

JavaScript

DOM, Examples



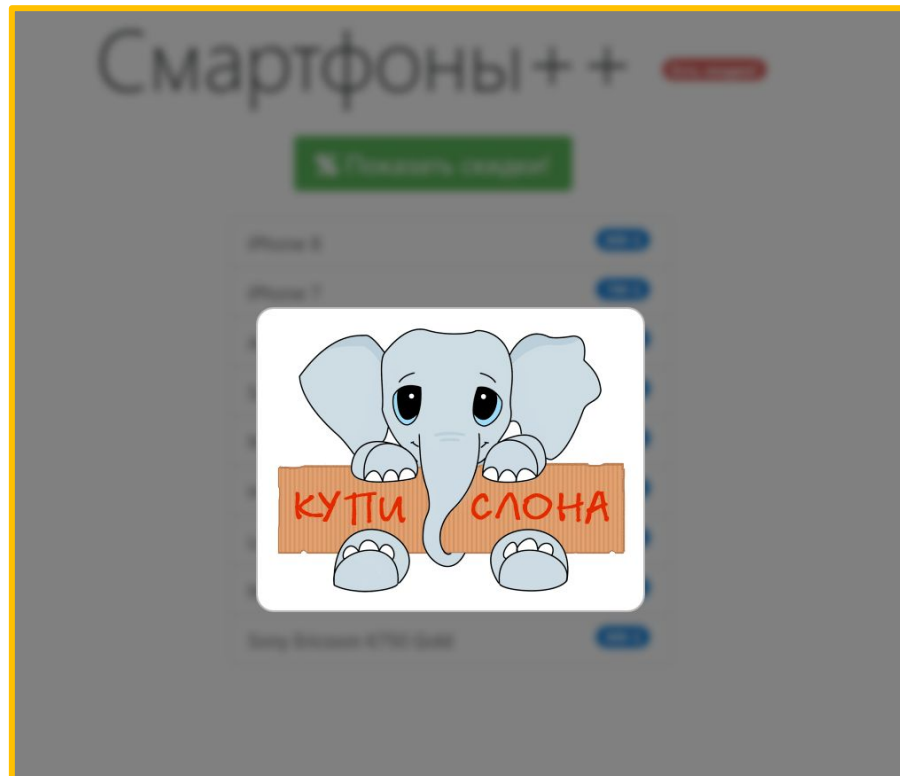
Web.Dev.
Courses

sp.courses.dp.ua

Немного практики

Баннер на всю страницу

Создадим «надоедливый» баннер который будет появляться спустя некоторое время после загрузки страницы, и который перекроет всю страницу.



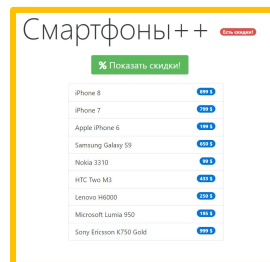
Баннер на всю страницу

Мы напишем код который будет расположен во внешнем JS-файле, **banner.js** и который можно будет подключить к любой странице при помощи просто

включения: `<script src=" ./banner.js "></script>`

В качестве страницы к которой мы будем подключать файл можем взять любую, например заготовку прошлого занятия:

<http://files.courses.dp.ua/web/10/ex01.html>



Баннер на всю страницу

Изображение для нашего примера доступно по адресу: <http://files.courses.dp.ua/images/pic.gif>

Исходный код файла *banner.js* (часть 1)

```
1
2  window.onload = function(){
3
4      function show_banner(){
5
6          //Создаём полупрозрачную подложку
7          var substrate = document.createElement("div");
8
9
10         substrate.style.position = "absolute";
11         substrate.style.width = "100vw";
12         substrate.style.height = "100vh";
13         substrate.style.backgroundColor = "black";
14         substrate.style.opacity = "0.5";
15         substrate.style.left = "0";
16         substrate.style.top = "0";
17         substrate.style.zIndex = 100;
18         substrate.className = "banner_aa6456bc55fde736c";
19         document.body.appendChild(substrate);
```



