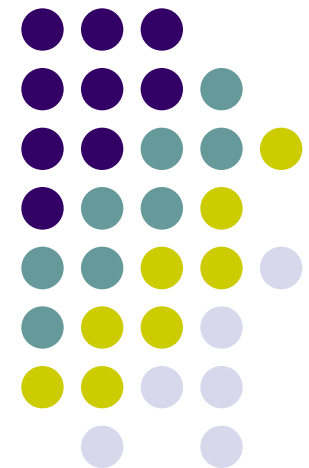


# HTTP Pipelining

## Оптимизация порядка загрузки



# HTTP Pipelining

## Введение

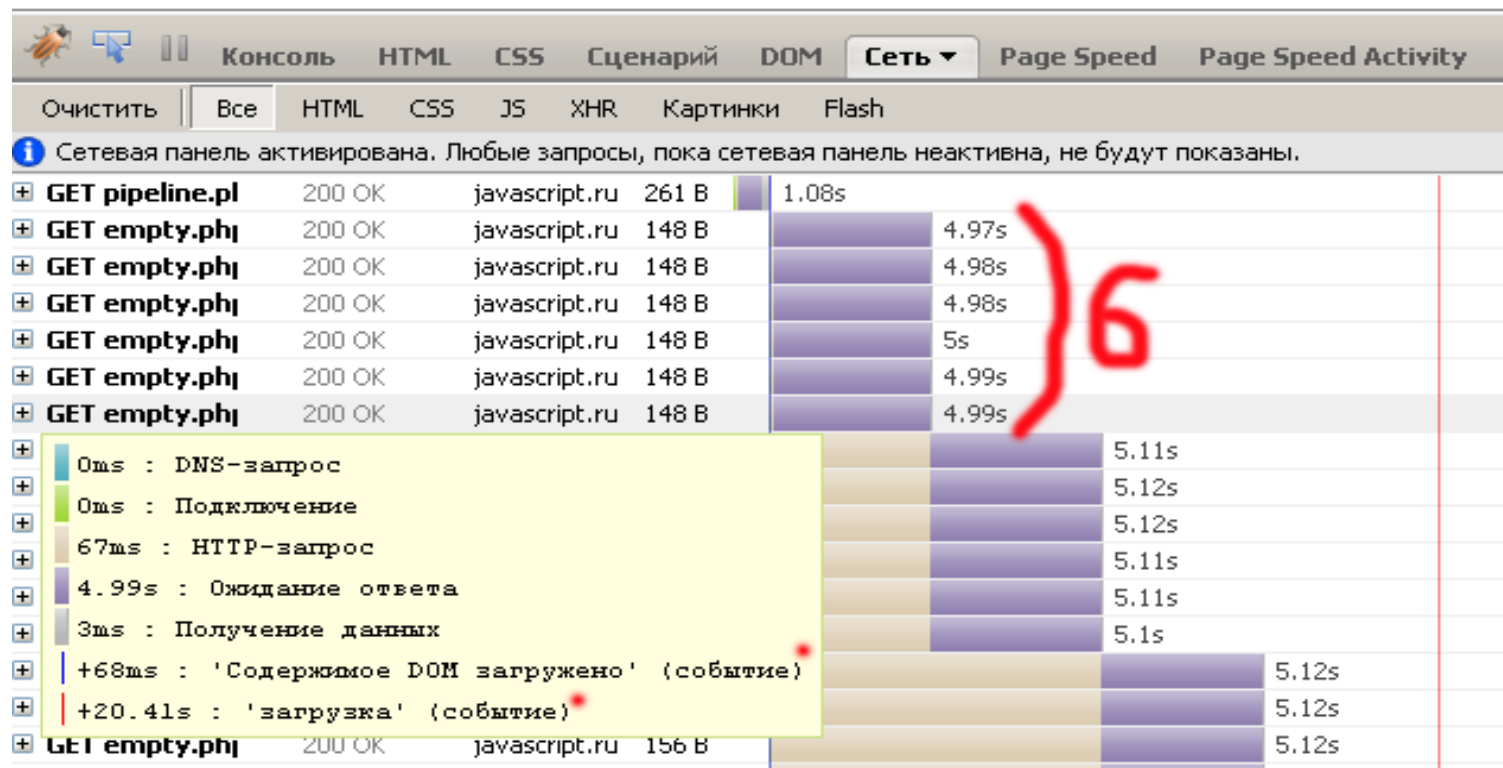
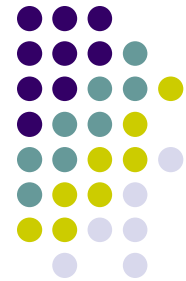


