

# Effektlar va animatsiya

---

# O'tish effektlari (transitions)

---

O'tish effektlarini ishlatish:

- qaysi CSS xossa uchun o'tish effekti ta'sir etadi (**transition-property**) (*majburiy*)
- o'tish effektining vaqti (**transition-duration**) (*majburiy*)
- o'tish effektining tezlashish usuli (**transition-timing-function**)
- o'tish effekti boshlanishidan oldin kutish vaqti (**transition-delay**)

O'tish effektlarida boshlang'ich holat va effekt tugashi holati berilishi majburiy.

Boshlang'ich holat – brauzerda yuklangandan keyingi holati.

Tugash holati **:hover**, **:focus**, **:active** kabi holatlar bilan aniqlanadi.

# O'tish effektlari

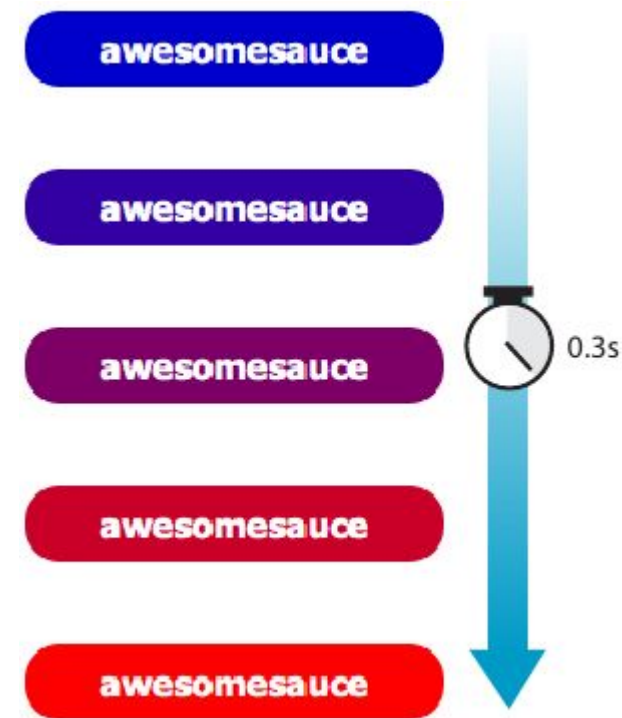
**transition-property** – qaysi xossaga o'tish effekti ta'sir ettirilishi

Qiymatlari: *xossa nomi* | all | none

**transition-duration** – o'tish effekti vaqti

Qiymatlari: *vaqt (sekundda (s) yoki millisekundda (ms))*

```
.smooth {  
  display: block;  
  text-decoration: none;  
  text-align: center;  
  padding: 1em 2em;  
  width: 10em;  
  border-radius: 1.5em;  
  color: #fff;  
  background-color: mediumblue;  
  transition-property: background-color;  
  transition-duration: 0.3s;  
}  
  
.smooth:hover, .smooth:focus {  
  background-color: red;  
}
```



# O'tish effektlari

---

## **Fonlar**

background-color

background-position

## **Chegaralar**

border-bottom-color

border-bottom-width

border-left-color

border-left-width

border-right-color

border-right-width

border-top-color

border-top-width

border-spacing

outline-color

outline-width

## **Rang va xiralik**

color

opacity

visibility

## **Shrift va matn**

font-size

# O'tish effektlari

---

**transition-timing-function** – o'tish effektining bajarilish vaqti funksiyalari

Qiymatlari: ease | linear | ease-in | ease-out | ease-in-out | step-start | step-end | steps | cubic-bezier(##,##,##,##)

ease – sekin-asta boshlanadi, tezlashadi va oxirida sekinlashadi.

linear – o'tishning boshidan oxirigacha bir xil tezlikda bo'ladi.

ease-in – sekin boshlanadi, keyin tezlashadi.

ease-out – tez boshlanadi, keyin sekinlashadi.

ease-in-out – sekin-asta boshlanadi, tezlashadi va oxirida yana sekinlashadi (boshlanishi ease dan sekinroq)

steps(X, start|end) – o'tish effektini X ta qadamga bo'ladi, start|end orqali o'tish effekti qadam holatiga o'rnatiladi.

step-start, step-end – mos ravishda steps(1, start) va steps(1, end) ga teng.

cubic-bezier(##,##,##,##) – o'tish effekti vaqti Bezye egri chizig'i orqali o'rnatiladi.

