Анимации.

План урока

1. Свойство transition для анимирования перехода свойств
2. Трансформации
3. scale, rotate, translate, skew.
4. Определение анимации с помощью @keyframes
5. Использование анимации указанной через @keyframes
6. Настройка анимации с помощью дополнительных свойств.
7. timing-function для анимации.

Свойство transition

Свойство **transition** это краткая запись четырех свойств:

transition-property	какие именно свойства следует анимировать (all - все, одно, или перечисление через запятую)
transition-duration	сколько времени займет анимация
transition-timing-function	какой тип анимации будет использован (задается функцией Безье)
transition-delay	задержка, после которой начнется анимация.

Пример применения transition

```
transition-property       : color;
transition-duration       : 1.2s;
transition-timing-function: cubic-bezier(.20, .96, .74, .07);
transition-delay          : .5s;
```

Свойство animation

Преимущества перед transition

- не нужен обязательный инициатор
- между начальным и конечным состояними возможно много промежуточных состояний

Алгоритм использования:

- создаем ключевые кадры (keyframes) - минимум два ключевых кадра
- задаем список анимируемых CSS-свойств
- назначаем анимацию элементам на странице (возможны индивидуальные скорости выполнения/задержки и пр.)

Пример применения анимации

```
@keyframes nameAnimation {
    from { opacity: 0; }
    to { opacity: 1; }
}

.className {
    animation-name: nameAnimation;
    animation-duration: 1s;
}
```

Какие свойства есть у анимации

1. **animation-name:** имя (обязательное указывать)
2. **animation-duration:** продолжительность (обязательно указывать)
3. **animation-timing-function:** функция распределения скорости по времени
4. **animation-iteration-count:** количество раз
5. **animation-direction:** направление
6. **animation-delay:** задержка
7. **animation-fill-mode:** режим заполнения

Подробнее про animation-direction и animation-fill-mode

```
animation-direction: normal|reverse|alternate|alternate-reverse|
initial|inherit

animation-fill-mode: none|forwards|backwards|both|initial|inherit;
```

Краткая запись анимации

```
.fade {
  animation: fadeOut 2s ease-in-out
infinite alternate 5s forwards;
}
```

```
.fade {
  animation-name: fadeOut;
  animation-duration: 2s;
  animation-timing-function: ease-in-out;
  animation-iteration-count: 2;
  animation-direction: alternate;
  animation-delay: 5s;
  animation-fill-mode: forwards;
}
```

Поговорим о трансформациях

transform: scale(number)

transform: rotate(angle)

transform: skew(angle)

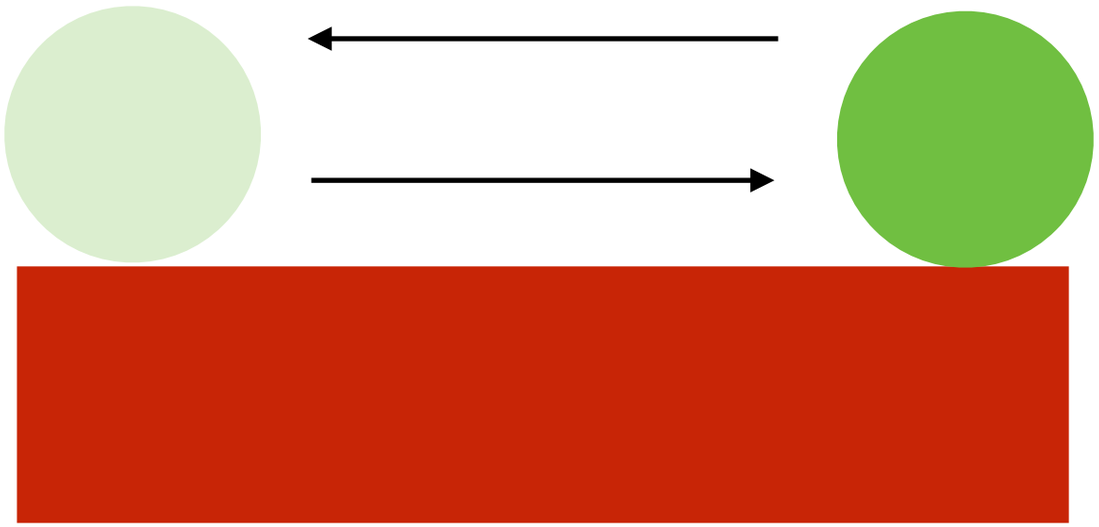
transform: translate(x, y)

Пауза анимации при наведении

```
.fade:hover {
  animation-play-state: paused;
}
```

Практическое задание

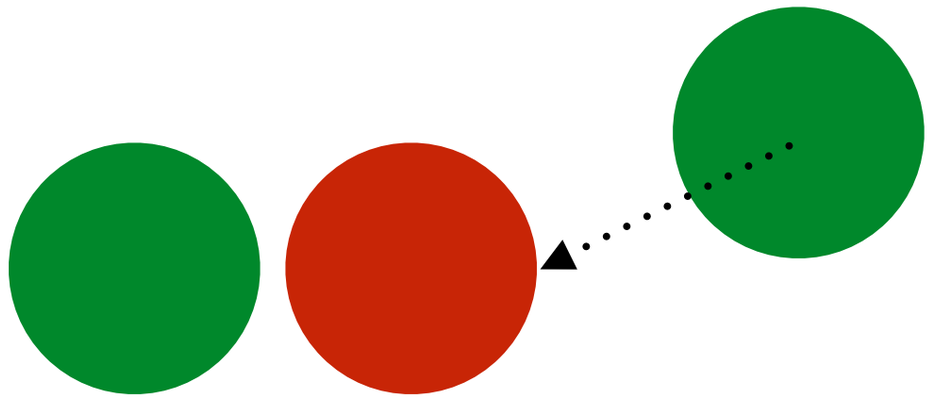
Выполнить следующую анимацию



При наведении мыши на шар, он перестает двигаться

Практическое задание 2

1 шаг



2 шаг