



Оптимизация сайта.

Асинхронная загрузка CSS

loadCSS - простой js скрипт
позволяющий загружать CSS асинхронно

Можно скачать на GitHub:

<https://github.com/filamentgroup/loadCSS>

loadCSS. Зачем?

Большинство браузеров, встретив `link[rel=stylesheet]` или `@import`, остановят рендер страницы до тех пор пока не загрузится файл стилей.

Не все из загружаемых стилей необходимы для базового рендера страницы.

loadCSS

После подключения функции loadCSS, чтобы асинхронно загрузить стили, достаточно такого скрипта:

```
<script>loadCSS( "path/to/mystylesheet.css"  
);</script>
```

loadCSS callback

При необходимости, можно подключить к проекту функцию onLoadCSS, чтобы отследить загрузку стилей.

```
var stylesheet = loadCSS(  
  "path/to/mystylesheet.css" ); onloadCSS(  
  stylesheet, function() {  
    console.log( "Stylesheet has loaded." );  
  });
```

loadCSS. Поддержка браузеров.

Browser	CSS Loads Successfully	CSS Loads without Blocking Render
Chrome Mac (latest and many recent versions)	Yes	Yes
Firefox Desktop (latest and many recent versions)	Yes	Yes
Opera Mac (latest and many recent versions)	Yes	Yes
Safari Mac (latest and many recent versions)	Yes	Yes
Safari iOS (latest and many recent versions)	Yes	Yes
Chrome Android 5.x	Yes	Yes
Chrome Android 4.x	Yes	Yes
Android Browser 2.3	Yes	No
Kindle Fire HD	Yes	Yes
Windows Phone IE 8.1	Yes	Yes
IE 11	Yes	Yes
IE 10	Yes	Yes
IE 9	Yes	Yes
IE 8	Yes	No
IE 7	Yes	No
IE 6	Yes	No

Оптимизация шрифтов

Загрузка шрифтов блокирует отображение страницы.

К сожалению, использование google.com/fonts ведет к потере производительности при рендере страницы.

Оптимизация шрифтов

Можно загрузить шрифт с google.com/fonts при помощи loadCSS.

Этот подход не даст шрифту блокировать отображение контента при загрузке страницы.

Минус: мигание текста при каждой загрузке страницы.

Оптимизация шрифтов

Для большей оптимизации будем использовать шрифты расположенные на своем сервере.

Web Open Font Format (WOFF) – оптимизированный под работу в web формат шрифтов.



Оптимизация шрифтов

Не забываем смотреть на то с какой лицензией распространяются шрифты!



Оптимизация шрифтов

Как перекодировать шрифт в woff и в woff2?

<https://www.fontsquirrel.com/tools/webfont-generator>

Оптимизация шрифтов

Для большего прироста скорости, можно закодировать шрифт и передавать его как CSS файл с base64 данными.

Этот файл можно сгенерировать в `fontsquirrel generator`.

Оптимизация шрифтов

Теперь если через `loadCSS` грузить шрифты как `.css` файлы скорость рендера страницы будет лучше.

Но шрифт будет подгружаться заново при каждом открытии страницы.

localStorage

В стандарте HTML-5 введено веб-хранилище. Оно позволяет хранить на стороне клиента пары «ключ» : «значение».

Как и кукисы, эти данные сохраняются даже после ухода с сайта, закрытия вкладки браузера, с выходом из браузера.

localStorage

Используя javascript можно получить доступ к хранилищу через объект **localStorage** глобального объекта **window**.

```
function supports_html5_storage() {  
    try {  
        return 'localStorage' in window &&  
            window['localStorage'] !== null;  
    } catch (e) {  
        return false;  
    }  
}
```

localStorage

Интерфейс хранилища {

Получить через getItem(ключ);

Установить через setItem(ключ,
данные);
};

Можно использовать как ассоциативный массив.

localStorage

```
var foo = localStorage.getItem("bar");  
// ...  
localStorage.setItem("bar", foo);
```

```
var foo = localStorage["bar"];  
// ...  
localStorage["bar"] = foo;
```


localStorage

Интерфейс хранилища {

Удалить через `removeItem(ключ);`

`clear();`

}

localStorage

Событие `storage` позволяет отслеживать изменения хранилища.

```
if (window.addEventListener) {  
    window.addEventListener("storage",  
    handle_storage, false);  
} else {  
    window.attachEvent("onstorage",  
    handle_storage);  
};
```

localStorage

Функция обратного вызова `handle_storage` будет вызвана с объектом `StorageEvent`, за исключением Internet Explorer, где события хранятся в `window.event`.

```
function handle_storage(e) {  
    if (!e) { e = window.event; }  
}
```

localStorage

В данном случае переменная `e` будет объектом `StorageEvent`, который обладает следующими полезными свойствами.

localStorage

Объект StorageEvent		
Свойство	Тип	Описание
key	string	Ключ может быть добавлен, удален или изменен.
oldValue	любой	Предыдущее значение (если переписано) или null, если добавлено новое значение.
newValue	любой	Новое значение или null, если удалено.
url*	string	Страница, которая вызывает метод, приведший к изменению.

localStorage

Ограничения: 5 мб

`QUOTA_EXCEEDED_ERR` –

исключение при превышении квоты в 5 мб.

