

ИКТ В НОС

DOM

Тема №6



JS и браузер



Какво е BOM

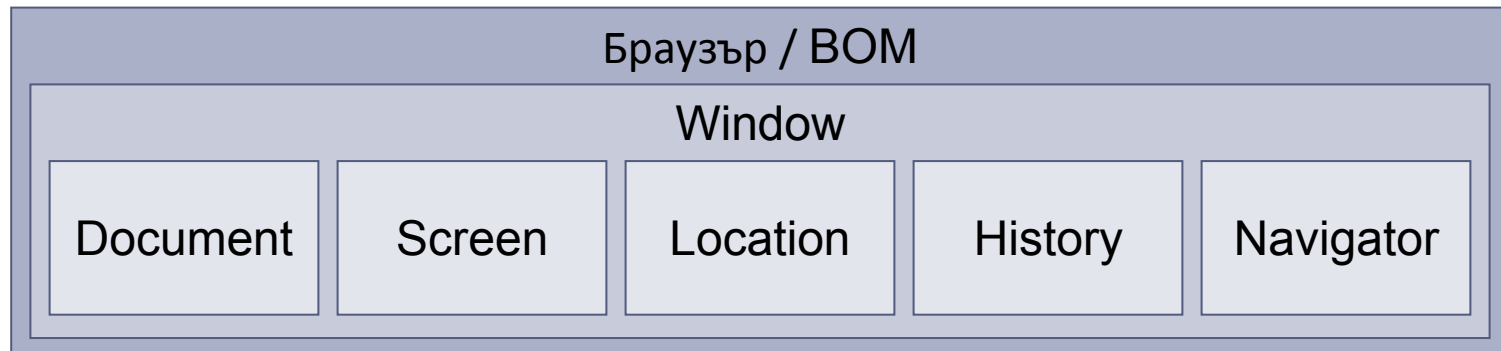
- Browser Object Model
- Програмен интерфейс към браузър
- Позволява достъп чрез JS до
 - Съдържанието на показваната страница
 - Параметрите на прозореца на браузъра
 - Историята на навигация, бисквитки и т.н.

Структура на BOM



Главен обект **window**

- Съдържание на страницата в обект **document**
- Свойства на прозореца в обект **screen**
- Текущ адрес и история в обекти **location** и **history**
- Настройки на браузъра в обект **navigator**



Още за ВОР

- Формално няма стандарт
- Неявно се поддържа от браузърите
- Някои функции могат да се различават

В курса

- Специално внимание на обекта `document`



DOM



Какво е DOM

- Document Object Model
- Модел за представяне на HTML документи чрез обекти
- Браузърите предоставят програмен интерфейс към DOM през обекта **window.document** на своя BOM

История на DOM



История

- 1995 – Legacy DOM / DOM level 0, липса на стандарт
- 1997 – Intermediate DOM, все още без стандарт
- 1998 – DOM Level 1 – първи стандарт
- 2000 – DOM Level 2
- 2004 – DOM Level 3
- 2014 – DOM Level 4

Какво може DOM



Възможности

- Достъп и промяна на HTML елементи
- Достъп и промяна на CSS стилове
- Достъп и промяна на събития, породени от работата на браузъра или действията на потребителя

Обаче

- DOM само позволява достъп и промяна
- Самият достъп и промяна се правят с код на JS

Структура на DOM



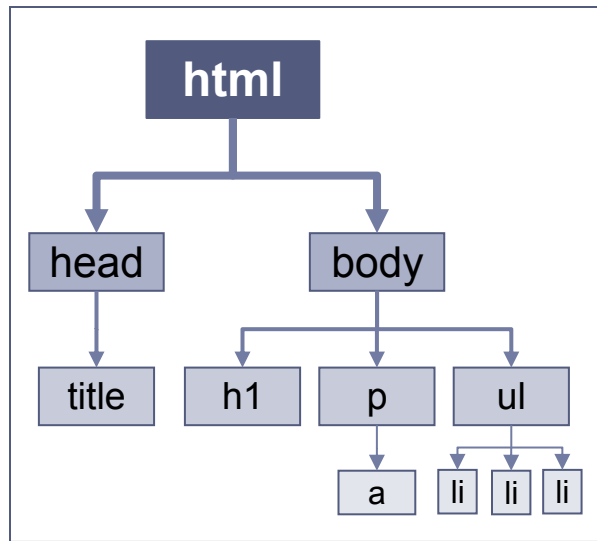
DOM през очите на JS

- Строга йерархия
- Обекти и свойства в дървовидна структура
- Коренът е „закачен“ към **window.document**
- Пълен достъп до всеки възел в дървото с възможност за промяна, добавяне и изтриване

Връзка с HTML и DOM

- Всеки елемент от HTML се представя като възел от DOM
- Браузърите предоставят средства за разглеждане на DOM структурата на страница

```
<html>
<head>
  <title>Пример 0601:
    HTML пример</title>
</head>
<body>
  <h1>Eleifend porttitor</h1>
  <p>Duis auctor <a href="#1">
    Fringilla</a></p>
  <ul>
    <li>Nulla</li>
    <li>Dapibus</li>
    <li>Ligula</li>
  </ul>
</body>
</html>
```



Пример 0601: HTML при x

← → ↺ 🔍

Eleifend porttitor

Duis auctor [Fringilla](#)

- Nulla
- Dapibus
- Ligula

🔍 📄 Elements » ⌵ ⚙️ 🖨️ ✕

<!DOCTYPE html>

▼ <html>

▼ <head>

<meta charset="utf-8">

<title>Пример 0601: HTML пример</title>

</head>

▼ <body>

<h1>Eleifend porttitor</h1>

▼ <p>

" Duis auctor "

Fringilla

</p>

▼

Nulla

Dapibus

Ligula

</body>

</html>

<!DOCTYPE>

Styles Event Listeners DOM Breakpoints »

Find in Styles Filter

Работа с DOM



Общ алгоритъм

- Намираме желания елемент или елементи
- Получаваме JS обекти със свойства атрибутите на елемент
- Променяме техните свойства