Баннер на всю страницу

Исходный код файла *banner.js* (часть

2)

```
21 //Создаём изображение
22 var banner = document.createElement("img");
23
24 banner.src = "http://files.courses.dp.ua/images/pic.gif";
25 banner.style.padding = "20px";
26 banner.style.backgroundColor = "white";
27 banner.style.borderRadius = "20px";
28 banner.style.border = "2px solid #bbb";
29 banner.style.display = "block";
30 banner.style.position = "absolute";
31 banner.style.left = "calc(50vw - 200px)";
32 banner.style.top = "calc(50vh - 150px)";
33 banner.style.zIndex = 200;
34 banner.style.filter = "none";
35 banner.className = "banner_aa6456bc55fde736c";
36
37 document.body.appendChild(banner);
38
39 //Создаём новый набор стилей и добавляем его в документ
40 var style = document.createElement("style");
41 document.head.appendChild(style);
42 var rule = "body > *:not(.banner_aa6456bc55fde736c) { filter: blur(5px); }";
43 style.sheet.insertRule(rule, 0);
44
45
46
47 //Создаём обработчик событий для подложки, который при клике скроет баннер
48 substrate.onclick = function(){
49     substrate.remove();
50     banner.remove();
51     style.remove();
52 }
53 }
54
55 //Запустим показ баннера через 5 секунд
56 setTimeout(show_banner, 5000);
57 }
```



Подробнее о DOM

Добавление элемента на страницу

Чтобы добавить элемент на страницу, необходимо определить к какому из существующих элементов его необходимо прикрепить.

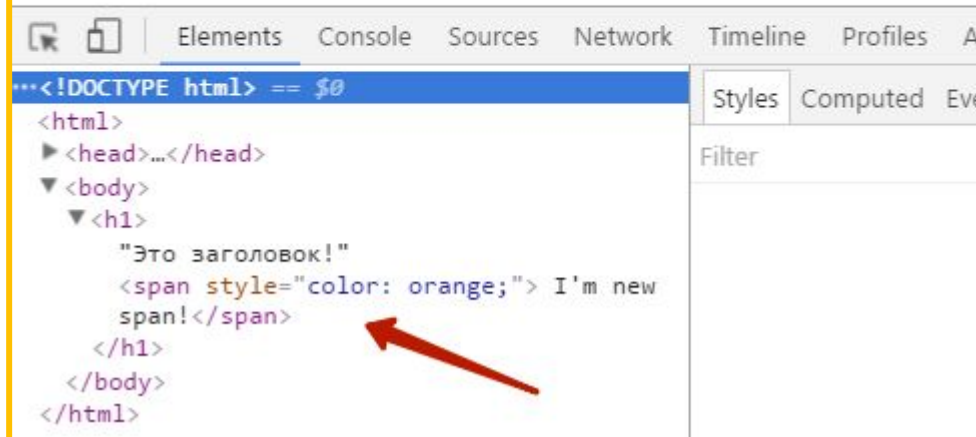
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <script>
5
6  window.onload = function(){
7
8      var element = document.createElement("span");
9      element.innerHTML = " I'm new span!";
10     element.style.color = "orange";
11
12     var h1 = document.querySelector("h1");
13
14     h1.appendChild(element);
15
16 }
17
18 </script>
19 </head>
20 <body>
21     <h1>Это заголовок!</h1>
22 </body>
23 </html>
```



Добавление элемента на

с

Это заголовок! I'm new span!