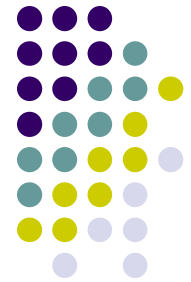
- Параллельная загрузка быстрее
  - Доказано FlashGet
    - Даже на модеме
    - Даже в 1990 годах
- Поэтому браузер грузит страничку параллельно

# HTTP Pipelining

## параллельная загрузка в браузере

Демо





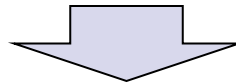
# HTTP Pipelining

## Кол-во потоков по браузерам

- ⑩ IE <8
  - 2 потока
- ⑩ IE 8+, FF 3.5+
  - 6 потоков
- ⑩ Opera 10
  - 8 потоков
- ⑩ Safari/Chrome
  - 6 потоков

... А еще браузеры можно ПОТЮНИТЬ...

<code>network.http.pipelining</code>	по умолчанию	логическое	false
<code>network.http.pipelining.maxrequests</code>	по умолчанию	целое	4
<code>network.http.pipelining.ssl</code>	по умолчанию	логическое	false
<code>network.http.proxy.pipelining</code>	по умолчанию	логическое	false



<code>network.http.pipelining</code>	установлено п...	логическое	true
<code>network.http.pipelining.maxrequ...</code>	установлено п...	целое	16
<code>network.http.pipelining.ssl</code>	установлено п...	логическое	true
<code>network.http.proxy.pipelining</code>	установлено п...	логическое	true



# Скрипты и стили

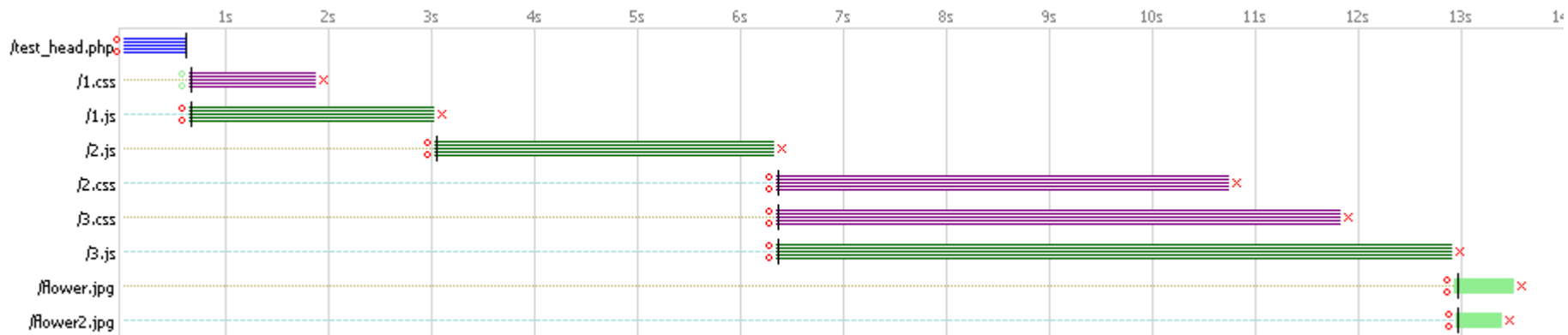
## ⑩ Тэг `<script>`

- во время загрузки скрипта
  - блокирует вообще все загрузки (IE<8, Opera)
  - блокирует все, кроме стилей и других скриптов
- во время выполнения
  - блокирует все

