Разработка пользовательского веб-интерфейса

History API, расширенные возможности современных браузеров и HTML5. PWA. Анимации CSS

Сергей Геннадьевич Синица

КубГУ, 2021

sin@kubsu.ru

Видео

```
<video class="video-wrapper" preload="auto" autoplay="true" loop="true" muted="muted">

  <source src="video.mp4" type="video/mp4"><!-- MP4 для Safari, IE9+, iPhone, iPad, Android, и Windows Phone 7 -->

  <source src="video.webm" type="video/webm"><!-- WebM/VP8 для Firefox4, Opera, и Chrome -->

  <source src="video.ogv" type="video/ogg"><!-- Ogg/Vorbis для старых версий браузеров Firefox и Opera -->

  <object data="video.swf" type="application/x-shockwave-flash"><!-- добавляем видеоконтент для устаревших браузеров, в которых нет поддержки элемента video →
    <param name="movie" value="video.swf">
  </object>

</video>
```

Шрифт

```
@font-face {
    font-family: "WebFont";
    src: url(WebFont.woff) format("woff"), /* все современные браузеры, IE9+ */
         url(WebFont.ttf) format("truetype"); /* все современные браузеры */
}
p {font-family: "WebFont", Arial, sans-serif; }
```

Объект window.location

http://www.google.com:80/search?q=javascript#test
window.location.toString()
window.location.assign()

Свойство	Описание	Пример
hash	часть URL, которая идет после символа решетки '#', включая символ '#'	#test
host	хост и порт	www.google.com:80
href	весь URL	http://www.google.com:80 /search?q=javascript#test
hostname	хост (без порта)	www.google.com
pathname	строка пути (относительно хоста)	/search
port	номер порта	80
protocol	протокол	http:
search	часть адреса после символа ?, включая символ ?	?q=javascript

При изменении любых свойств window.location, кроме hash, документ будет перезагружен

window.location.hash

http://www.google.com:80/search?q=javascript#test

```
$(window).on('hashchange', function() {
  //.. work ..
});
```

HTML5 onhashchange

```
if ("onhashchange" in window) {
    alert("The browser supports the hashchange event!");
}

function locationHashChanged() {
    if (location.hash === "#somecoolfeature") {
        somecoolfeature();
    }
}

window.onhashchange = locationHashChanged;
```

Caniuse 98%

History API

Объект window.history…

Позволяет двигаться вперед и назад по истории пользователя:

```
window.history.back();
window.history.forward();
window.history.go(-1);
window.history.go(1);
var numberOfEntries = window.history.length;
```

Начиная с HTML5 - манипулировать содержимым стека истории.

Caniuse 98%

history.pushState()

Скрипт на странице http://mozilla.org/foo.html:

```
var stateObj = { foo: "bar" };
history.pushState(stateObj, "page 2", "bar.html");
```

Адресная строка станет показывать http://mozilla.org/bar.html, но браузер не будет загружать страницу bar.html и даже проверять существует ли она.

Переходим на http://google.com, а затем кликаем Назад.
В URL отобразится http://mozilla.org/bar.html, а на странице произойдет событие popstate, тогда объект состояния будет содержать копию stateObj.

Сама страница будет выглядеть как foo.html, хотя страница могла изменить своё содержание из-за события popstate.

Пример

Изменяет текущую запись в истории.

```
window.onpopstate = function(event) {
  alert("location: " + document.location + ", state: " +
JSON.stringify(event.state));
};

history.pushState({page: 1}, "title 1", "?page=1");
history.pushState({page: 2}, "title 2", "?page=2");
history.replaceState({page: 3}, "title 3", "?page=3");
history.back(); // alerts "location: http://example.com/example.html?
page=1, state: {"page":1}"
history.back(); // alerts "location: http://example.com/example.html,
state: null
history.go(2);  // alerts "location: http://example.com/example.html?
page=3, state: {"page":3}
```

window.localStorage / sessionStorage

```
// use localStorage for persistent storage
// use sessionStorage for per tab storage

saveButton.addEventListener('click', function () {
  window.localStorage.setItem('value', area.value);
  window.localStorage.setItem('timestamp', (new Date()).getTime());
}, false);

textarea.value = window.localStorage.getItem('value');
```

Сессия страницы остается активной все время пока окно браузера открыто и сохраняется между перезагрузками страниц. Открытие той же страницы в новом окне браузера или новой вкладке приводит к созданию новой сессии страницы.

