

[Animation]

- animation lets an element gradually change from one style to another
- You can change as many CSS properties as you want, as many times you want.
- To use CSS animation, you must first specify some keyframes for the animation.
- keyframes hold what styles the element will have at certain times.

Keyframes

- When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times

```css
@keyframes example {
   from {
       background-color: red;
   }
   to {
       background-color: yellow;
   }
}
```

[Keyframes]

→ To get an animation to work, you must bind the animation to an element

```css
div {
   width: 100px;
   height: 100px;
   background-color: red;
   animation-name: example;
   animation-duration: 4s;
}
```

Delay an Animation

→ The animation-delay property specifies a delay for the start of an animation

→ The following example has a 2 seconds delay before starting the animation:

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-delay: 2s;
}
```

[Perspective - CCS3 property]

- perspective - Give a 3D-positioned element some perspective
- The perspective property defines how far the object is away from the user
- Lower value will result in a more intensive 3D effect than a higher value
- When you define the perspective property for an element, it is the CHILD elements that get the perspective view, NOT the element itself
- Default value of perspective is none;
- Syntax: perspective: length | none;

[Transform - CCS3 property]

- transform - change element size, form and position
- transform does not change or replace other elements. Other elements do not move with respect to it
- You can transform element with: display: block | inline-block | table-row | table-row-group | table-header-group | table-footer-group | table-cell | table-caption
- There are two types of transformations: 2D and 3D

[3D - CCS3 property]

- There are two important properties for 3D transformation: translateZ and scale
- translateZ is a CSS function that repositions an element along the z-axis in 3D space, i.e., closer to or farther away from the viewer
  - example: transform: perspective(500px) translateZ(200px);
- The scale() CSS function defines a transformation that resizes an element on the 3D plane. Because the amount of scaling is defined by a vector, it can resize the horizontal and vertical dimensions at different scales.
- example: transform: translateZ(-2px) scale(3);

[Control questions]

1. transition is a shorthand for which 4 properties?
2. How to add transition effect to all changed CSS properties?
3. How does cubic-bezier value for transition-timing-function work?
4. What is the default value for animation-iteration-count?
5. How does animation-fill-mode: backwards work?
6. How will the following animation work?

```
 0%   { motion-offset: 0; }
100% { motion-offset: 100%; }
```

[Control questions]

7. What is parallax effect?
8. How to create simple parallax effect with pure CSS?
9. How does “perspective” CSS property work?
10. How to change depth of parallax effect?
11. How to change speed of parallax effect during scrolling?