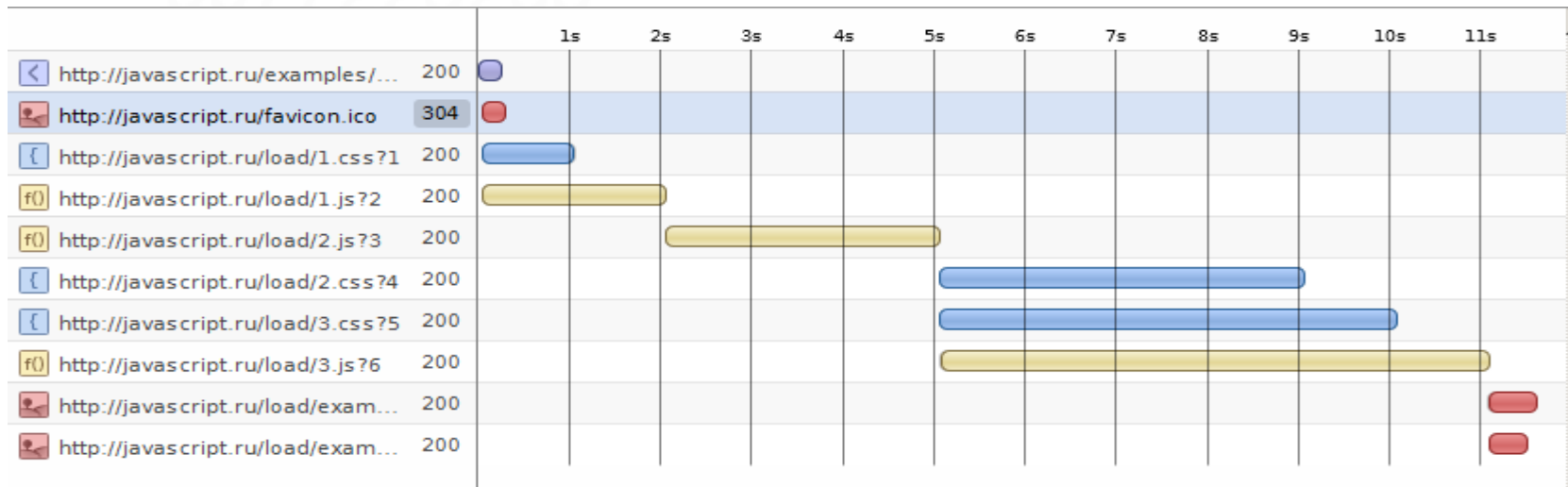
## ⑩ Типичная страничка

- Speculative parsing

# IE 6,7 (VM + Fiddler)



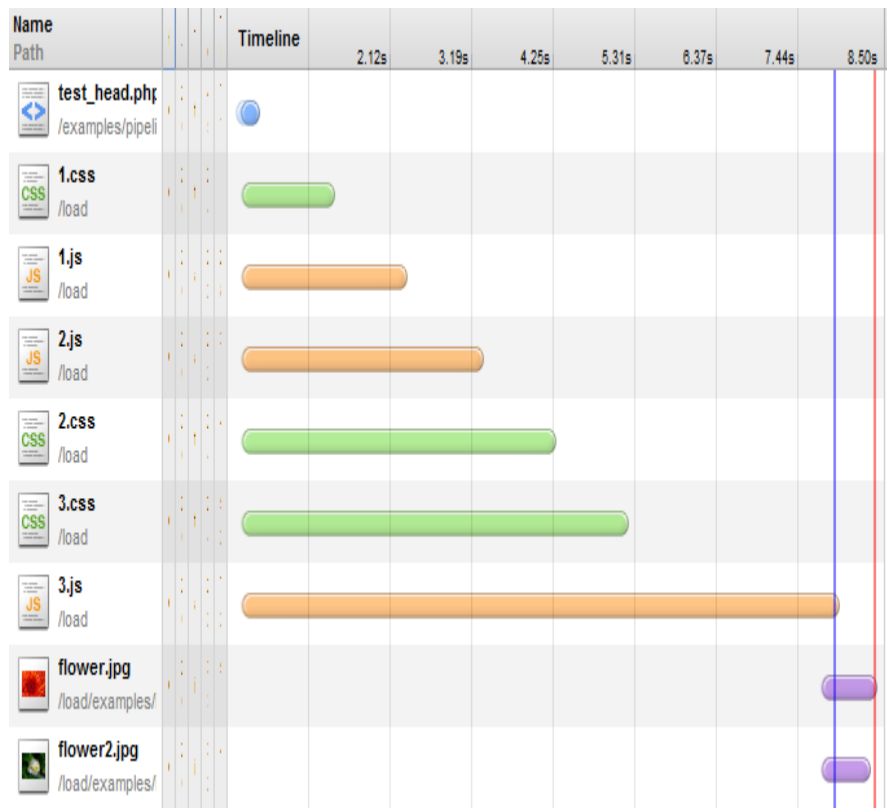
# Opera 11.50



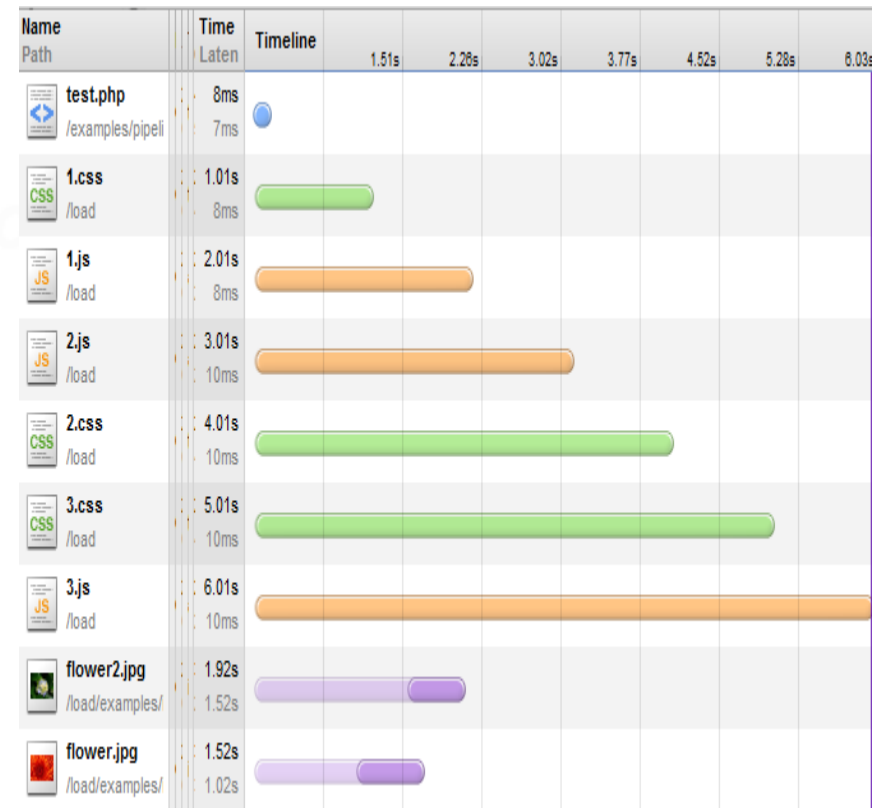


# Chrome 14, Safari 5.1

## Демо: Скрипты/CSS в HEAD



## Демо: Скрипты/CSS в BODY



# Firefox 4-6



URL	Статус	Домен	Временная линия
+ GET test.php	200 OK	javascript.ru	6ms
+ GET 1.css?1	200 OK	javascript.ru	1.01s
+ GET 1.js?2	200 OK	javascript.ru	2.01s
+ GET 2.js?3	200 OK	javascript.ru	3.01s
