МДК.08.02. Графический дизайн и мультимедиа

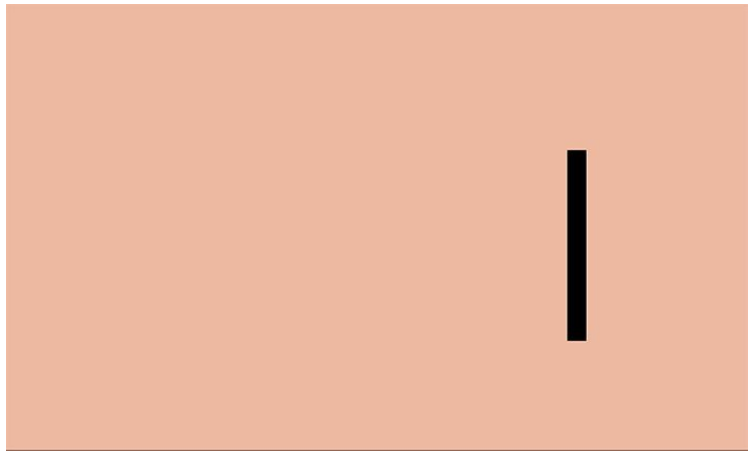
Тема 08.02.04 Динамическая графика

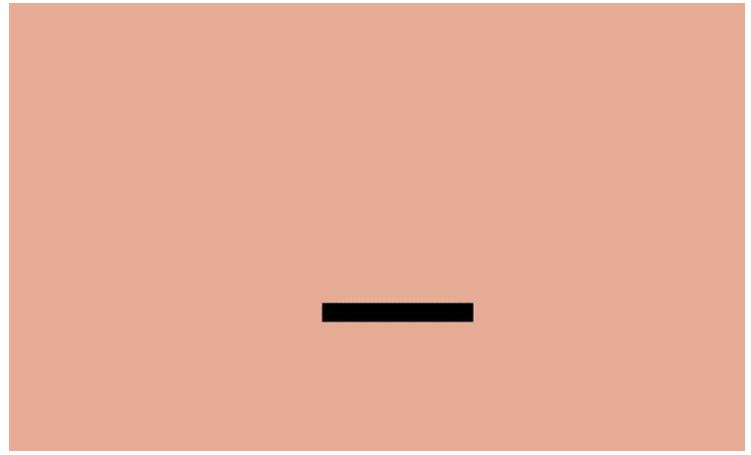
Преподаватель:
к.ф.-м.н. Монахова Ольга Александровна
tim.moa@ya.ru

Понятие динамической графики

Принципы анимации для веба

- **Анимация в вебе**
  - мощный инструмент для создания комфортного взаимодействия посетителей сайта с веб-интерфейсом
- **Истоки**
  - Книга «Иллюзия жизни: Анимация Disney» (1981 год)
    - 12 принципов анимации Диснея

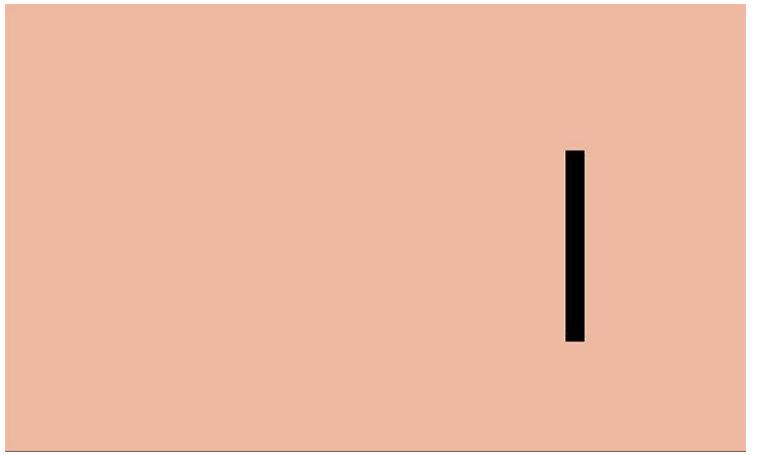




12 принципов анимации Диснея

1. **Привлекательность**
   - качественные и хорошо подобранные анимации позволяют сайту выглядеть более привлекательно и авторитетно