Caniuse 98%.

Web SQL Storage

```
var db = window.openDatabase("DBName", "1.0", "description", 5*1024*1024);
//5MB
db.transaction(function(tx) {
  tx.executeSql("SELECT * FROM test", [], successCallback, errorCallback);
});
```

Caniuse 77% (80% в 2020, 82% в 2019).

W3C не развивают с 2010 года.

Indexed DB

IndexedDB – это встроенная база данных, более мощная, чем localStorage.

Хранилище ключей/значений: доступны несколько типов ключей, а значения могут быть (почти) любыми.
Поддерживает транзакции для надёжности.
Поддерживает запросы в диапазоне ключей и индексы.
Позволяет хранить больше данных, чем localStorage.

Caniuse 98% (95.7% в 2020, 92% в 2019).

https://www.w3.org/TR/IndexedDB/

Indexed DB

```
let openRequest = indexedDB.open("db", 2);

// создаём хранилище объектов для books, если ешё не существует
openRequest.onupgradeneeded = function() {
  let db = openRequest.result;
  if (!db.objectStoreNames.contains('books')) { // if there's no "books" store
    db.createObjectStore('books', {keyPath: 'id'}); // create it
  }
};

let transaction = db.transaction("books", "readwrite"); // (1)

// получить хранилище объектов для работы с ним
let books = transaction.objectStore("books"); // (2)

let book = {
  id: 'js',
  price: 10,
  created: new Date()
};

let request = books.add(book); // (3)

request.onsuccess = function() { // (4)
  console.log("Книга добавлена в хранилище", request.result);
};

request.onerror = function() {
  console.log("Ошибка", request.error);
};
```

Web Sockets

```
//https://tools.ietf.org/html/rfc6455

// Создает WebSocket - подключение.
const socket = new WebSocket('ws://localhost:8080');

// Соединение открыто
socket.addEventListener('open', function (event) {
    socket.send('Hello Server!');
});

// Наблюдает за сообщениями
socket.addEventListener('message', function (event) {
    console.log('Message from server ', event.data);
});
```

Caniuse 98%

Web Workers

main.js:

```
var worker = new Worker('task.js');
worker.onmessage = function(event) { alert(event.data); };
worker.postMessage('data');
```

task.js:

```
self.onmessage = function(event) {
  // Do some work.
  self.postMessage("recv'd: " + event.data);
};
```

Caniuse 98%

Canvas

```
<canvas id="canvas" width="838" height="220"></canvas>

<script>
  var canvasContext = document.getElementById("canvas").getContext("2d");
  canvasContext.fillRect(250, 25, 150, 100);

  canvasContext.beginPath();
  canvasContext.arc(450, 110, 100, Math.PI * 1/2, Math.PI * 3/2);
  canvasContext.lineWidth = 15;
  canvasContext.lineCap = 'round';
  canvasContext.strokeStyle = 'rgba(255, 127, 0, 0.5)';
  canvasContext.stroke();
</script>
```

Caniuse 98%

Inline SVG

```
<html>
  <svg width="300px" height="300px">
    <defs>
      <linearGradient id="myGradient" x1="0%" y1="100%" x2="100%" y2="0%">
        <stop offset="5%" stop-color="red"></stop>
        <stop offset="95%" stop-color="blue" stop-opacity="0.5"></stop>
      </linearGradient>
    </defs>
    <circle id="myCircle" class="important" cx="50%" cy="50%" r="100"


        fill="url(#myGradient)" onmousedown="alert('hello');"/>
  </svg>
</html>
```

Caniuse 98%

WebGL

```
<canvas id="canvas" width="838" height="220"></canvas>

<script>
  var gl = document.getElementById("canvas").getContext("experimental-
webgl");
  gl.viewport(0, 0, canvas.width, canvas.height);
  ...
</script>
```

Caniuse 98%

Progressive Web Apps (PWA)

Позволяет устанавливать оффлайн-версию сайта в виде мобильного приложения на телефон прямо из браузера.