Синхронизация със зареждането

- Атрибут **onload** на елемента `body` – активира се при завършване на зареждането на страница



Намиране на елемент

Търсене по id

- Търсене по атрибут id с `getElementById`
- Резултатът е обект с намереният елемент или `undefined`
- Стилът на елемента е свойството `style` на обекта

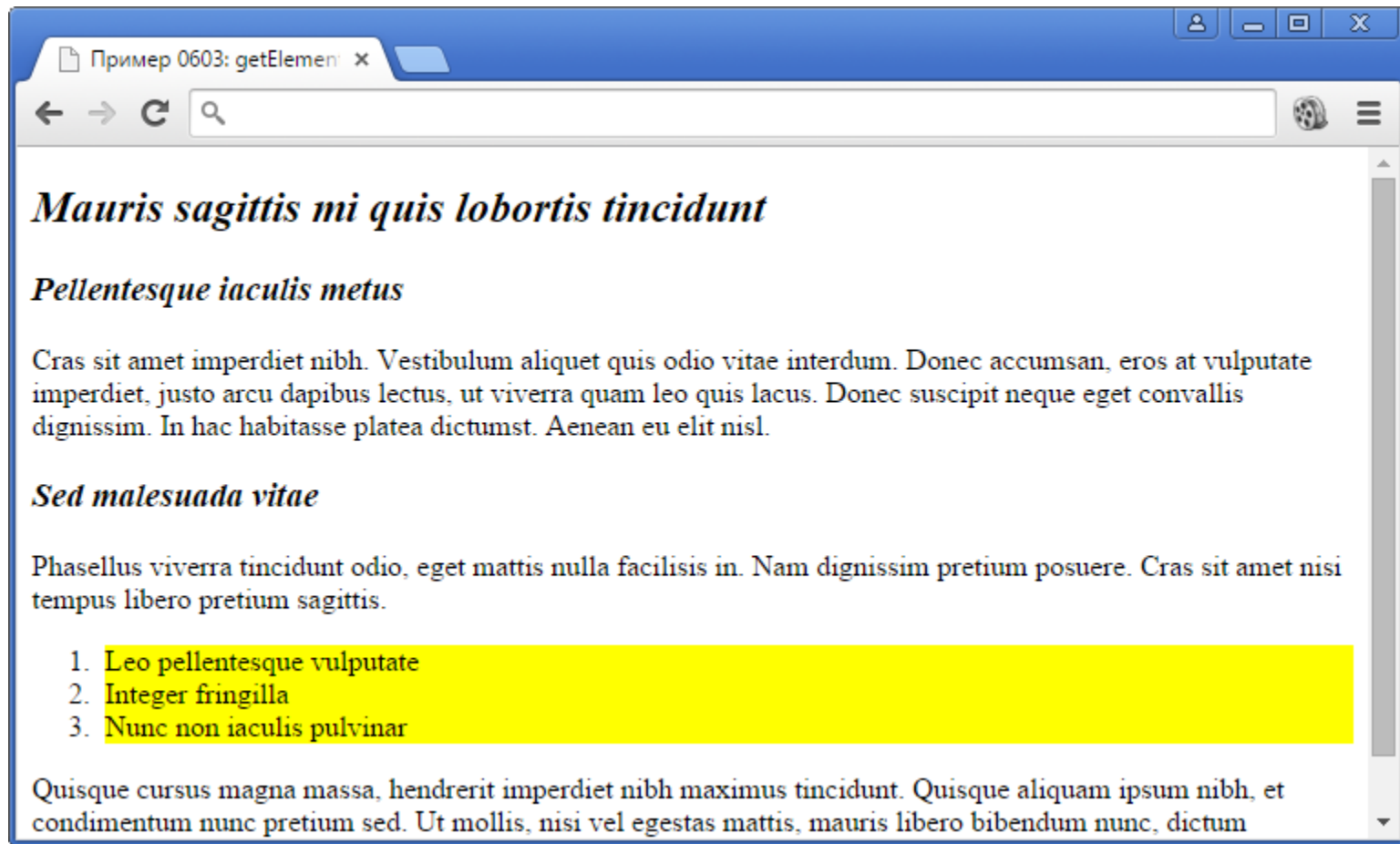
```
var e = document.getElementById('one');  
e.style.color = 'red';  
  
document.getElementById('three').style.color =  
                                '#0000FF';
```



Фамилия методи `getElements`

- Търсене по атрибут `class` с `getElementsByClassName`
- Търсене по тип на елемента с `getElementsByTagName`
- Резултатът е масив от намерените елементи

```
var li = document.getElementsByTagName('li');  
  
for (var e=0; e<li.length; e++)  
    li[e].style.backgroundColor = 'yellow';
```