+ GET 2.css?4	200 OK	javascript.ru	4.01s
+ GET 3.css?5	200 OK	javascript.ru	5.06s
+ GET 3.js?6	200 OK	javascript.ru	6.03s
+ GET flower.jpg?0.5	200 OK	javascript.ru	505ms
+ GET flower2.jpg?0.	200 OK	javascript.ru	417ms

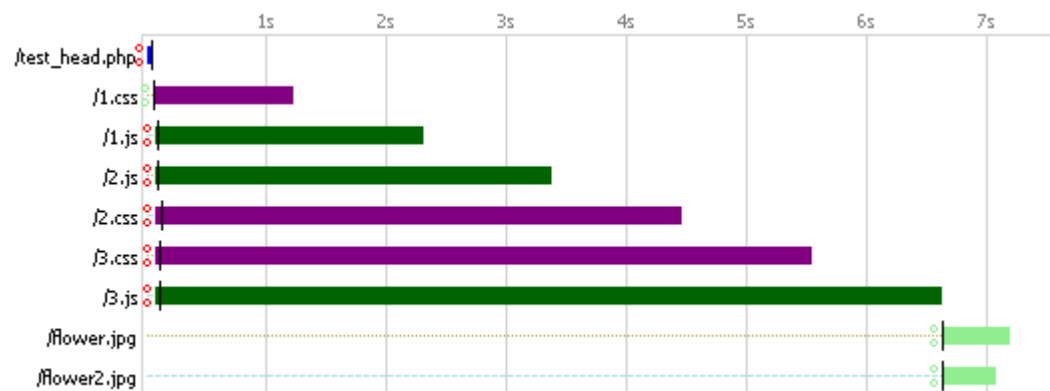




# IE 9

URL-адрес	Затрач...	Расписание
http://javascript.ru...	16 мс	
/load/1.css?1	1.12 с	
/load/1.js?2	2.18 с	
/load/2.js?3	3.16 с	
/load/2.css?4	4.11 с	
/load/3.css?5	5.11 с	
/load/3.js?6	6.11 с	
/load/examples/img...	1.62 с	
/load/examples/img...	2.02 с	