**transition-delay** – o'tish effekti boshlanishini kutib turish vaqti

Qiymatlari: time

cubic-bezier(.17,.67,.83,.67)

COPY

Preview & compare GO!

Duration: 1.5 seconds

Library IMPORT EXPORT

Click on a curve to compare it with the current one.

ease linear ease-in ease-out ease-in-out

Tip: Right click on any library curve and select "Copy Link Address" to get a permalink to it which you can share with others

SAVE TO LIBRARY

PROGRESSION

TIME

Made by Lea Verou with care

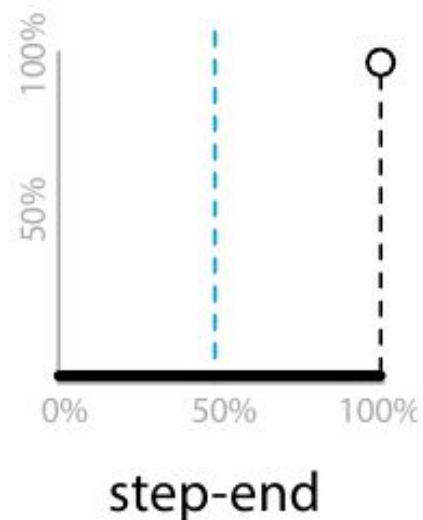
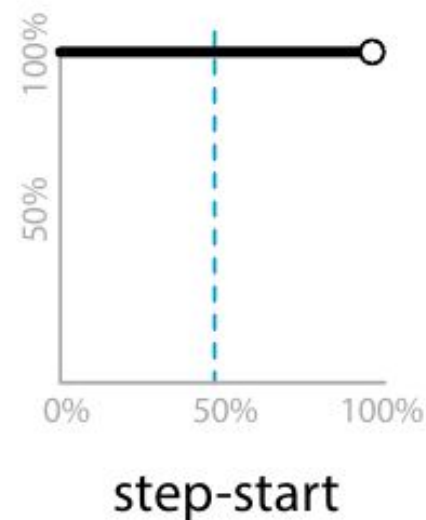
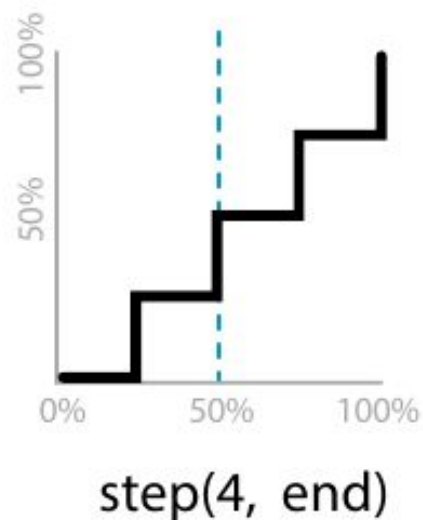
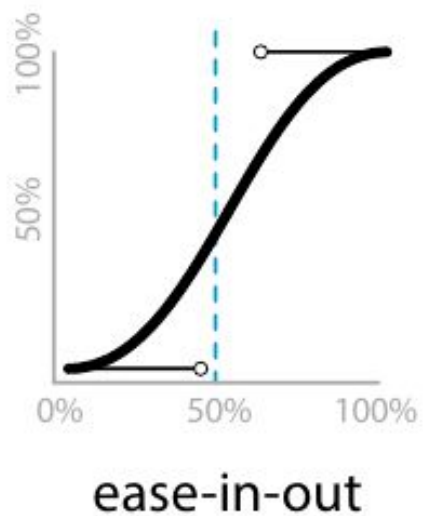
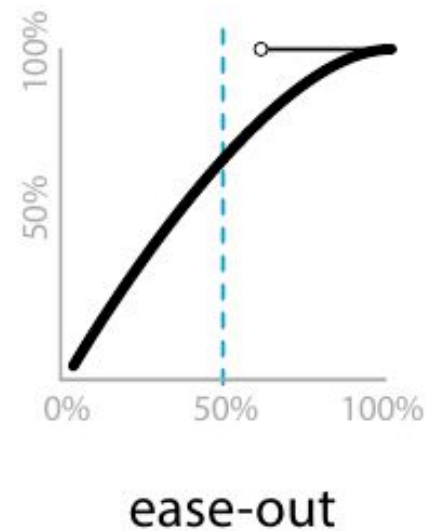
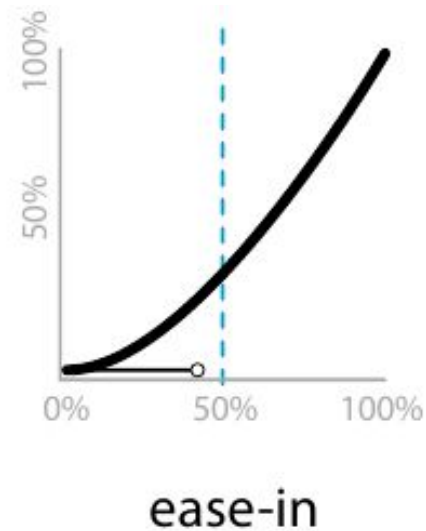
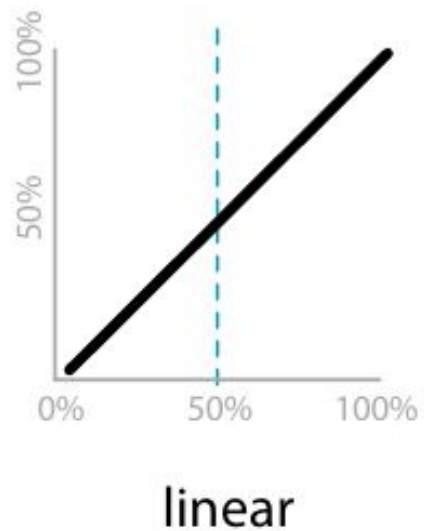
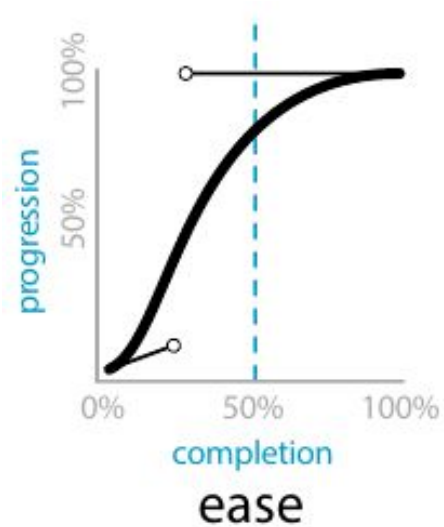
ABOUT