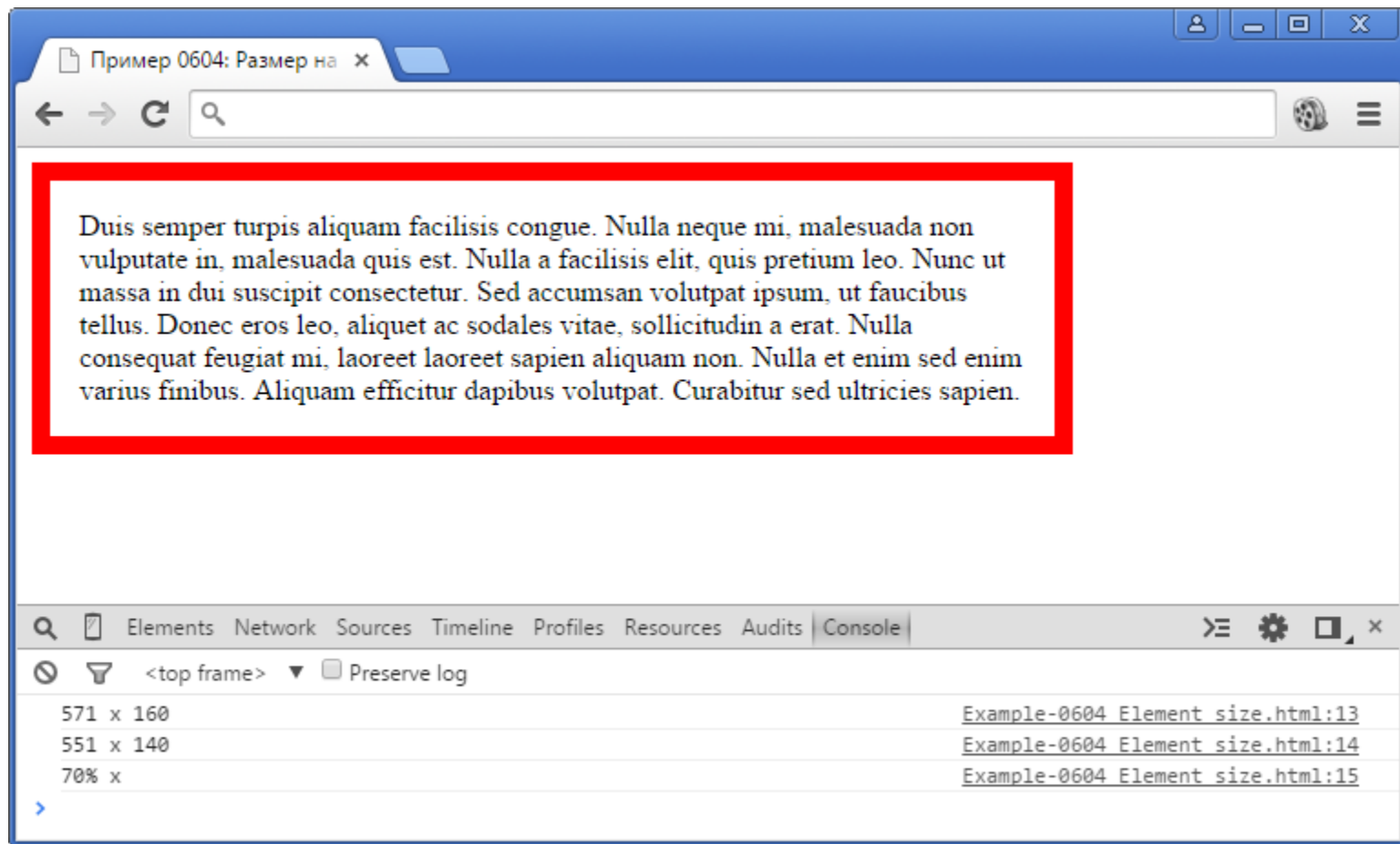
Оразмеряване на обекти



Намиране на размер

- С рамката – `offsetWidth` и `offsetHeight`
- Без рамката – `clientWidth` и `clientHeight`
- Според стила – `style.width` и `style.height`

```
var e = document.getElementById('box');  
console.log(e.offsetWidth+' x '+e.offsetHeight);  
console.log(e.clientWidth+' x '+e.clientHeight);  
console.log(e.style.width+' x '+e.style.height);
```



ПРОБА

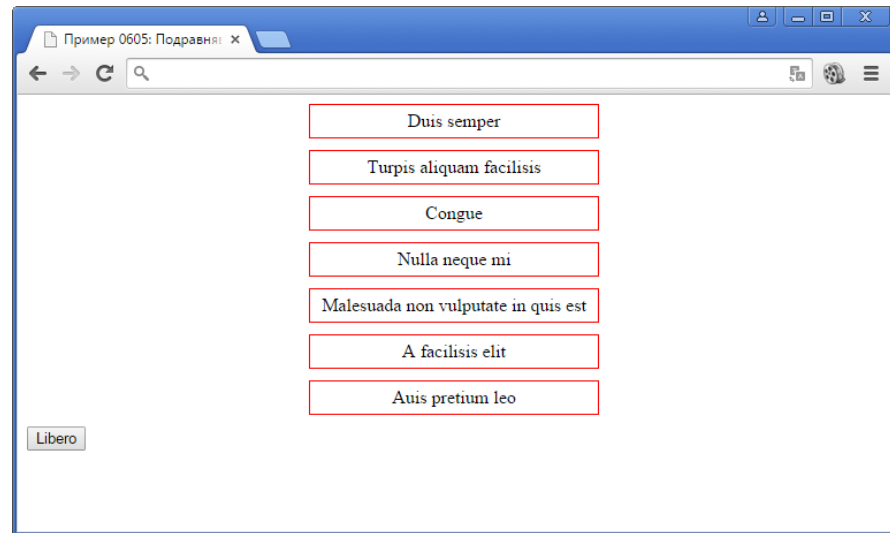
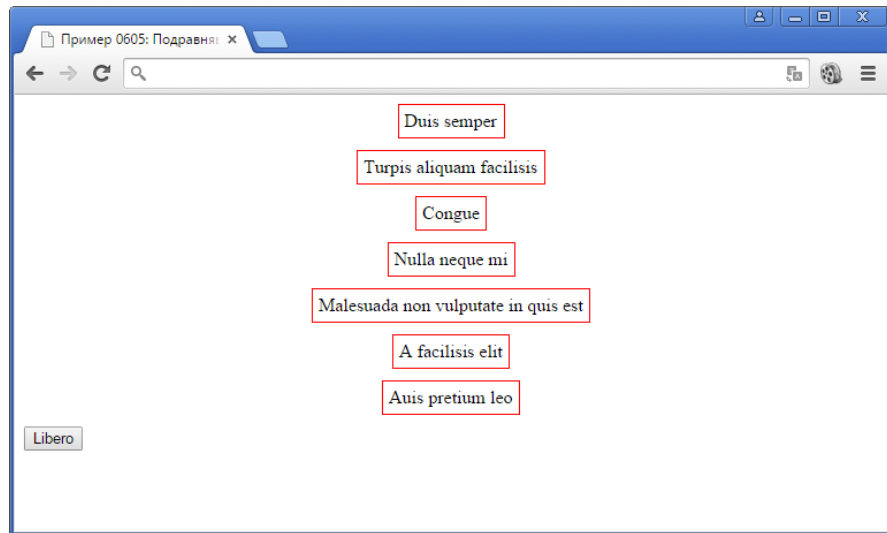
Промяна на размер