# IE 8 (VM + Fiddler)





## ...Доки в сети? (Google)

- 10 <http://code.google.com/intl/ru-RU/speed/page-speed/docs/rtt.html>

```
<head>
<link rel="stylesheet" type="text/css" href="stylesheet1.css" />
<script type="text/javascript" src="scriptfile1.js" />
<script type="text/javascript" src="scriptfile2.js" />
<link rel="stylesheet" type="text/css" href="stylesheet2.css" />
<link rel="stylesheet" type="text/css" href="stylesheet3.css" />
</head>
```

Assuming that each one takes exactly 100 milliseconds to download, that the browser can't get information about this, see [Parallelize downloads across hostnames](#)), and that the cache is



Только для Opera / IE<8 [Демо](#)



## ...Доки в сети? (Google)

⑩ <http://code.google.com/intl/ru-RU/speed/page-speed/docs/rtt.html>

Another, more subtle, issue is caused by the presence of an *inline* script following a stylesheet, such as the following:

```
<head>
<link rel="stylesheet" type="text/css" href="stylesheet1.css" />
<script type="text/javascript">
  document.write("Hello world!");
</script>
<link rel="stylesheet" type="text/css" href="stylesheet2.css" />
<link rel="stylesheet" type="text/css" href="stylesheet3.css" />
<link rel="alternate" type="application/rss+xml" href="front.xml" title="Say hello" />
<link rel="shortcut icon" type="image/x-icon" href="favicon.ico">
</head>
```

In this case, the reverse problem occurs: the first stylesheet actually blocks the inline script from being executed, which then in turn blocks other resources from being downloaded.

Только в IE<8, Опера [Демо](#)



## ...Доки в сети? (Yahoo)

⑩ <http://developer.yahoo.com/performance/rules.html>

An alternative suggestion that often comes up is to use deferred scripts. The `DEFER` attribute indicates that the script does not contain `document.write`, and is a clue to browsers that they can continue rendering. Unfortunately, Firefox doesn't support the `DEFER` attribute. In Internet Explorer, the script may be deferred, but not as much as desired. If a script can be deferred, it can also be moved to the bottom of the page. That will make your web pages load faster.

*smiroshnick@gmail.com*

*0677223233*

Firefox поддерживает `defer`, начиная с версии 3.1



## ...Доки в сети? (webo.in)

- ⑩ <http://webo.in/articles/habrahabr/56-non-blocking-javascript/>
  - Это перевод
  - Очень устарело и, кстати...

IE поддерживает ~~замечательный атрибут для тега <script>: defer.~~ Присутствие этого атрибута указывает IE, что загрузку скрипта нужно отложить до тех пор, пока не загрузится DOM. Однако, это работает только для внешних скриптов. ~~Следует также заметить, что этот атрибут нельзя выставлять, используя другой скрипт. Это означает, что нельзя создать <script> с этим атрибутом, используя DOM-методы, — атрибут будет просто проигнорирован.~~

Нет. Загрузка начинается тут же.



# Актуальные доки в сети

- 10 <http://www.browserscope.org/?category=network&v=top>
- Один из лучших ресурсов по этой теме

Top Browsers		Connections per Hostname	Max Connections	Script Script	Script Stylesheet	Script Image	Script Iframe	Async Scripts
name	score							
<input type="checkbox"/> Android 2.2 →	9/16	4	4	yes	yes	yes	no	no
<input type="checkbox"/> Chrome 6 →	11/16	6	30	yes	yes	yes	no	no
<input type="checkbox"/> Chrome 7 →	11/16	6	30	yes	yes	yes	no	no
<input checked="" type="checkbox"/> Firefox 3.6 →	14/16	6	30	yes	yes	yes	no	yes
<input type="checkbox"/> Firefox 4.0 →	14/16	6	30	yes	yes	yes	no	yes

iframe и скрипт не параллелятся?

Не совсем так

Разная очередь!