DONATE

Preview & compare GO!

Duration: 2.8 seconds





# Qisqa transition xossasi

---

transition: *property duration timing-function delay*;

```
.smooth {  
    ...  
    transition: background-color 0.3s ease-in-out 0.2s;  
}
```



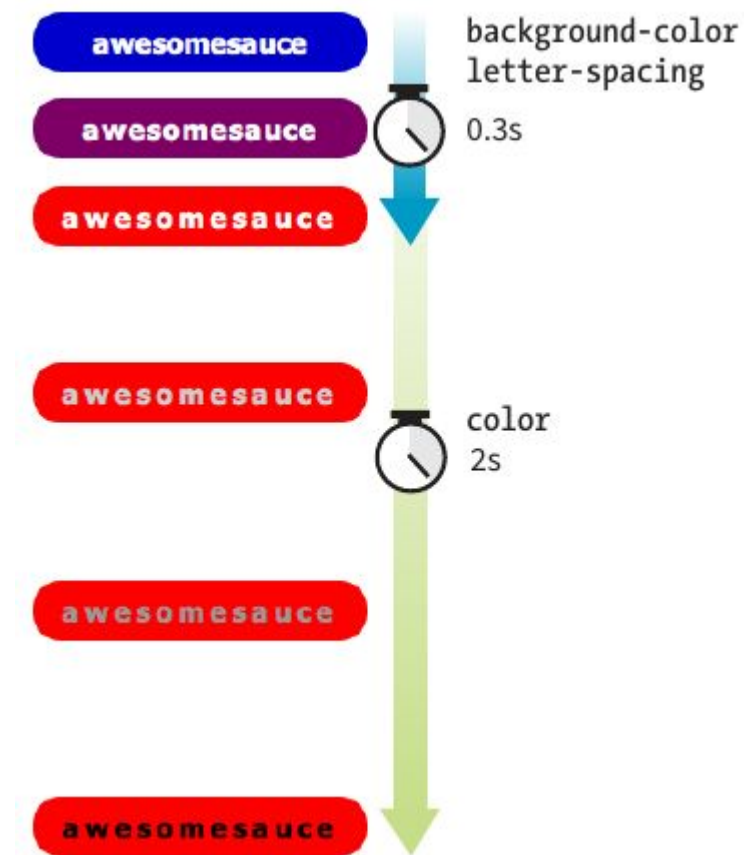
# Bir nechta o'tish effektini tadbiq qilish

```
.smooth {  
  ...  
  transition-property: background-color, color, letter-spacing;  
  transition-duration: 0.3s, 2s, 0.3s;  
  transition-timing-function: ease-out, ease-in, ease-out;  
}
```

```
.smooth:hover, .smooth:focus {  
  background-color: red;  
  letter-spacing: 3px;  
  color: black;  
}
```