Приложение можно публиковать в Apple Appstore и Google Play наравне с нативными мобильными приложениями.

Доступны все веб-технологии, офлайновая работа, Push-уведомления (только Android).

Google развивает, Apple тормозит развитие.

Сервисы типа pwabuilder.com от Microsoft анализируют сайт и дают пошаговую инструкцию по построению PWA.

Caniuse Add to Home Screen 87%

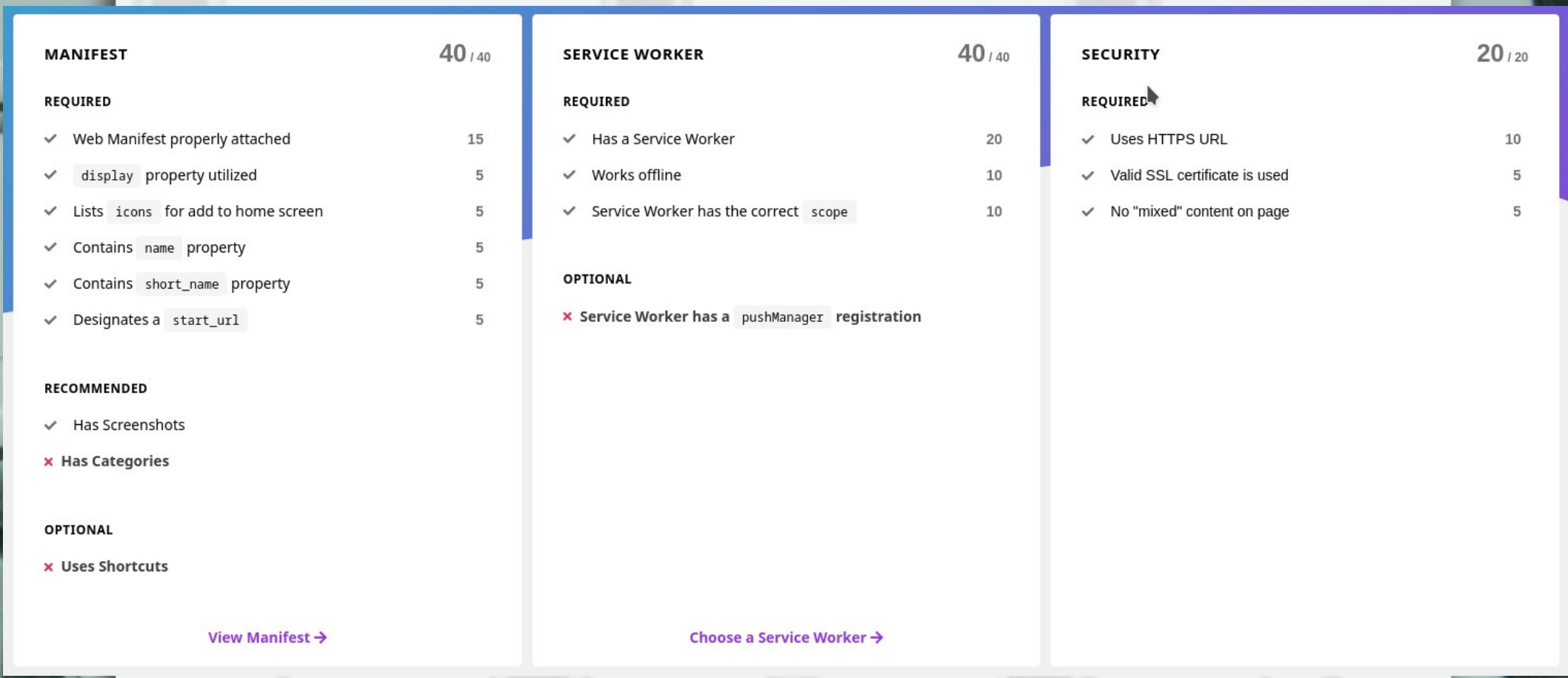
Progressive Web Apps (PWA)

Google indicates that a progressive web app, PWA, is:

Progressive - Works for every user, regardless of browser choice because it's built with progressive enhancement as a core tenet.
Responsive - Fits any form factor: desktop, mobile, tablet, or whatever is next.
Connectivity independent - Enhanced with service workers to work offline or on low-quality networks.
App-like - Feels like an app, because the app shell model separates the application functionality from application content .
Fresh - Always up-to-date thanks to the service worker update process.
Safe - Served via HTTPS to prevent snooping and to ensure content hasn't been tampered with.
Discoverable - Is identifiable as an "application" thanks to W3C manifest and service worker registration scope, allowing search engines to find it.
Re-engageable - Makes re-engagement easy through features like push notifications.
Installable - Allows users to add apps they find most useful to their home screen without the hassle of an app store.
Linkable - Easily share the application via URL, does not require complex installation.