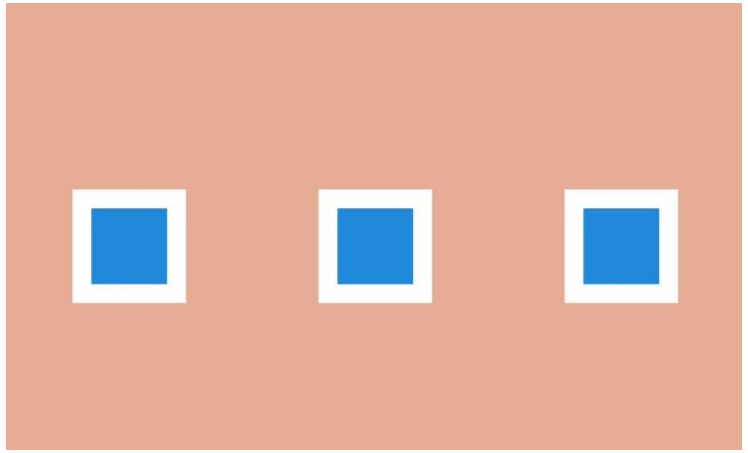
2. **Сплющивание и растяжение**
   - Деформация, связанная с физическим контактом одного графического объекта с другим



12 принципов анимации Диснея

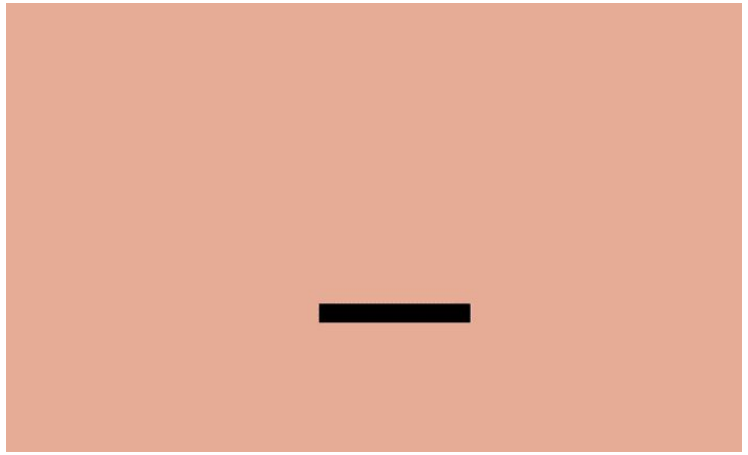
3. Фокусировка

- Один объект в качестве фокуса сцены, с ним происходит главное действие, остальные объекты – на заднем плане