Yoki

```
.smooth {  
  ...  
  transition: background-color 0.3s ease-out,  
              color 2s ease-in,  
              letter-spacing 0.3s ease-out;  
}
```

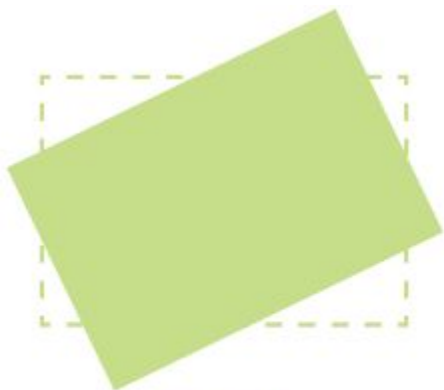


# Elementlarni o'zgartirish (transformatsiya)

---

**transform** (qatorli elementlarga va jadval ustunlariga qo'llanilmaydi)

Qiymatlari: `rotate()` | `rotateX()` | `rotateY()` | `rotateZ()` | `rotate3d()` | `translate()` | `translateX()` | `translateY()` | `scale()` | `scaleX()` | `scaleY()` | `skew()` | `skewX()` | `skewY()` | `none`



`rotate()`



`translate()`



`scale()`



`skew()`

# Elementni aylantirish

---

**rotate()** funksiyasi ishlatiladi, musbat (soat yo'nalishi bo'yicha) va manfiy (soat yo'nalishi teskari yo'nalish bo'yicha) burchak qiymatlarini qabul qilishi mumkin.

Odatda element markaziy nuqtasiga nisbatan aylantiriladi.

```
img {  
  width: 400px;  
  height: 300px;  
  transform: rotate(-10deg);  
}
```



transform: rotate(-10deg);

# Burchak qiymatlari

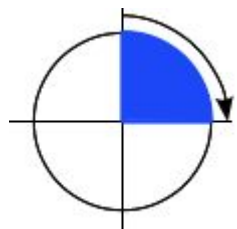
---

deg – burchakni darajada aniqlaydi (degree, bir to'liq aylana – 360 degree)

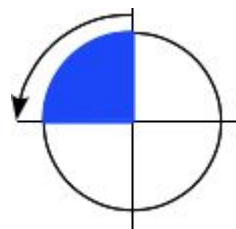
grad – burchakni gradianda aniqlaydi (gadian, bir to'liq aylana – 400 gradian)

rad – burchakni radianda aniqlaydi (gadian, bir to'liq aylana –  $2\pi$ ,  $1\text{rad} = 180/\pi$  degree)

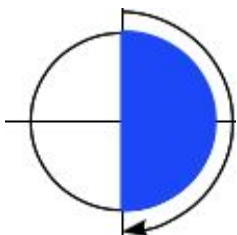
turn – burchakni burilishlar miqdorida aniqlaydi (bir to'liq aylana – 1 turn)



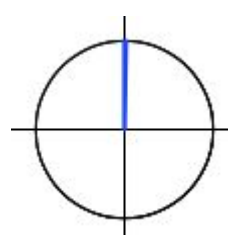
$$90\text{deg} = 100\text{grad} = 0.25\text{turn} \approx 1.5708\text{rad}$$