Демо



# ПОДКЛЮЧЕНИЕ СКРИПТОВ

*smiroshnick@gmail.com*

*0677223233*



# Roadmap

1. DOM
2. `document.write('script')`
3. `defer`
4. `async`
5. `xhr`





# Подключение скрипта

## Script DOM

### Демо

```
<script>
function addScriptDom(src) {
    var elem = document.createElement('script');
    elem.src = src
    document.getElementsByTagName('head')[0].appendChild(elem)
}

addScriptDom('jquery.min.js')
addScriptDom('plugin.js')
</script>
<script> $.plugin() </script>
```

- FF4+, IE, Safari/Chrome
  - все параллельно
  - не гарантируют порядок
- Opera:
  - гарантирует порядок выполнения
  - блокируется все, кроме JS/CSS

# Script DOM

## кросс-браузерные тонкости



### Демо

```
<script>
function addScriptDOM(src) {
    var elem = document.createElement('script');
    elem.src = src
    document.getElementsByTagName('head')[0].appendChild(elem)
}
</script>
<script> addScriptDOM('jquery.js') </script>
...
<script> addScriptDOM('plugin.js') </script>
```




Опера заблокируется



# Script DOM + CSS

Инлайн-скрипт ожидает загрузки CSS (кроме Opera)

```
<link rel="stylesheet" type="text/css" href="main.css" />
<script>
addScriptDOM('jquery.min.js')
addScriptDOM('plugin.js')
</script>
```



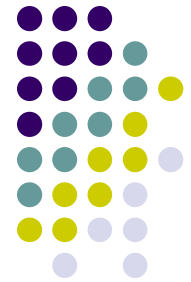
all_parallel.	200 OK	javascript.ru	619 B	1.04s
main.css	200 OK	javascript.ru	35 B	1.04s
jquery.min.	200 OK	javascript.ru	22 KB	1.14s
plugin.js	200 OK	javascript.ru	81 B	1.28s

[Демо](#)

Если скрипты по важности как CSS:

```
<script>
addScriptDOM('jquery.min.js')
addScriptDOM('plugin.js')
</script>
<link rel="stylesheet" type="text/css" href="main.css" />
```

[Демо](#)



# document.write('script')

## Демо

```
<script>
  document.write('<script src="'+src+'"></scr' + 'ipt>')
</script>
```

- ⑩ Лечит IE<8, Opera
  - Такой скрипт не блокирует js/css
  - Сохраняет порядок



# Script Defer

## Демо

```
<script defer src="jquery.min.js"></script>
<script defer src="plugin.js"></script>
```

- ⑩ IE4, Firefox 3.1, Chrome 8, Safari 5.1
- ⑩ Не блокирует рендеринг и другие скрипты
- ⑩ Гарантирует порядок выполнения
  - buggy в IE < 10: [Демо](#)  
innerHTML из первого скрипта заставляет браузер выполнять второй
- ⑩ Может выполняться до или после onDomContentLoaded
- ⑩ [По стандарту](#) только для внешних скриптов
  - **требует src**

```
<script defer>
// 
</script>
```



# Script Async /HTML 5/

## Демо

```
<script async src="jquery.js"></script>
```

- ⑩ FF 3.5, Chrome 8, Safari 5.1, IE 10
- ⑩ Ничего не блокирует
- ⑩ Не гарантирует порядок выполнения
  - В отличие от defer
  - [Демо](#)
  - [Стандарт](#)



# XMLHttpRequest

```
function addScriptXhr(file) {  
    var xhr = getXmlHttp()  
    xhr.open("GET", file, true)  
    xhr.onreadystatechange = function(){  
        if (xhr.readyState != 4) return  
        if (xhr.status >= 200 && xhr.status < 300 || xhr.status == 304){  
            evalGlobal(xhr.responseText)  
        }  
    }  
    xhr.send(null);  
}
```

- 10 Полностью кросс-браузерный способ
- 10 Ничего не блокирует
- 10 Полный контроль над выполнением и ошибками
- 10 Нет индикации загрузки браузером



# Обнаружение конца загрузки

## ⑩ Связывание

- Без обработки ошибок
- [Демо](#)

```
if (scriptElem.readyState){ //IE
    scriptElem.onreadystatechange = function(){
        if (this.readyState.match(/loaded|complete/)) {
            this.onreadystatechange = null
            setTimeout(callback,0)
        }
    }
} else {
    scriptElem.onload = callback
}
```