- Няколко текстови полета с различна ширина
- С натискане на бутон ги подравняваме по най-широкото
- Но `clientWidth` не може да се променя програмно, затова променяме `style.width`

```
var spans = document.getElementsByTagName('span');

var max = 0;
for (var i=0; i<spans.length; i++)
    max = Math.max(max, spans[i].clientWidth);

for (var i=0; i<spans.length; i++)
    spans[i].style.width = max+'px';
```



ПРОБА

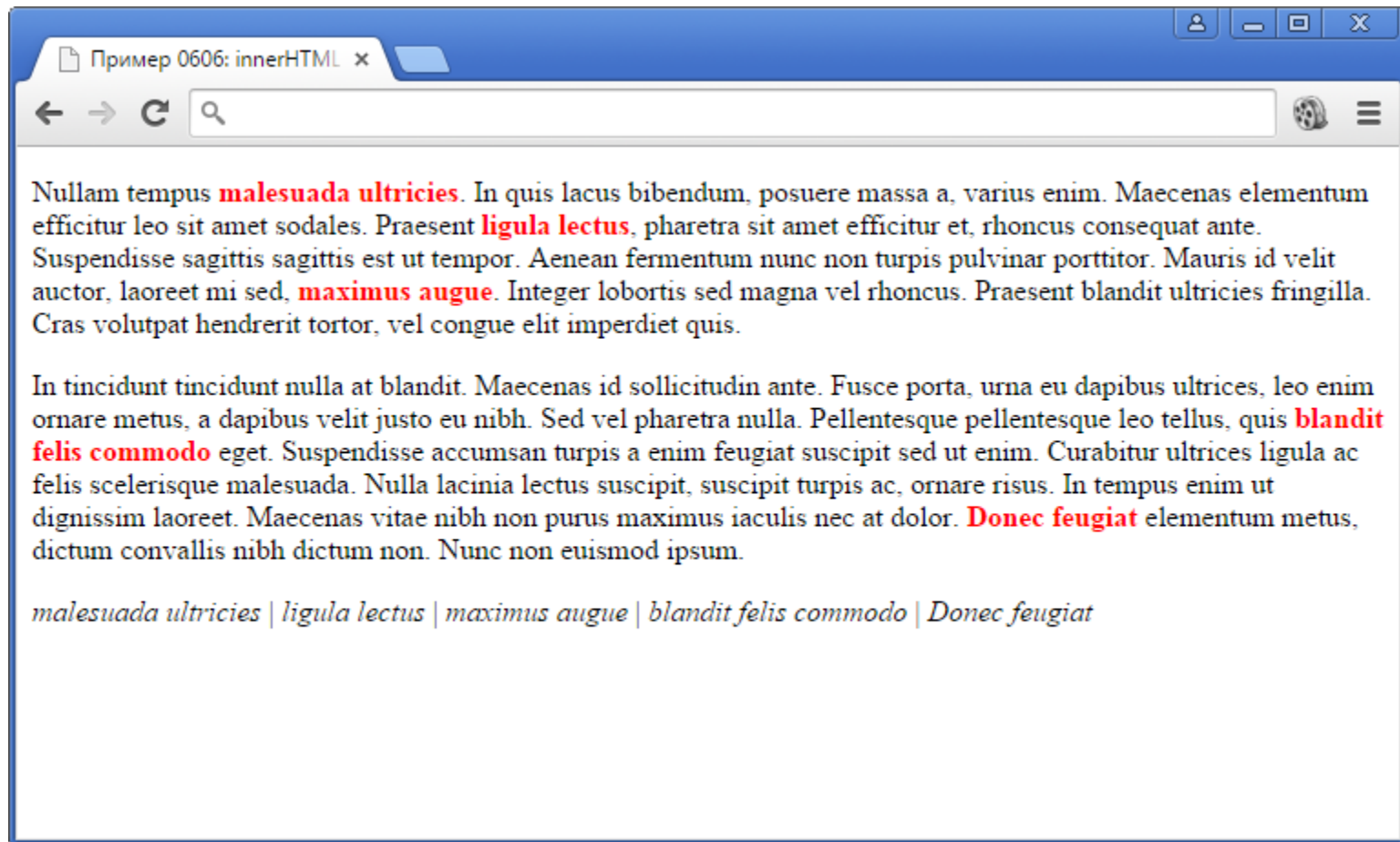
Съдържание на елемент



HTML съдържание

- Пълно съдържание на елемент като HTML текст с **innerHTML**
- Ако има вътрешни елементи, те са под формата на тагове
- Пример с извличане на съдържание от едни елементи и записването му в други