$$-90\text{deg} = -100\text{grad} = -0.25\text{turn} \approx -1.5708\text{rad}$$



$$180\text{deg} = 200\text{grad} = 0.5\text{turn} \approx 3.1416\text{rad}$$



$$0 = 0\text{deg} = 0\text{grad} = 0\text{turn} = 0\text{rad}$$

# Aylantirish markazini aniqlash

**transform-origin** (ikkita qiymat qabul qiladi)

Qiymatlari: percentage | length | left | center | right | top | bottom



`transform-origin: center top;`



`transform-origin: 100% 100%;`



`transform-origin: 400px 0;`

# Joylashuvini o'zgartirish

translate(), translateX(), translateY() funksiyalari ishlatiladi. translate() funksiyasida 1ta qiymat berilsa X o'qi bo'yicha ta'sir qiladi. Qiymatlari uzunlik o'lchovlari berilishi kerak.

```
transform: translateX(50px);  
transform: translateY(25px);  
transform: translate(50px, 25px); /* (translateX, translateY) */
```



transform: translate(90px, 60px);



transform: translate(-5%, -25%);



# Shkalani o'zgartirish

`scale()`, `scaleX()`, `scaleY()` funksiyalari orqali shkala o'zgartiriladi. `scale()` funksiyasida 1ta qiymat berilsa X va Y o'qlari bo'yicha ta'sir qiladi. Qiymatida shkala miqdori (haqiqiy sonda) yoki foizi beriladi.



`transform: scale(1.25);`



`transform: scale(.75);`



`transform: scale(1.5, .5);`

# Shkalani o'zgartirish

```
div {  
  border: 1px solid red;  
  width: 200px;  
  height: 200px;  
  overflow: hidden;  
  display: inline-block;  
  margin: 5px;  
}  
  
.sc1 {  
  width: 100%;  
}  
  
.sc2 {  
  width: 100%;  
  transform: scale(125%);  
}  
  
.sc3 {  
  width: 100%;  
  transform: scale(.5);  
}
```





# Elementni qiyshaytirish

`skew()`, `skewX()`, `skewY()` funksiyalari ishlatiladi. `skew()` funksiyasida 1ta qiymat berilsa X o'qi bo'yicha ta'sir qiladi. Qiymati sifatida burchak berilishi kerak.



```
transform: skewX(15deg);
```



```
transform: skewY(30deg);
```



```
transform: skew(15deg, 30deg);
```

# Bir nechta o'zgartirish effektini qo'llash

```
transform: function(value) function(value);
```

```
img:hover, img:focus {  
  transform: scale(1.5) rotate(-5deg) translate(50px, 30px);  
}
```