See https://web.dev/progressive-web-apps/#what_is_a_progressive_web_app

Progressive Web Apps (PWA)



<link rel='manifest' href='/manifest.json'>


```
{
    "dir": "ltr",
    "lang": "ru",
    "name": "Drupal Coder - разработка сайтов на drupal",
    "scope": "/",
    "display": "browser",
    "start_url": "https://drupal-coder.ru/",
    "short_name": "Drupal Coder",
    "theme_color": "transparent",
    "description": "Создание сайтов на drupal 8 и 7, поддержка и доработка друпал
        сайтов любой сложности и запущенности. DrupalCoder - это команда с 10 летним
        опытом в drupal.",
    "orientation": "any",
    "background_color": "#000000",
    "related_applications": [],
    "prefer_related_applications": false,
    "icons": [
        {
            "src": "https://drupal-coder.ru/sites/default/files/
                android-chrome-256x256.png",
            "sizes": "256x256"
        }
    ],
    "url": "https://drupal-coder.ru",
    "screenshots": []
}
```

CSS-переходы

CSS-transition позволяют декларативно задать плавное изменение свойств DOM для создания эффекта анимации.

При необходимости анимации в ответ на действия пользователя, например, для плавного раскрытия меню, на клик срабатывает JS, который включает или отключает у элемента класс CSS. При смене класса CSS меняется положение элемента, а CSS-transition обеспечивает анимацию.

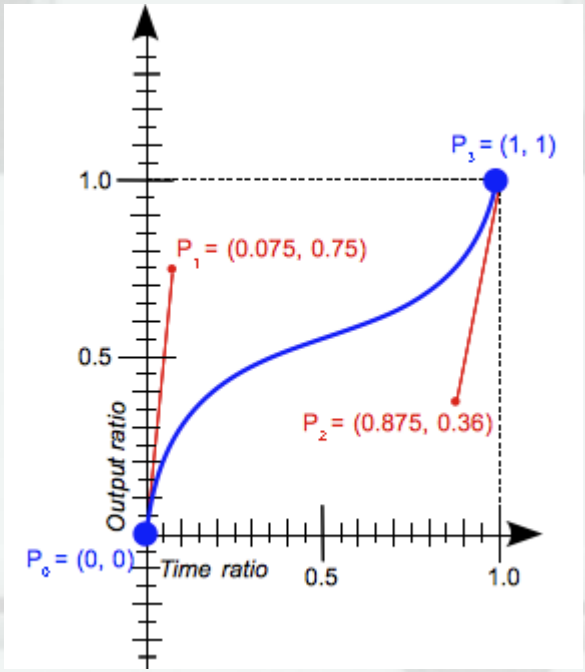
https://html5book.ru/css3-transition/

https://developer.mozilla.org/ru/docs/Web/CSS/CSS_Transitions/Using_CSS_transitions

Caniuse 98%

```
<style>


/* Свойства закрытого меню */
.button {
  width: 100px;
  height: 25px;
  background-color: green;
  color: white;
  /* CSS-трансформация, все свойства будут плавно меняться одну секунду */
  transition: all 1s ease;
}


/* Свойства раскрытого меню */
.opened {
  background-color: red;
  color: black;
  height: 400px;
}
</style>


<script>
// Ждем загрузки DOM. Элемент <body> будет загружен автоматически.
window.addEventListener('DOMContentLoaded', function() {
  var button = document.getElementById('button');
  button.addEventListener('click', function(event) {
    // Переключаем класс .opened
    event.target.classList.toggle('opened');
  });


});
</script>


<div id="button" class="button">Click Me!</div>
```

