

[Lesson 12]

Roi Yehoshua 2018

[What we learnt last time?]

Flexbox:

- How to create multi-column design without necessity to clear the flow
- How to create columns of same height
- How to center elements in the parent block horizontally and vertically
- How to make responsive and fixed width columns

[What we learnt last time?]

- 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

[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

[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

```
@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

```
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;  
}
```

[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;  
}
```

[Control questions]

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

0% { **motion-offset: 0;** }

100% { **motion-offset: 100%;** }

[Materials]

Core materials:

<https://stfalcon.com/ru/blog/post/animation-css>

<https://github.com/FrontenderMagazine/ochen-prostoe-rukovodstvo-po-css-animatsiyam/blob/master/rus.md>

Video materials:

<https://www.youtube.com/watch?v=5Yzkz9xknQU>

<https://www.youtube.com/watch?v=sQsuDxVPmj8>

<https://www.youtube.com/watch?v=ql8e48XkZUI>

[Homework]

Homework 10 Star Wars Animation

[https://gitlab.com/dan-it/groups/fs6/tree/master/src/main/frontend/html_css/homework/home work10](https://gitlab.com/dan-it/groups/fs6/tree/master/src/main/frontend/html_css/homework/home%20work10)

Deadline for task fulfillment: 12.08.2018