Normal state



`:hover, :focus`

rotate(), translate(), and scale() applied



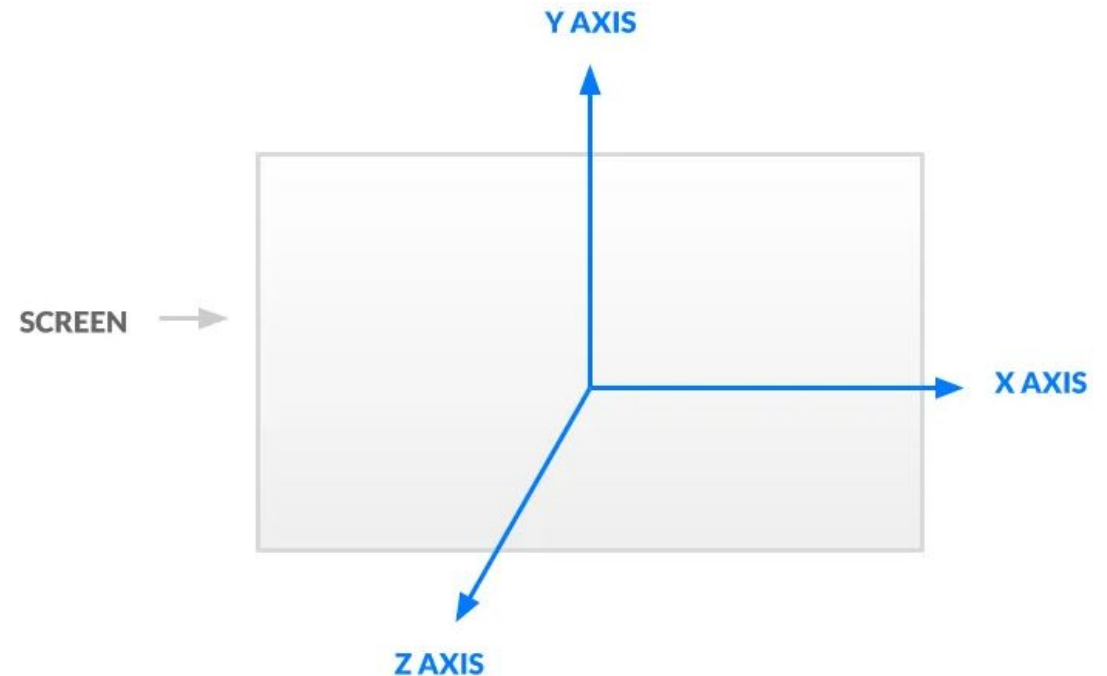
# 3D o'zgartirish effektlari

---

**perspective** – 3 o'lchovli soha uzunligini aniqlash

Qiymatlari: *uzunlik*

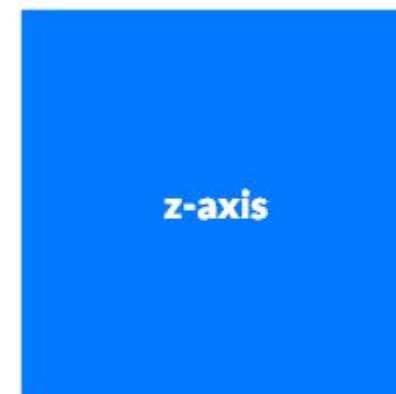
**transform** xossasining *translate3d*, *translateZ*, *scale3d*, *scaleZ*, *rotate3d*, *rotateX*, *rotateY*, *rotateZ* va *matrix3d* funksiyalari



# 3D o'zgartirish effektlari

---

transform



rotate



# 3D o'zgartirish effektlari

```
div {  
  border: 1px solid red;  
  width: 200px;  
  height: 200px;  
  overflow: hidden;  
  display: inline-block;  
  margin: 5px;  
  perspective: 100px;  
}  
  
.ex {  
  width: 100%;  
  transform: rotateX(45deg);  
}  
  
.ey {  
  width: 100%;  
  transform: rotateY(45deg);  
}  
  
.ez {  
  width: 100%;  
  transform: rotateZ(45deg);  
}
```



# Animatsiya

---

Animatsiya jarayoni 2 qismdan tashkil topgan:

1. @keyframes qoidasi orqali animatsiya qoidalarini o'rnatish
2. Animatsiya xossalari orqali animatsiyani elementga ulash

```
@keyframes animation-name {  
  keyframe { property: value; }  
  /* additional keyframes */  
}
```

Bu yerda: keyframe – 0-100% gacha bo'lgan animatsiya qismlari, shuningdek from (0%) va to (100%) kalit so'zlarini ham ishlatish mumkin.

# Animatsiya

```
@keyframes colors {  
  0% { background-color: red; }  
  20% { background-color: orange; }  
  40% { background-color: yellow; }  
  60% { background-color: green; }  
  80% { background-color: blue; }  
  100% { background-color: purple; }  
}  
  
@keyframe slide {  
  from { margin-left: 100% }  
  to { margin-left: 0%; }  
}
```



# Animatsiya xossalari

---

**animation-name** – animatsiya nomi (@keyframes qoidasi orqali aniqlanadi, majburiy)

**animation-duration** – animatsiya vaqti (majburiy)

**animation-timing-function** – animatsiya tezlashish funksiyasi (o'tish effektlari bilan bir xil)

**animation-delay** – animatsiyadan oldin kutish kerak bo'lgan vaqt

**animation-iteration-count** – animatsiya qaytarilish miqdori (*son* | infinite)

**animation-direction** – animatsiya yo'nalishi (normal, reverse, alternate, alternate-reverse)

**animation-fill-mode** – animatsiyadan oldin va keyingi elementning holati (none, forwards – animatsiya tugaganidan keyin oxirgi keyframe qoladi, backwards – animatsiya boshlanishidan oldin birinchi keyframe qo'llaniladi (delay vaqtida), both – ikkala holat ham qo'llaniladi)