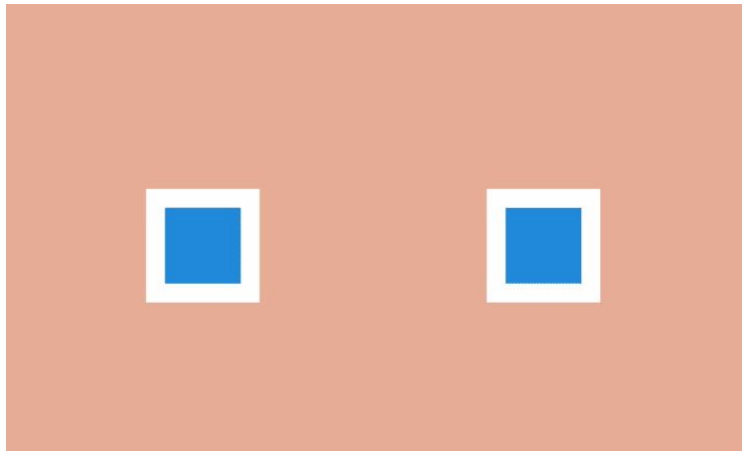
4. Ожидание

- Действие, предваряющее активное движение
- Подготовка к активному движению

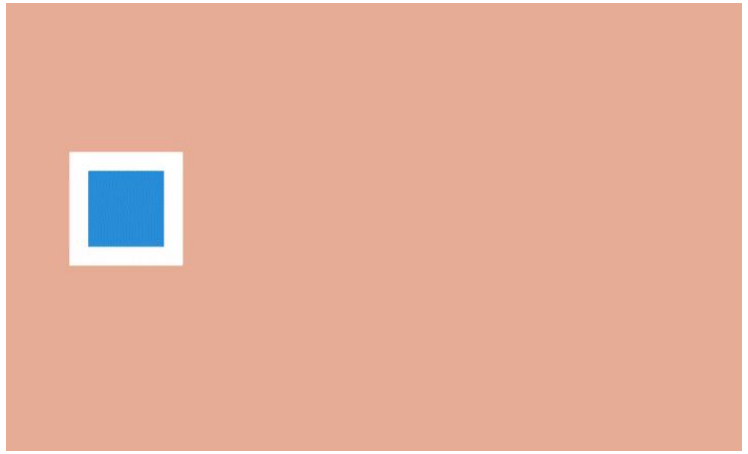


12 принципов анимации Диснея

5. **Пошаговая анимация**
   - Создание промежуточных кадров (твиннинг) между главными кадрами



6. **Следование и захлёстывание**
   - Придание объектам различной скорости движения
   - Движение по инерции
   - Торможение в преддверии остановки



12 принципов анимации Диснея

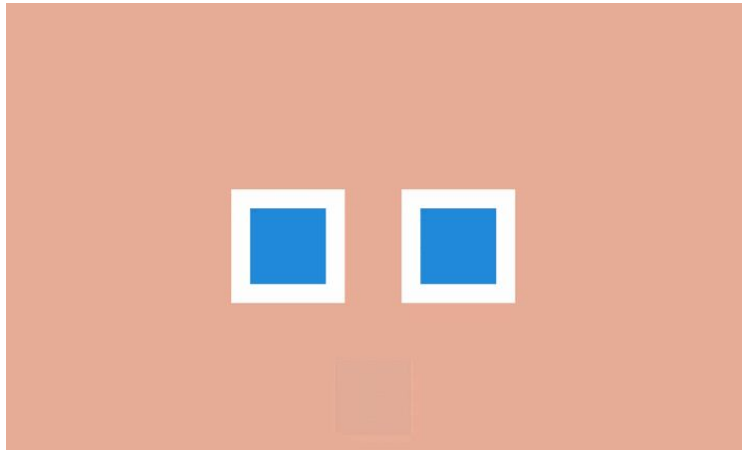
7. **Плавное ускорение и замедление**
   - Плавное начало и окончание анимации объекта

8. **Дугообразные траектории**
   - в жизни объекты редко двигаются по прямой, как правило движение идёт по изогнутой дуге