```
...<!DOCTYPE html> == $0
<html>
  <head>...</head>
  <body>
    <h1>
      "Это заголовок!"
      <span style="color: orange;"> I'm new
      span!</span>
    </h1>
  </body>
</html>
```

`document.createElement("tag_name")` – метод создающий пустой элемент (объект, тег), которые еще не входит в документ, но его свойства уже можно наполнять необходимыми данными;

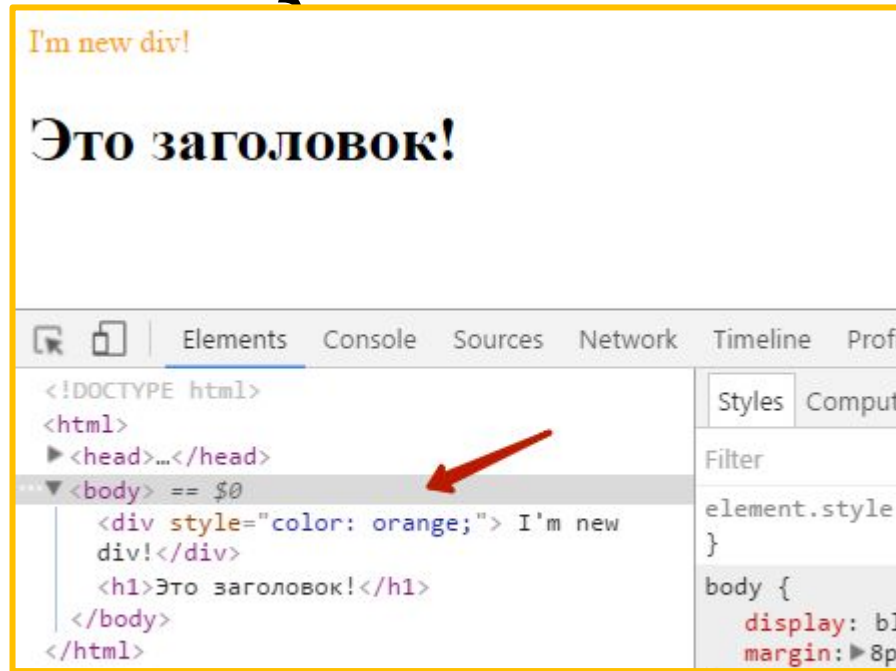
`.appendChild(new_element)` – метод добавляющий новый элемент к потомкам того элемента у которого **`.appendChild`** вызывается.

Изменение позиции элемента в документе

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <script>
5
6     window.onload = function(){
7
8         var element = document.createElement("div");
9         element.innerHTML = " I'm new div!";
10        element.style.color = "orange";
11
12        var h1 = document.querySelector("h1");
13
14        document.body.insertBefore(element, h1);
15    }
16
17 </script>
18 </head>
19 <body>
20     <h1>Это заголовок!</h1>
21 </body>
22 </html>
```

Метод ***.appendChild()*** всегда добавляет элемент в конец списка. Но есть возможность задать позицию вставки элемента среди потомком тега. Для этого существует метод ***.insertBefore()***.

Изменение позиции элемента в



.insertBefore(new_el, ref_el) – добавляет элемент в список дочерних элементов родителя перед указанным элементом. ***new_el*** – элемент который вставляется, ***ref_el*** – элемент перед которым вставляется элемент.

Не менее полезные свойства объектов-

.parentNode – свойство, которое содержит ссылку на родительский элемент (тег).

.className – свойство содержит полный список всех классов которые присвоены тегу (одной строкой).

.classList – свойство содержит список всех классов которые присвоены тегу (в виде массива).

.classList.add('cat') – метод добавляет класс к тегу (если есть другие классы то они остаются).

.classList.remove('cat') – метод удаляет класс у тегу (если есть другие классы то они не затрагиваются).

.classList.contains('cat') – метод проверяет наличие у тега заданного класса (возвращает true/false).

Свойства `.children` и `.nodeChild`

МАССИВЫ С ПОТОМКАМИ ТЕГА

```
<div>  
  <p>Text #1</p>  
  <p>Text #2</p>  
  <p>Text #3</p>  
  <p>Text #4</p>  
  <p>Text #5</p>  
</div>
```

```
▼ NodeList[11] ⓘ  
  ▶ 0: text  
  ▶ 1: p  
  ▶ 2: text  
  ▶ 3: p  
  ▶ 4: text  
  ▶ 5: p  
  ▶ 6: text  
  ▶ 7: p  
  ▶ 8: text  
  ▶ 9: p  
  ▶ 10: text  
  length: 11  
  ▶ __proto__: Object
```

.childNodes

```
▼ [p, p, p, p, p] ⓘ  
  ▶ 0: p  
  ▶ 1: p  
  ▶ 2: p  
  ▶ 3: p  
  ▶ 4: p  
  length: 5  
  ▶ __proto__: Object
```

.children

Таймер

В JavaScript есть возможность отложить какое-то действие на время, или выполнять действие многократно через заданные интервалы времени.

setTimeout(some_func, time, param) – функция которая после истечения времени заданного в параметре ***time*** (задаётся в миллисекундах) выполнить один раз функцию имя которой передана в параметре ***some_func***, если этой функции нужно передать какие-либо параметры их можно заранее указать в ***param***.

setInterval(some_func, time, param) – функция которая будет через каждый промежуток времени ***time*** (задаётся в миллисекундах) выполнять функцию имя которой передана в параметре ***some_func***, если этой функции нужно передать какие-либо параметры их можно заранее указать в ***param***.