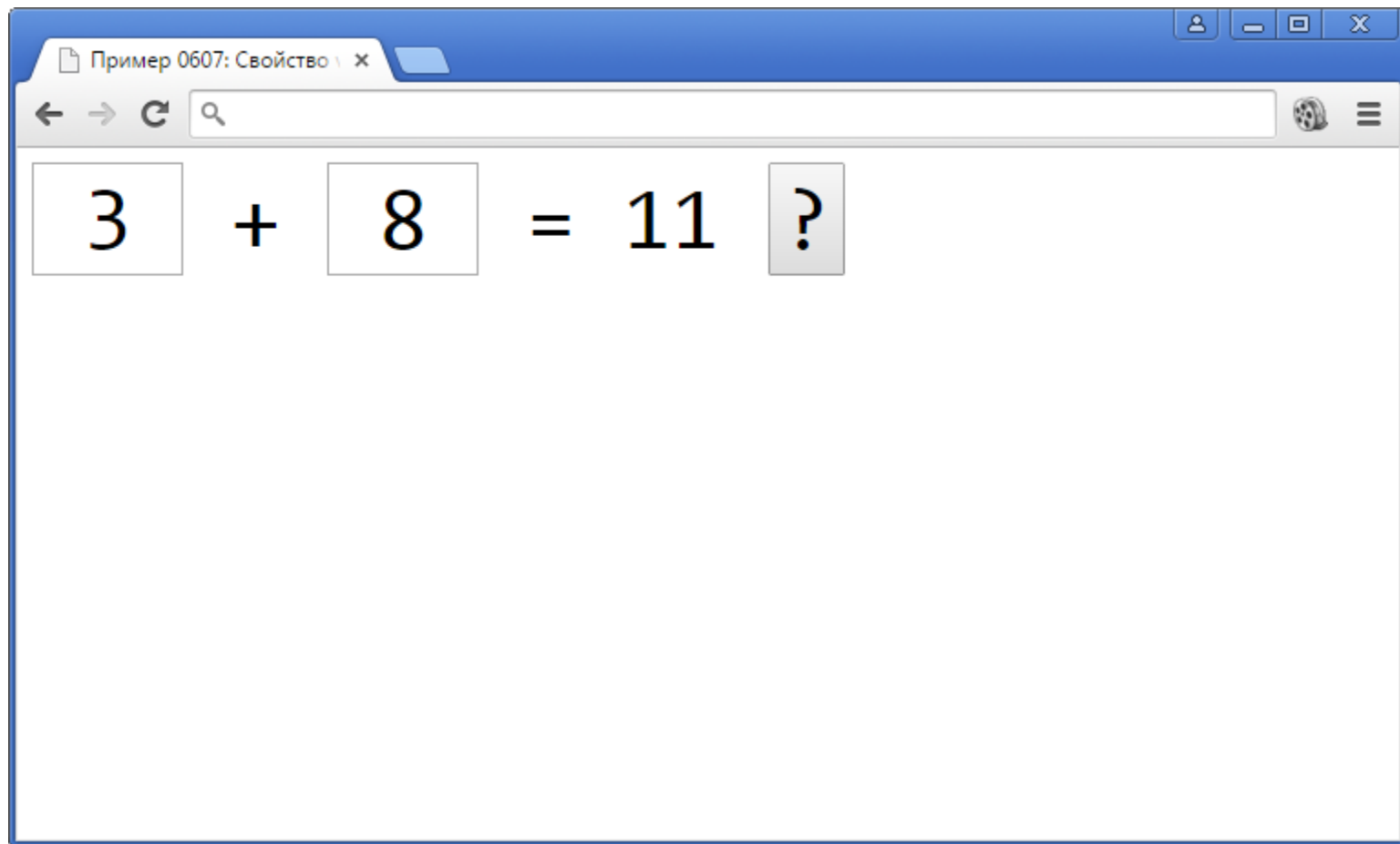
```
var q = '';  
var b = document.getElementsByTagName('b');  
for (var i=0; i<b.length; i++) q += b[i].innerHTML;  
  
document.getElementById('result').innerHTML = q;
```



Съдържание на текстово поле

- Показваният текст е в атрибута **value**
- Ако трябва да се ползва като число, трябва да преобразуваме с **Number**, **parseInt** или **parseFloat**

```
var n1=Number(document.getElementById('num1').value);  
var n2=Number(document.getElementById('num2').value);  
  
document.getElementById('res').innerHTML=n1+n2;
```



ПРОБА

Съдържание на списък

- Елементите са дефинирани с тагове **select** и **option**, като показваната и реалната стойност могат да са различни
- Изборът на елемент от списъка активира **onchange**

```
<select id="list" onchange="newSelection()">  
  <option value="Morbi ipsum primis">Morbi</option>  
  <option value="Vel orci luctus">Vel</option>  
</select>
```

- Достъпът до избраната стойност е през **value**

```
document.getElementById('list').value;
```



ПРОБА

Събития



Някои видове събития

- При движение с мишката
- При активиране на HTML елемент
- При работа с клавиатурата
- При промени в размера на прозореца
- При завършване на зареждането на страница

Свойства на събитията

- Където се е случило – това е HTML, DOM или BOM елемент
- Функция на JS, която трябва да го обработи
- Допълнителни параметри към събитието, които зависят от типа на събитието
- Повечето събития са характерни за много елементи и се ползват по един и същ начин
- Някои елементи имат специфични събития, само за тях

Често използвани събития

- Общи събития
 - `onload` – при пълно зареждане на страницата
 - `onresize` – при промяна на размера на страницата
- Събития към HTML елементи
 - `onclick` – при кликване върху елемент
 - `onchange` – промяна на съдържанието на елемент
 - `onmousemove` – при движение на мишката над елемент
 - `onkeypress` – при натискане на клавиш