Оптимизация шрифтов

Асинхронно загружая закодированный в base64 шрифт, можно поместить его в localStorage и при следующем файле загружать шрифт оттуда.

Загрузка из localStorage 5-50мс. Со второй загрузки страницы пользователи не увидят мерцания.

Оптимизация шрифтов

```
<script>
(function(){
  function addFont() {
    var style =
document.createElement('style');
    style.rel = 'stylesheet';
    document.head.appendChild(style)
;
    style.textContent =
localStorage.sourceSansPro;
  }

  try {
    if (localStorage.sourceSansPro) {
      // The font is in localStorage, we
      can load it directly
      addFont();
    } else {
      // We have to first load the font
      file asynchronously
      var request = new
XMLHttpRequest();
      request.open('GET',
'/path/to/source-sans-pro.woff.css',
true);
```

```
      request.onload = function() {
        if (request.status >= 200 &&
request.status < 400) {
          // We save the file in localStorage
          localStorage.sourceSansPro =
request.responseText;

          // ... and load the font
          addFont();
        }
      }

      request.send();
    }
  } catch(ex) {
    // maybe load the font synchronously for
    woff-capable browsers
    // to avoid blinking on every request when
    localStorage is not available
  }
})();
</script>
```


Оптимизация шрифтов

Скрипт loadFont. Позволяет загружать асинхронно woff и woff2 шрифты. Если woff2 не подгрузится, сработает woff. Работает с localStorage.

<https://github.com/bdadam/OptimizedWebfontLoading>

Оптимизация шрифтов. Пример:

```
<script>function loadFont(a,b,c){function
d(){if(!window.FontFace)return!1;var a=new
FontFace("t",'url("data:application/font-woff2,") format("woff2"));return
a.load(),"loading"===a.status}var
e=navigator.userAgent,f=!window.addEventListener||e.match(/(Android
(2|3|4.0|4.1|4.2|4.3))|(Opera
(Mini|Mobi)))/)&&!e.match(/Chrome/);if(!f){var
g={};try{g=localStorage||{}}catch(h){}var i="x-font-
"+a,j=i+"url",k=i+"css",l=g[j],m=g[k],n=document.createElement("style");
if(n.rel="stylesheet",document.head.appendChild(n),!m||l!==b&&l!==c){v
ar o=c&&d()?c:b,p=new
XMLHttpRequest;p.open("GET",o),p.onload=function(){p.status>=200&
&p.status<400&&(g[j]=o,g[k]=n.textContent=p.responseText)},p.send()}
else n.textContent=m}}</script>
```

```
<script>loadFont('Comfortaa', '/fonts/comfortaa-woff.css',
'/fonts/comfortaa-woff2.css')</script>
```

```
<script>loadFont('PT Sans', '/fonts/ptsans-woff.css', '/fonts/ptsans-
woff2.css')</script>
```

Оптимизация CSS

Мы можем убрать блокирующее отображение CSS стилей. Но как при этом оставить быстрым отображением видимого контента?

Оптимизация CSS

Разбиваем наш CSS на два файла.

Первый – оформление видимой при загрузке страницы части сайта.

Второй – остальные стили.

Оптимизация CSS

Скрипт из первого файла загружаем непосредственно в HTML страницу. В head в блоке `<style></style>`. Сразу после вызова `loadFont`.

Затем подключаем `loadCSS` и асинхронно подгружаем остальные стили.

Оптимизация CSS

Минифицируйте весь основной CSS код.

Дополнительным плюсом будет использование autoprefixer.

Оптимизация JS

Сжатие JS:

[http://closure-
compiler.appspot.com/home](http://closure-compiler.appspot.com/home)



Оптимизация на сервере

Использовать сжатие Gzip

Кеширование статичных элементов.

Оптимизация изображений

Нет смысла хранить изображения формата 1920px на 1080px, когда на сайте они не отображаются размером больше 800 на 600.

Уменьшаем все изображения до необходимого соотношения сторон.

Оптимизация изображений

Затем можно сжать изображения еще сильнее (не теряя качество) используя одну из рекомендуемых для этого утилит.

На windows: FileOptimizer

Под linux: OptiPNG или PNGOUT.



Оптимизация изображений

Еще одна оптимизация – используйте спрайты.

Инструменты для оценки скорости загрузки сайта

PageSpeed Insights:

<https://developers.google.com/speed/pagespeed/insights/>

Test a website's performance:

<http://www.webpagetest.org/>