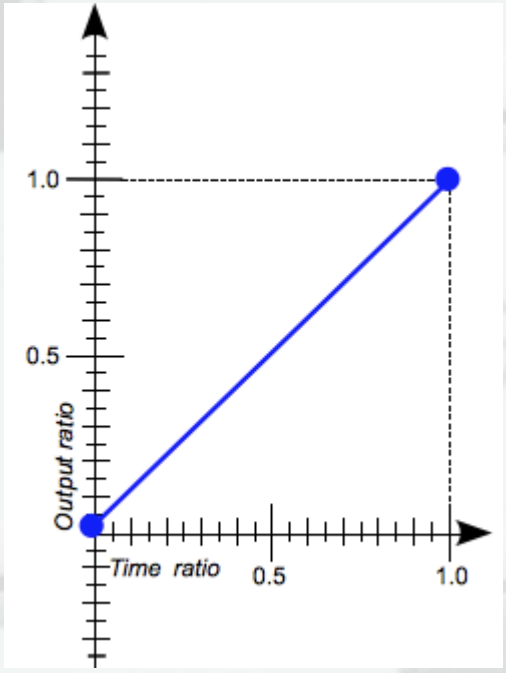
CSS-переходы



cubic-bezier(x1, y1, x2, y2)

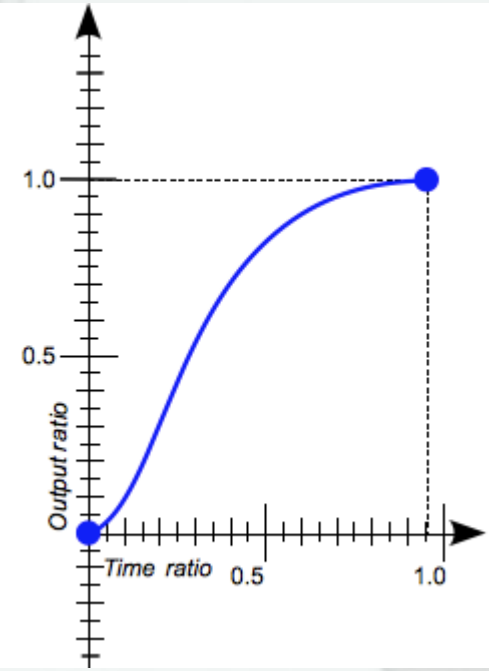
https://developer.mozilla.org/ru/docs/Web/CSS/timing-function

CSS-переходы



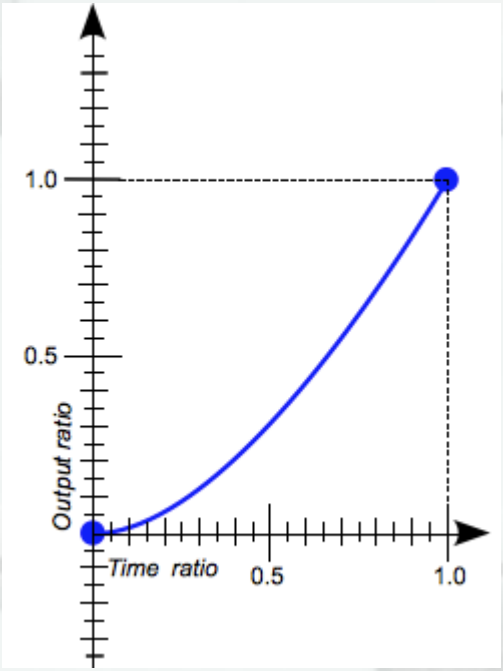
linear
cubic-bezier(0.0, 0.0, 1.0, 1.0)