12 принципов анимации Диснея

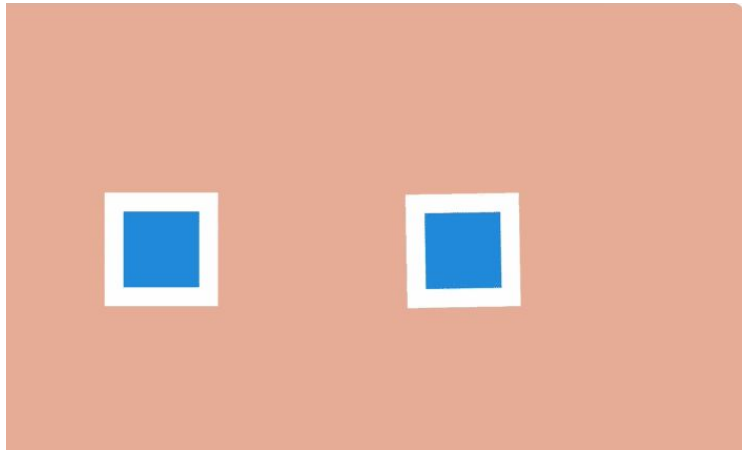
9. Второстепенное действие

- Основное действие провоцируется «второстепенной» анимацией



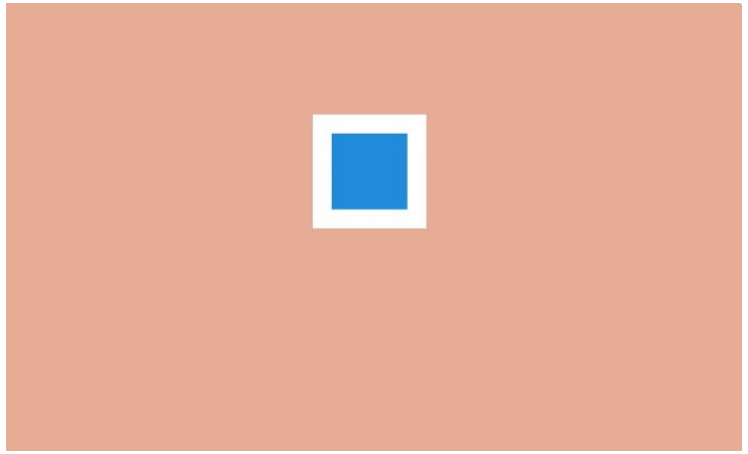
10. Время выполнения

- длительность анимации может сделать так, что одни объекты будут казаться тяжёлыми, а другие — лёгкими

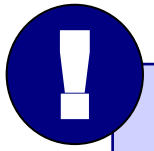


12 принципов анимации Диснея

11. **Преувеличение**
    - Привлечение внимания к определённым действиям
    - Придание действиям большей драматичности



12. **Объём**
    - При работе с объёмными объектами необходимо учитывать правила перспективы



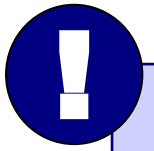
[CSS-анимации 12 принципов Диснея]

Основные инструменты

CSS - анимация

Плавные переходы

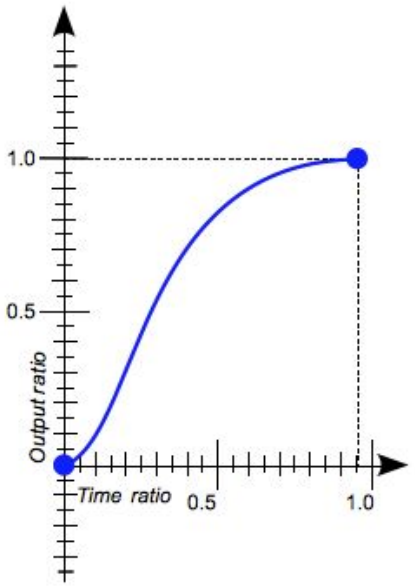
- Управление переходом между начальным и конечным состояниями
  - Что трансформируется - transition-property
  - Длительность каждой трансформации - transition-duration
  - Задержка в выполнении трансформации - transition-delay
  - «Форма» перехода - transition-timing-function



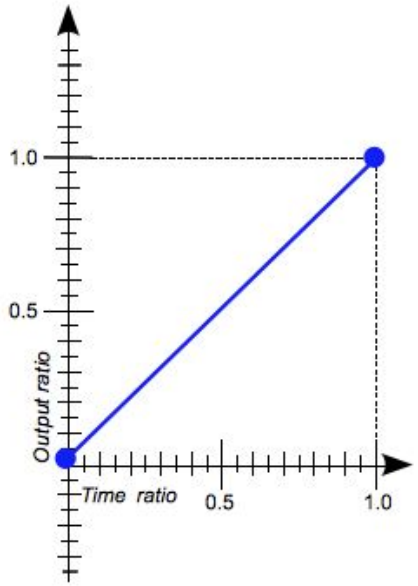
Пример использования

Какой принцип анимации Диснея?