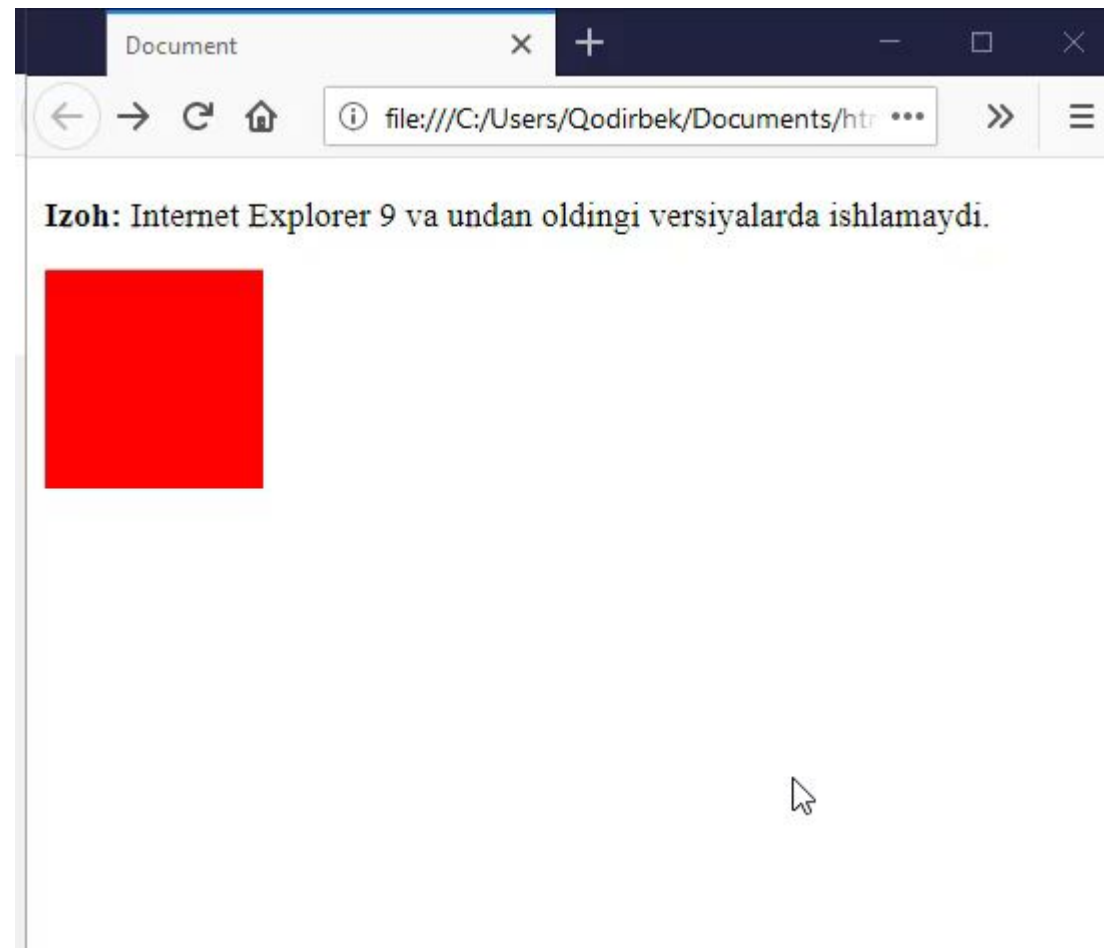
**animation-play-state** – animatsiyaning bajarilish holati (running – ishga tushgan holati, paused – pauza qilingan holati, JavaScript yordamida qayta ishlash uchun qulay)



# Animatsiya

```
div {  
  width: 100px;  
  height: 100px;  
  background-color: red;  
  position: relative;  
  animation-name: example;  
  animation-duration: 4s;  
  animation-iteration-count: 3;  
}
```

```
@keyframes example {  
  0% {  
    background-color: red;  
    left: 0px;  
    top: 0px;  
  }  
  25% {  
    background-color: yellow;  
    left: 200px;  
    top: 0px;  
  }  
  50% {  
    background-color: blue;  
    left: 200px;  
    top: 200px;  
  }  
}
```



```

body {
  display: flex;
}

.scene {
  width: 100px;
  height: 100px;
  margin: 2em;
  border: 3px solid #ccc;
  /* 3D effekt berish uchun */
  perspective: 100px;
}

.square {
  width: 100%;
  height: 100%;
  background-color: #0379ff;
  line-height: 100px;
  text-align: center;
  color: #FFF;
}

.square.x-axis {
  animation: x-axis-animation 1s
    ease-in-out 0s infinite alternate;
}

.square.y-axis {
  animation: y-axis-animation 1s
    ease-in-out 0s infinite alternate;
}

.square.z-axis {
  animation: z-axis-animation 1s
    ease-in-out 0s infinite alternate;
}

```



# Savollar?

---