DOM – Document Object Model

*(объектная модель
документа)*

*Стандарт который определяет из
каких объектов браузер собирает
дерево документа, и какие свойства
есть у этих объектов у этих.*

<https://learn.javascript.ru/document>

Немного практики

Немного практики

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>DOM.task1</title>
5 <script>
6   window.onload = function(){
7     //Начало вашего кода.....
8
9     //.....Конец вашего кода
10  }
11 </script>
12 </head>
13 <body>
14 <h1>Каталог телефонов <button>Отсортировать</button></h1>
15 <div id="phone_list">
16   <div>iPhone 3Gs <span>582$</span></div>
17   <div>Lenovo <span>997$</span></div>
18   <div>HTC <span>698$</span></div>
19   <div>Samsung <span>242$</span></div>
20   <div>Fly <span>793$</span></div>
21   <div>Asus <span>771$</span></div>
22   <div>Nokia <span>108$</span></div>
23   <div>LG <span>214$</span></div>
24   <div>IPhone 5s <span>216$</span></div>
25   <div>IPhone 6+ <span>326$</span></div>
26   <div>IPhone 6s+ <span>799$</span></div>
27 </div>
28 </body>
29 </html>
```

Каталог телефонов

iPhone 3Gs 582\$
Lenovo 997\$
HTC 698\$
Samsung 242\$
Fly 793\$
Asus 771\$
Nokia 108\$
LG 214\$
IPhone 5s 216\$
IPhone 6+ 326\$
IPhone 6s+ 799\$



По нажатию на кнопку необходимо отсортировать список по возрастанию цены.

Скопируйте код:

<http://courses.dp.ua/files/is/tasks/task1.html>

Немного практики

```
5 <script>
6   window.onload = function() {
7     //Начало вашего кода.....
8     var btn = document.querySelector("button");
9     btn.onclick = function() {
10      var list = document.querySelector("h1 + div");
11      var elements = list.children;
12      for(var i = 0; i < elements.length - 1; i++){
13        for(var j = 0; j < elements.length - 1 - i; j++){
14          var price1 = parseInt(elements[j].querySelector("span").innerHTML);
15          var price2 = parseInt(elements[j+1].querySelector("span").innerHTML);
16          if(price1 > price2) {
17            list.insertBefore(elements[j+1], elements[j]);
18          }
19        }
20      }
21    }
22    //.....Конец вашего кода
23  }
24 </script>
```

*По нажатию на кнопку необходимо
отсортировать список по возрастанию
цены.*

Домашнее задание

Немного практики

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>DOM.task1</title>
5 <script>
6   window.onload = function(){
7     //Начало вашего кода.....
8
9     //.....Конец вашего кода
10  }
11 </script>
12 </head>
13 <body>
14 <h1>Каталог телефонов <button>Отсортировать</button></h1>
15   <div id="phone_list">
16     <div>iPhone 3Gs <span>582$</span></div>
17     <div>Lenovo <span>997$</span></div>
18     <div>HTC <span>698$</span></div>
19     <div>Samsung <span>242$</span></div>
20     <div>Fly <span>793$</span></div>
21     <div>Asus <span>771$</span></div>
22     <div>Nokia <span>108$</span></div>
23     <div>LG <span>214$</span></div>
24     <div>IPhone 5s <span>216$</span></div>
25     <div>IPhone 6+ <span>326$</span></div>
26     <div>IPhone 6s+ <span>799$</span></div>
27   </div>
28 </body>
29 </html>
```

Каталог телефонов

iPhone 3Gs 582\$
Lenovo 997\$
HTC 698\$
Samsung 242\$
Fly 793\$
Asus 771\$
Nokia 108\$
LG 214\$
IPhone 5s 216\$
IPhone 6+ 326\$
IPhone 6s+ 799\$



По нажатию на кнопку необходимо отсортировать список по возрастанию цены. При повторном нажатии сортировка должна выполняться по убыванию цены. И так по очереди переключать направление сортировки.

Скопируйте код:

<http://courses.dp.ua/files/is/tasks/task1.html>