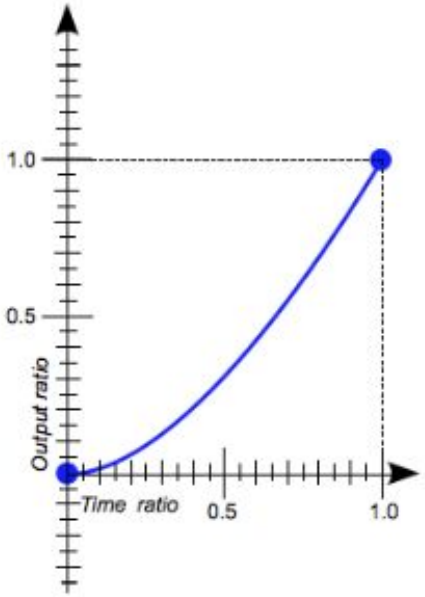
Плавные переходы: «форма» перехода



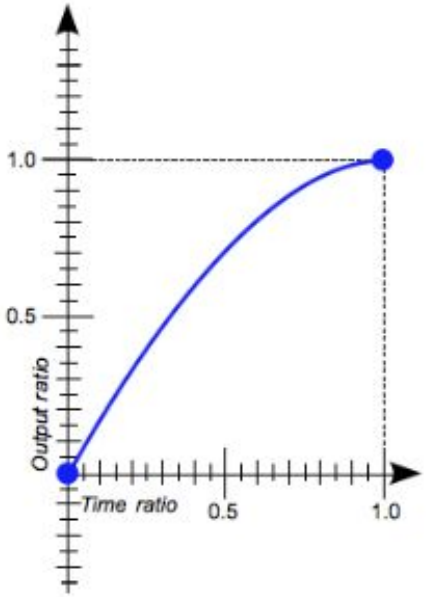
ease



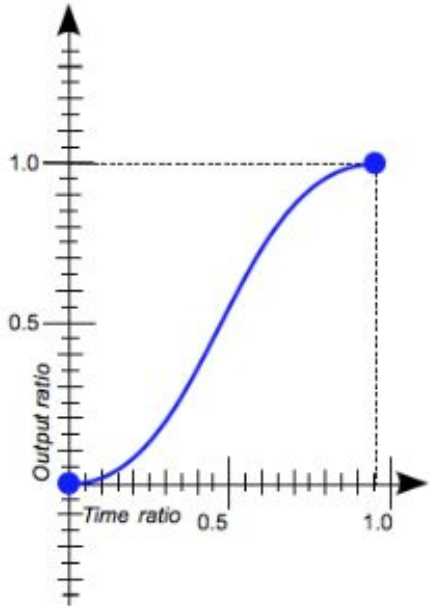
linear



ease-in



ease-out

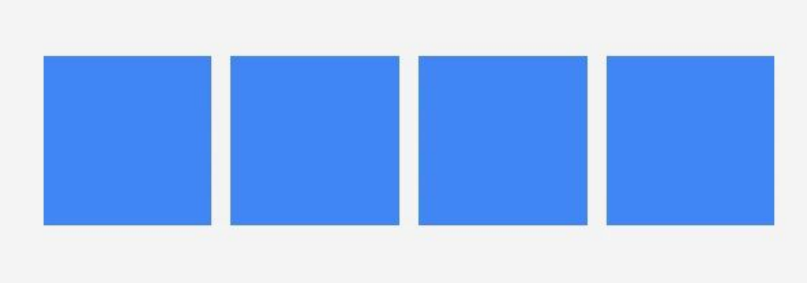


ease-in-out

cubic-bezier(0, 0, 1, 1)

Функции кривых Безье!