CSS-переходы



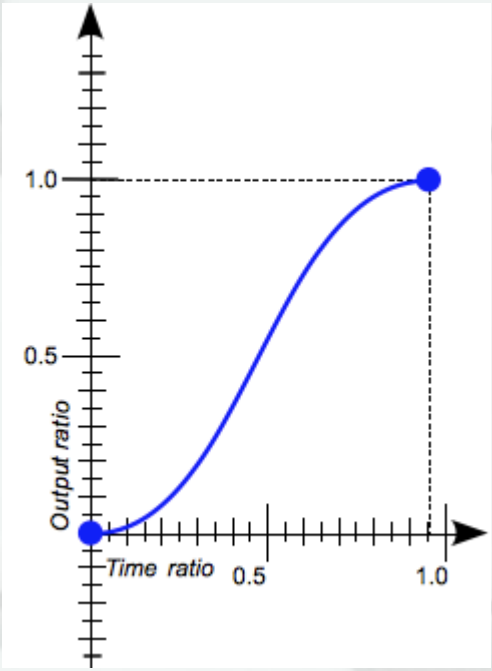
ease (default)
cubic-bezier(0.25, 0.1, 0.25, 1.0)

CSS-переходы



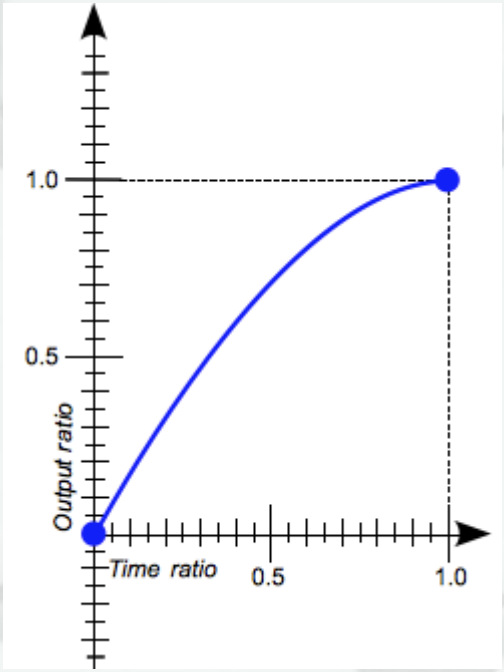
ease-in
cubic-bezier(0.42, 0.0, 1.0, 1.0)

CSS-переходы



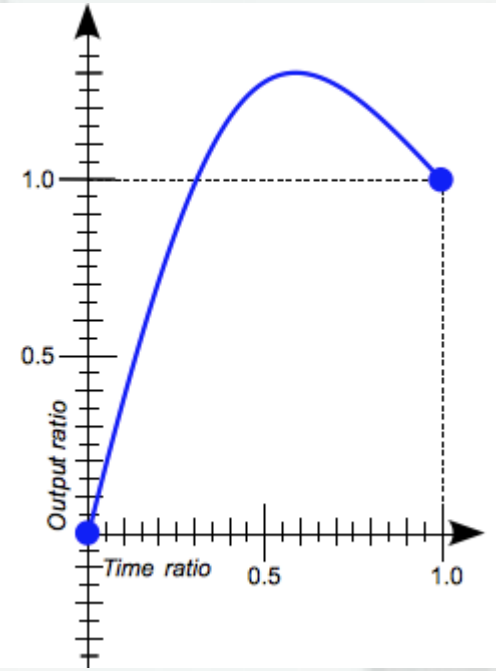
ease-in-out
cubic-bezier(0.42, 0.0, 0.58, 1.0)

CSS-переходы



ease-out
cubic-bezier(0.0, 0.0, 0.58, 1.0)

CSS-переходы



cubic-bezier(0.175, 0.885, 0.32, 1.275)

CSS-переходы

https://easings.net/ru

CSS-анимация

```
p {
  animation-duration: 3s;
  animation-name: slidein;
}

@keyframes slidein {
  from {
    margin-left: 100%;
    width: 300%;
  }

  to {
    margin-left: 0%;
    width: 100%;
  }
}
```

CSS-анимация

```
p {
  animation-duration: 3s;
  animation-name: slidein;
}

@keyframes slidein {
  from {
    margin-left: 100%;
    width: 300%;
  }

  to {
    margin-left: 0%;
    width: 100%;
  }

  75% {
    animation-timing-function: ease-in; /* до следующего кадра */
    font-size: 300%;
    margin-left: 25%;
    width: 150%;
  }
}
```

CSS-анимация

```
var e = document.getElementById("watchme");
e.addEventListener("animationstart", listener);
e.addEventListener("animationend", listener);
e.addEventListener("animationiteration", listener);

e.className = "slidein";
```