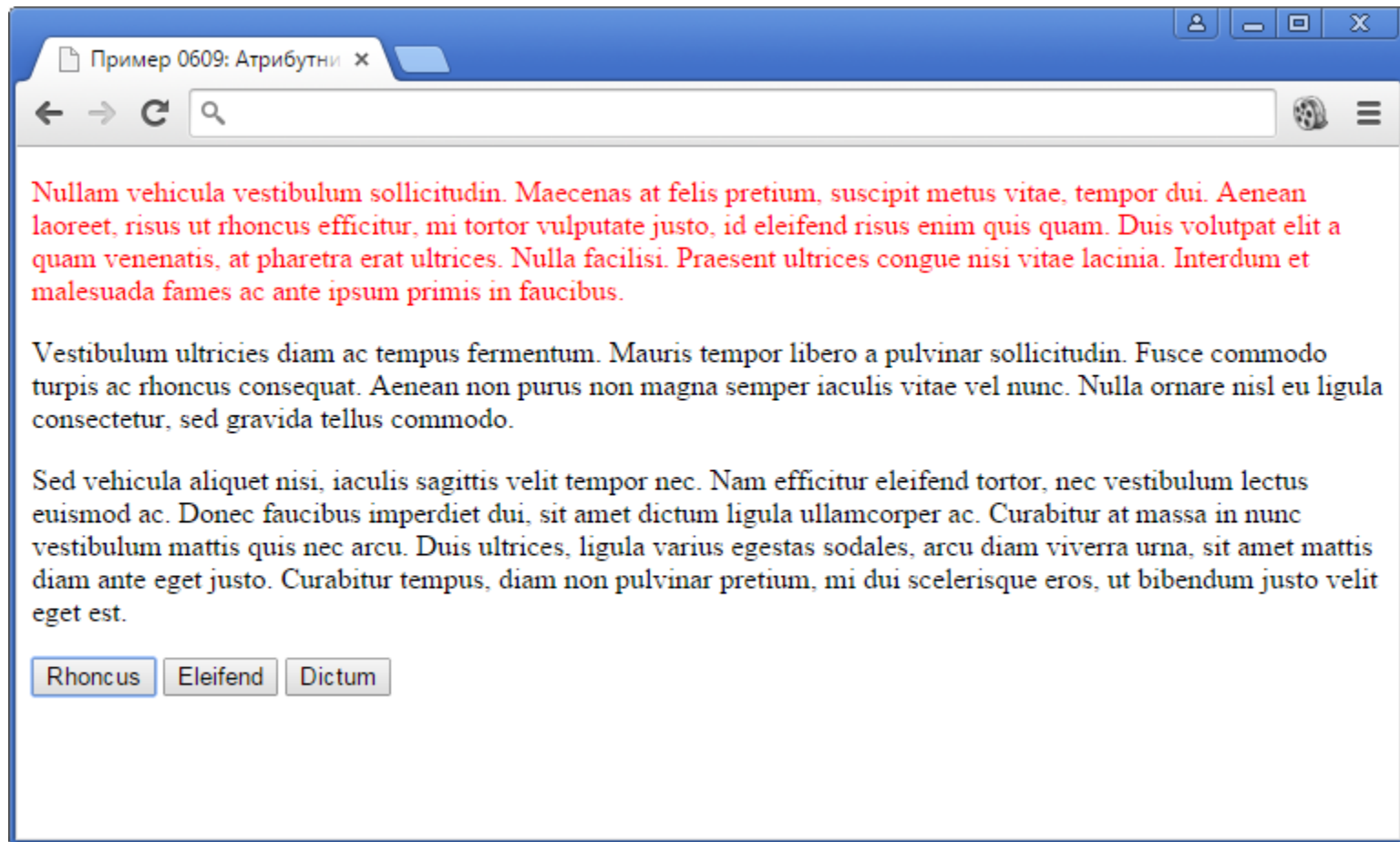
Улавяне на събития



Чрез атрибут на HTML елемент

- Името на атрибута е име на събитието
- Стойността на атрибута е JS код за изпълнение

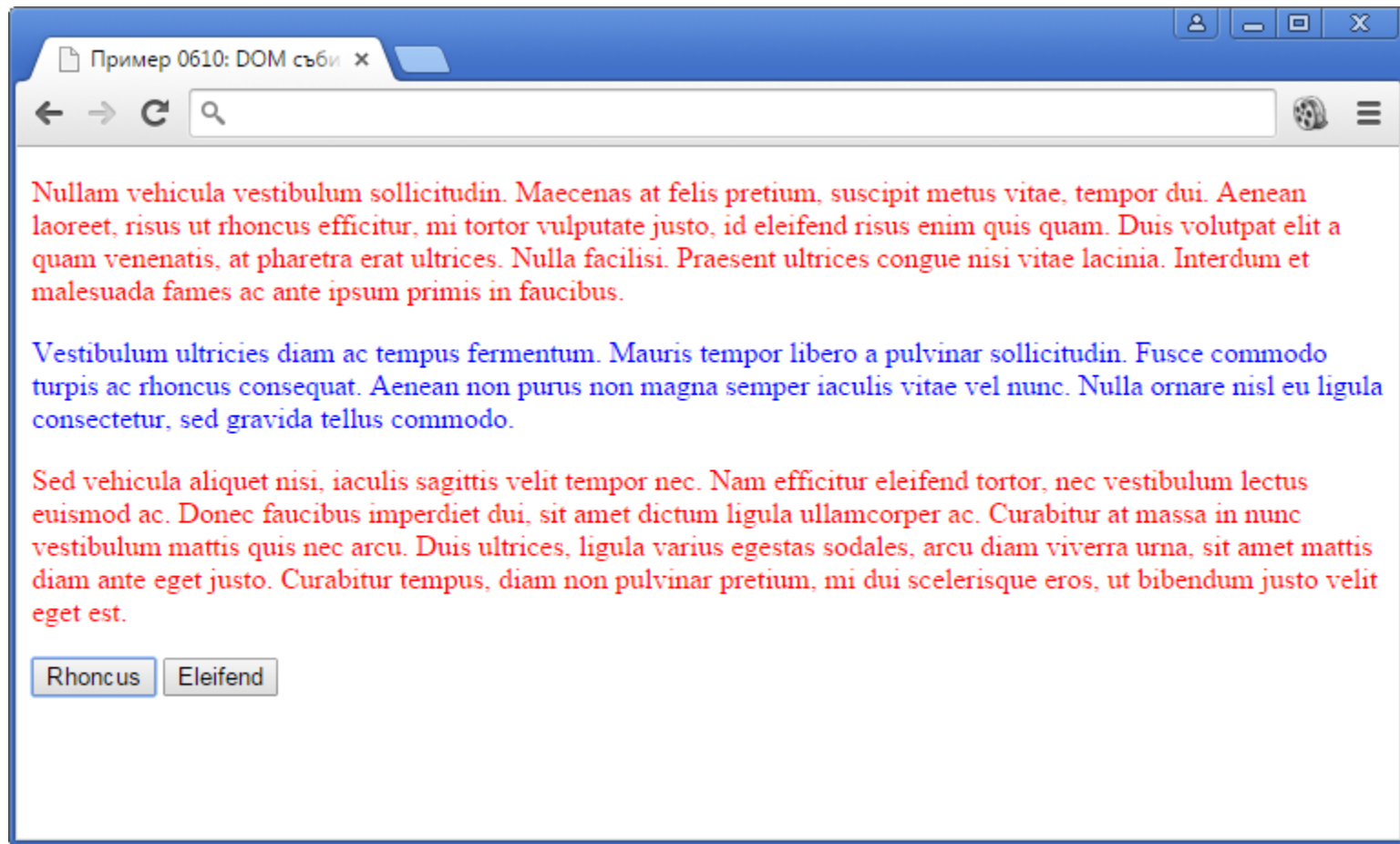
```
<button onclick="recolor(0)">Rhoncus</button>  
<button onclick="recolor(1)">Eleifend</button>  
<button onclick="recolor(2)">Dictum</button>
```