Исчезающий индикатор загрузки



Релиз индикатора

CSS-анимация

- Состоит из:
  - набора ключевых кадров keyframes
  - параметров самой анимации

HTML

```
<button>Кнопка</button>
```

Начальный кадр (from или 0%)
Конечный кадр (to или 100%)

CSS

```
/*набор ключевых кадров анимации*/
@keyframes stretching { /*имя анимации*/
  0% {width: 100px;} /*from {width: 100px;}*/
  100% {width: 200px;} /*to {width: 200px;}*/
}
button:hover {
  animation-name: stretching; /*имя анимации*/
  animation-duration: 1s; /*длительность анимации*/
  animation-delay: 1s; /* время задержки*/
}
```

Можно создавать промежуточные кадры!

Множественная CSS-анимация

- Одному элементу могут быть одновременно назначены несколько анимаций, в них меняются разные свойства элемента, но они будут проигрываться одновременно

HTML

```
<div class="element"></div >
```

CSS

```
@keyframes move {
  to { left: 100px; }
}
@keyframes stretch {
  to { width: 100px; }
}
.element {
  animation-name: move, stretch;
  animation-duration: 5s, 5s;
}
```

Повторяющаяся CSS-анимация

- Можно задать сколько раз будет повторяться анимация
  - animation-iteration-count: <кол-во повторов>; /* infinite - бесконечно*/

HTML

```
<div class="element"></div >
```

CSS

```
@keyframes move {
  to { left: 100px; }
}
@keyframes stretch {
  to { width: 100px; }
}
.element {
  animation-name: move, stretch;
  animation-duration: 5s, 5s;
  animation-iteration-count: infinite;
}
```

Направление CSS-анимации

- Можно задать направление анимации …
  - animation-direction: normal; /* с начального кадра к конечному*/
  - animation-direction: reverse; /* с конечного кадра к начальному*/
- … а также чередующееся направление анимации, когда количество проигрываний анимации (animation-iteration-count) больше одного
  - animation-direction: alternate; /*нечётные проигрывания в прямом направлении, а чётные — в обратном*/
  - animation-direction: alternate-reverse; /*чётные проигрывания в прямом направлении, а нечётные — в обратном*/

HTML

```
<div class="element"></div >
```

move выполнится два раза:

- **в первый (нечётный) раз направление будет прямым**
- **второй (чётный) — обратным**

CSS

```
.element {
  animation-name: move;
  animation-duration: 1s;
  animation-iteration-count: 2;
  animation-direction: alternate;
}
```

Состояние до и после анимации

- после проигрывания анимации возвращаются в исходное состояние, но можно задать свойство, которое
  - будет сохранять состояние после завершения анимации
    - animation-fill-mode: forwards;
  - определяет состояние элемента до начала анимации
    - animation-fill-mode: backwards;
  - объединяет действия forwards и backwards
    - animation-fill-mode: both;
- работает и в случае многоразовой анимации или чередующегося направления

Если элементу назначена анимация с задержкой и определено состояние элемента до начала анимации, то стили, описанные в первом ключевом кадре, будут применены сразу, ещё до начала проигрывания анимации

Остановка и запуск анимации

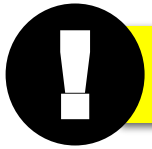
- можно поставить анимацию «на паузу», а потом возобновить с места остановки
  - animation-play-state: paused; /*приостанавливает анимацию*/
  - animation-play-state: running; /*начинает или возобновляет анимацию*/

HTML

```
<div class="element"></div >
```

CSS

```
.element {
  animation-name: move;
  animation-duration: 1s;
  animation-iteration-count: 2;
  animation-direction: alternate;
}
. element:active{
  animation-play-state: paused;
}
```

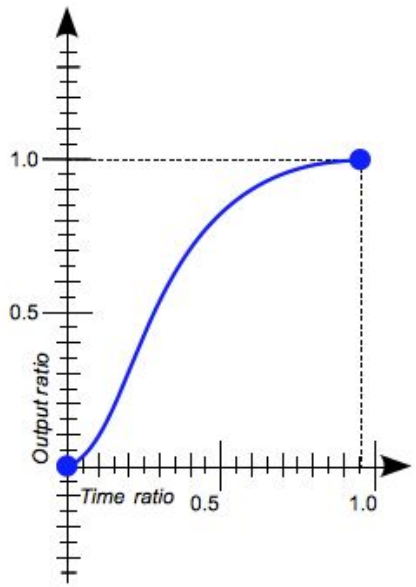


Пример анимации!