## ⑩ С проверкой ошибок (для сервисов)

- [Дока + Демо](#)

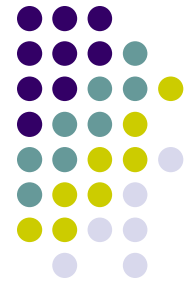
## ⑩ API: библиотека LABjs

- <http://labjs.com/>
- умеет загружать скрипты, не выполняя их





# DomContentLoaded



# Инициализация JS

## DOMContentLoaded блокируют:



- `<script>`, `document.write` (все браузеры)
- Opera – любая вставка скрипта
- Safari/Chrome – `defer`

Тест времени срабатывания `DOMContentLoaded`



# Инициализация CSS

## DomContentLoaded:

- ⑩ По HTML5 не требует стилей
- ⑩ Но `<script>...</script>` требует стилей
  - Везде, кроме Opera
  - ➔ Поэтому стили могут блокировать DomContentLoaded



[Демо](#)



# Еще о DomContentLoaded

- ⑩ DOMContentLoaded и Firefox
  - В FF/Chrome формы автозаполняются по DOMContentLoaded
  - [Демо](#)
  - лучше не задерживать DOMContentLoaded
- ⑩ DOMContentLoaded VS «init в конце body»
  - Разницы почти никакой, кроме IE6
  - В IE6 при init нельзя body.appendChild
- ⑩ Ресурсы
  - [DomContentLoaded в стандарте HTML5](#)
  - <http://molily.de/weblog/domcontentloaded>



# Рендеринг - не блокируют ...

- ⑩ FF – defer/async/dom
- ⑩ IE – dom/async(10)/defer
- ⑩ Opera – n/a
- ⑩ Safari/Chrome – dom/defer/async

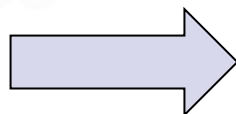
Тест (при отсутствии блокировки текст появляется сразу)



# Постепенное улучшение



# 1. Лоадер: [market.yandex.ru](http://market.yandex.ru)



все **современные** в продаже

Цена

от  до  руб.



Популярные производители | все

- |                                    |                                  |
|------------------------------------|----------------------------------|
| <input type="checkbox"/> BDK       | <input type="checkbox"/> Pioneer |
| <input type="checkbox"/> Hyundai   | <input type="checkbox"/> Rolsen  |
| <input type="checkbox"/> LG        | <input type="checkbox"/> Samsung |
| <input type="checkbox"/> Mystery   | <input type="checkbox"/> Sony    |
| <input type="checkbox"/> Panasonic | <input type="checkbox"/> SUPRA   |
| <input type="checkbox"/> Philips   | <input type="checkbox"/> Toshiba |

► Тип ?

☐ Поддержка Blu-Ray 3D ?

► Запись на DVD ?

► Портативный DVD ?

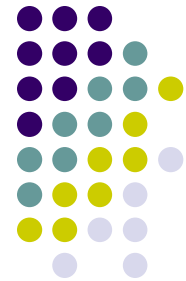
► Диагональ экрана ?

☐ Караоке ?

☐ Интерфейс USB Type A ?

☐ Выход HDMI ?

☐ Масштабирование до разрешения HDTV ?



## 2. Выцветание momondo.com

**Сравнение цены на авиабилеты**

Из:  ▼

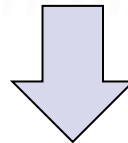
Дата вылета:

В:  ▼

Дата возвращения:

☐ → В один конец  
☐ ↔ Туда и обратно  
☐ ⤿ Составной маршрут

**Искать рейсы**



**Сравнение цены на авиабилеты**

Из:  ▼

Дата вылета:

В:  ▼

Дата возвращения:

☐ → В один конец  
☐ ↔ Туда и обратно  
☐ ⤿ Составной маршрут

**Искать рейсы**





### 3. Появление: google.ru

**Найти страницы**

со словами:

со словосочетанием:

с любым из слов:  OR  OR

**Не показывать страницы**

с любым из слов:

**Дополнительные настройки**

Отображение результатов:  Настройка недоступна, когда включен [Живой поиск Google](#).