Чрез метод на DOM елемент

- Задаване на стойност на метод на елемент
- Чрез съществуваща или чрез анонимна функция

```
b = document.getElementById('but1');  
b.onclick = recolorAll;  
  
b = document.getElementById('but2');  
b.onclick = function ()  
{  
    var p;  
    p = document.getElementsByTagName('p');  
    p[1].style.color='blue';  
};
```

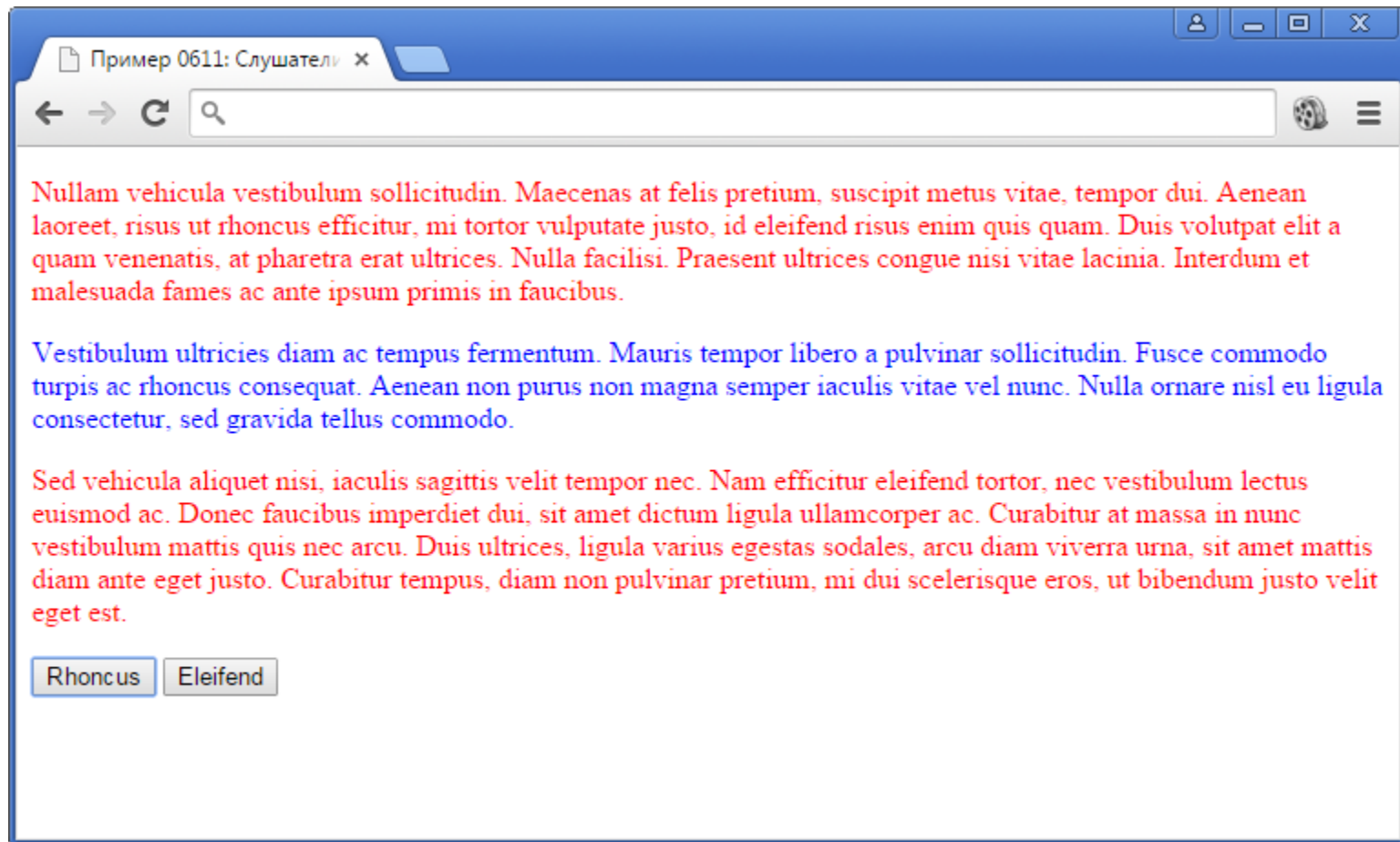


ПРОБА

Чрез „слушател на събития“

- Синтаксис `element.addEventListener(event,function)`
- Името на събитието е без представката on
- Функцията е или явна или анонимна

```
b = document.getElementById('but1');  
b.addEventListener('click',recolorAll);  
  
b = document.getElementById('but2');  
b.addEventListener('click',function ()  
{  
    var p = document.getElementsByTagName('p');  
    p[1].style.color='blue';  
});
```



Примери

Въпроси и отговори



Цел

- Списък от въпроси и техните отговори
- Първоначално отговорите са скрити
- При кликване върху въпрос се появява отговорът му
- При повторно кликване отговорът се скрива
- Броят въпроси и отговори не е твърдо фиксиран

Реализация на въпросите и отговорите

- Ще са в последователни двойки от h3 и p елементи

```
<h3>Q1. Fusce purus enim?</h3>  
<p>Scelerisque sit amet aliquet vitae, ... </p>  
<h3>Q2. Mauris eros?</h3>  
<p>Rutrum a elementum sed, volutpat a nisi. ...</p>
```