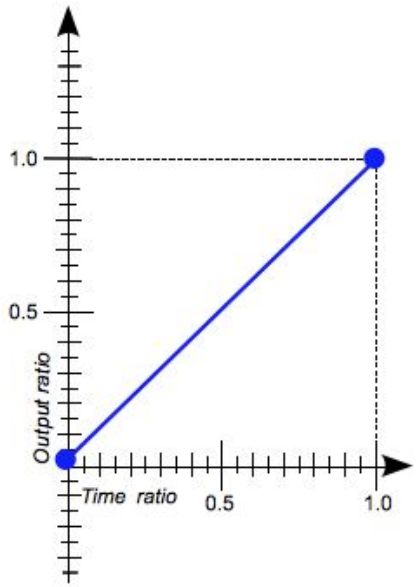
«Форма» анимации

Где-то мы это уже видели!

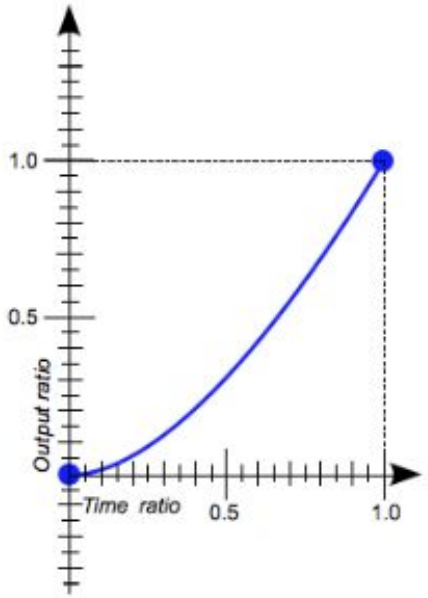
- определяет, как именно будет происходить анимация: с какой скоростью и ускорением будут меняться свойства, задействованные в ней
  - animation-timing-function



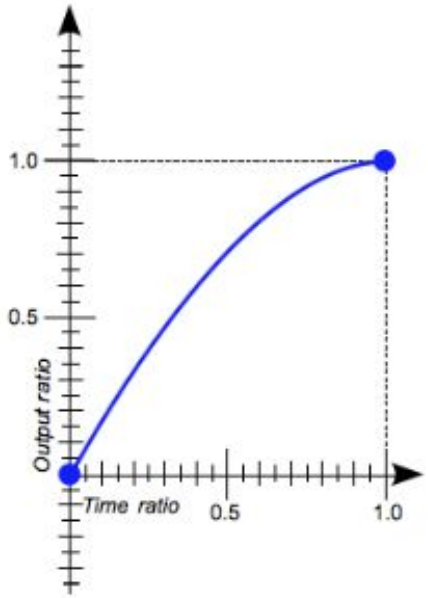
ease



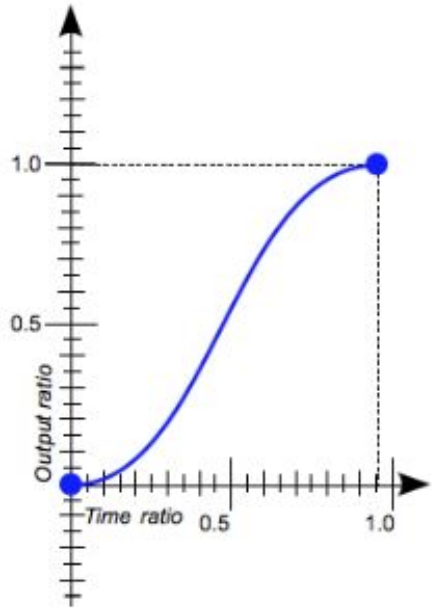
linear



ease-in



ease-out



ease-in-out

cubic-bezier(0, 0, 1, 1)

Функции кривых Безье!