Найти страницы на:

Найти файлы в формате:

Искать на сайте или в домене:   
(например, ru.youtube.com или .edu)

[Дата, права использования, регион и другие сведения](#)

По мере заполнения формы ваш запрос будет отображаться в данном поле

**Найти страницы**

со словами:

со словосочетанием:  [Совет](#)

с любым из слов:  OR  OR  [Совет](#)

**Не показывать страницы**

с любым из слов:  [Совет](#)

**Дополнительные настройки**

Отображение результатов:  Настройка недоступна, когда включен [Живой поиск Google](#).

Найти страницы на:

Найти файлы в формате:

Искать на сайте или в домене:   
(например, ru.youtube.com или .edu)

[+ Дата, права использования, регион и другие сведения](#)



# Рецепт:

## Скрипты в конце BODY

1. «Все скрипты в конец body»
  - На одном (статик) домене
  - Минусы: одна очередь загрузки, ждет загрузки документа
  - [Демо](#)
2. «Все скрипты в конец body + другой домен»
  - Все равно ждет документа

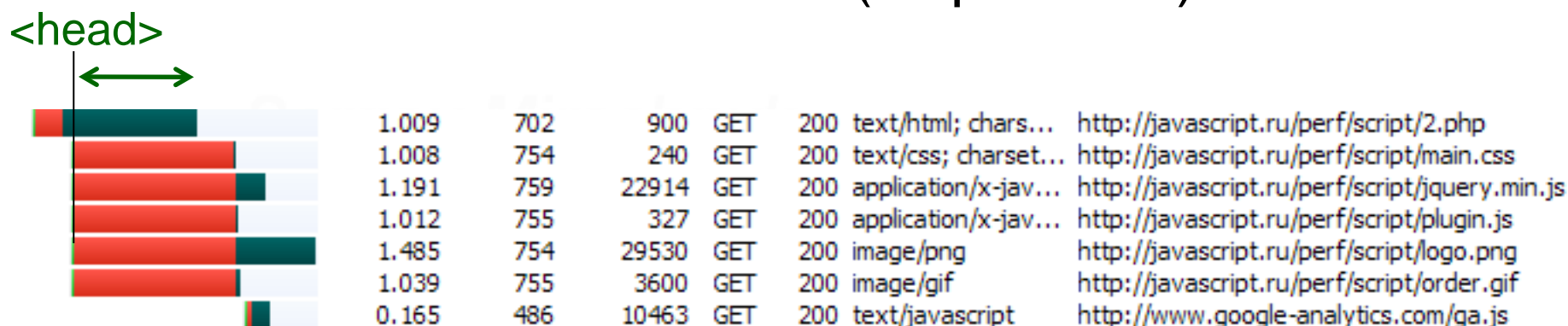
*smiroshnick@gmail.com*

*0677223233*



# Скрипты нужны как можно раньше!

- ⑩ Скрипты в HEAD
- ⑩ Demo: amazon.com (Http Watch)



Transfer-Encoding: chunked

Content-Encoding: gzip

Как добавлять в HEAD и не блокировать рендер? `dom/defer`

Как отслеживать окончание загрузки? `onload/onreadystatechange`

\*Опера – все не как у людей



# Скрипты после рендеринга

## 1. Bootstrap / Preload

- Заглушки интерфейсов
- Важнейшая инициализация сразу
  - логин
- API для отложенной инициализации
  - `onready( [bootstrap level] )`

## 2. В коде компонент:

- `onready(func)`, или
- `onready(bootstrap event, func)`

```
onready('jQuery', initSelect);
```

```
onready('searchAPI', initSearchForm);
```

**(сборка с учетом отдельной подгрузки скриптов)**



# Дополнительно

## 10 Conditional Comments (IE)



- Также блокируют: [Демо](#)
- [Лечение: <!-- \[if IE\]><!\[endif\]-->](#)
  - Пустой CC перед CSS

Sergey Miroshnyk  
smiroshnick@gmail.com  
0677223233



# Материалы

## 10 Книга

- Even Faster Web Sites (Steve Souders)

## 10 Записи

- <http://blogs.msdn.com/kristoffer/archive/2006/12/22/loading-javascript-files-in-parallel.aspx>
- <http://www.stevesouders.com/blog/2008/12/27/coupling-async-scripts/>
- <http://www.stevesouders.com/blog/2009/04/27/loading-scripts-without-blocking/>
- <http://www.stevesouders.com/blog/2009/05/06/positioning-inline-scripts/>
- <http://www.nczonline.net/blog/2009/06/23/loading-javascript-without-blocking/>
- <http://www.nczonline.net/blog/2009/07/28/the-best-way-to-load-external-javascript/>