- Оформлението е с каскадни стилове, като за елемента h3 е зададен курсор като че ли е хипервръзка

```
h3 {  
    color: #000080;  
    padding: 0.3em;  
    cursor: pointer;  
}  
p {  
    margin: 0.25em auto 0.75em 2.5em;  
}
```

Първоначално скриване

- При зареждане на страницата се скриват всички отговори

```
<script>
  function main()
  {
    var p = document.getElementsByTagName('p');
    for (var i=0; i<p.length; i++)
      p[i].style.display = 'none';
  }
</script>

<body onload="main()">
</body>
```

Обработване на събитията

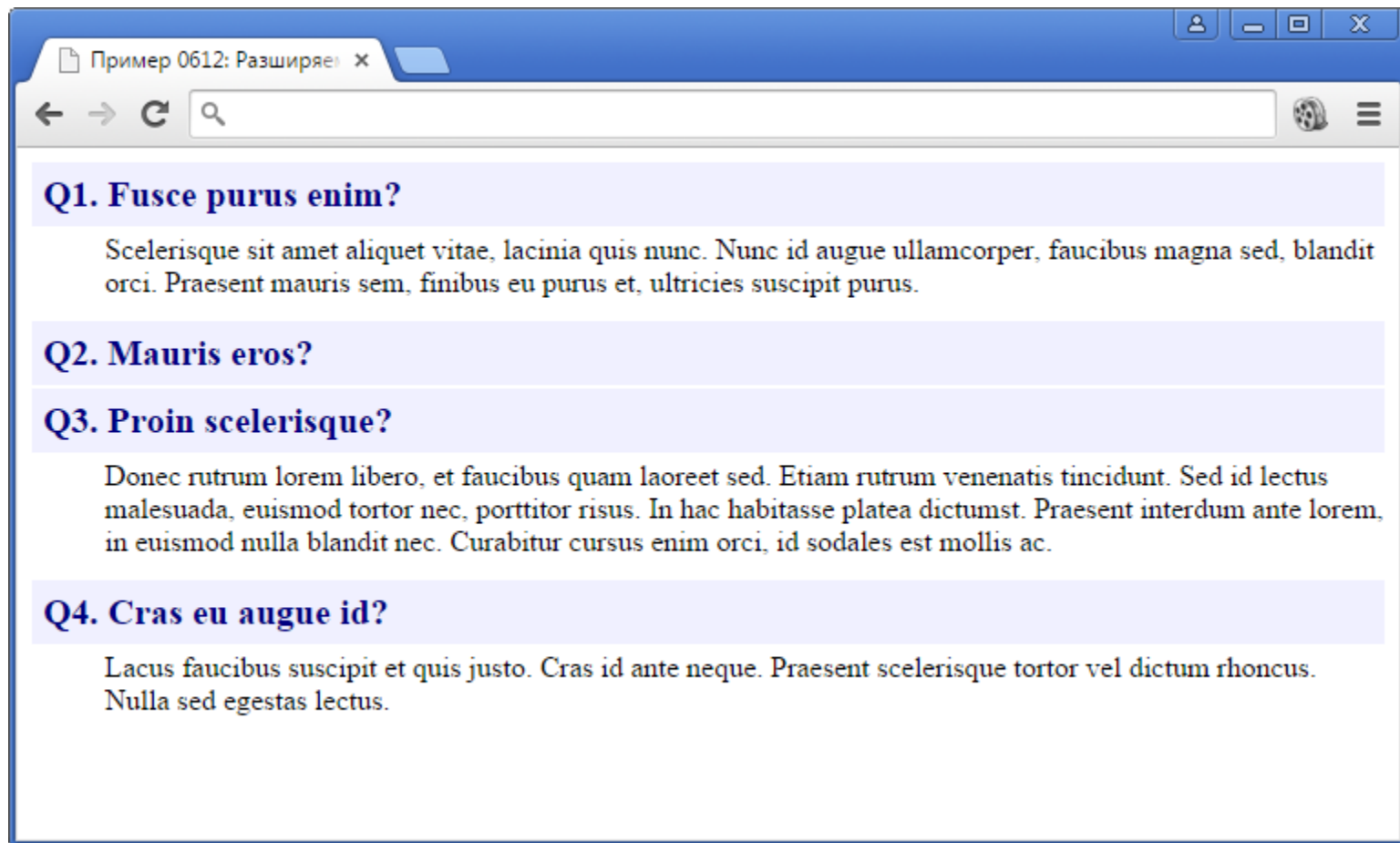
- Във всеки DOM елемент на въпрос запомняме връзка към съответния DOM елемент с отговора

```
function main()
{
    var h = document.getElementsByTagName('h3');
    var p = document.getElementsByTagName('p');

    for (var i=0; i<p.length; i++)
    {
        p[i].style.display = 'none';
        h[i].elem = p[i];
    }
}
```

- Създаваме слушател на **onclick**, който според видимостта на съответния отговор или го скрива, или го показва
- Обектът **event** съдържа данни за самото събитие
- Полето **target** е DOM елементът, където се е случило събитието, а **elem** е запомненият DOM елемент за скриване или показване

```
h[i].addEventListener('click',function(event)
{
    var style = event.target.elem.style;
    if (style.display=='none')
        style.display = 'block';
    else
        style.display = 'none';
});
```



Обобщение

Обектен модел на документ



Обектни модели

- На браузъра – BOM (Browser Object Model)
- На документа – DOM (Document Object Model)
- Програмен достъп с JS до страница и елементите ѝ
- Обектът `{window.document}` е корен на пълното дърво с HTML елементи на страница

Работа с елементи

- Търсене по идентификатор, клас или име на таг
- Намиране на конкретен елемент или на всички елементи от даден тип
- Оразмеряване на елементи
- Достъп до съдържание на елементи
- Реагиране на събития



Допълнителна информация

- Елементи и събития в DOM:

http://www.w3schools.com/jsref/dom_obj_document.asp

http://www.w3schools.com/jsref/dom_obj_event.asp

<http://www.w3schools.com/jsref>



ИКТ В НОС

Край

Коментари, въпроси