

# ВЕМ! БЭМ!

Вадим Пацев  
Яндекс

# В реальном мире

Верстка конечна, сделал и забыл.

Дизайн страницы зафиксирован и меняется не часто.

Заказчик не может менять HTML-разметку.

Для изменений доступен только текстовый контент.

# В Яндексе

Страницы меняют свой вид.

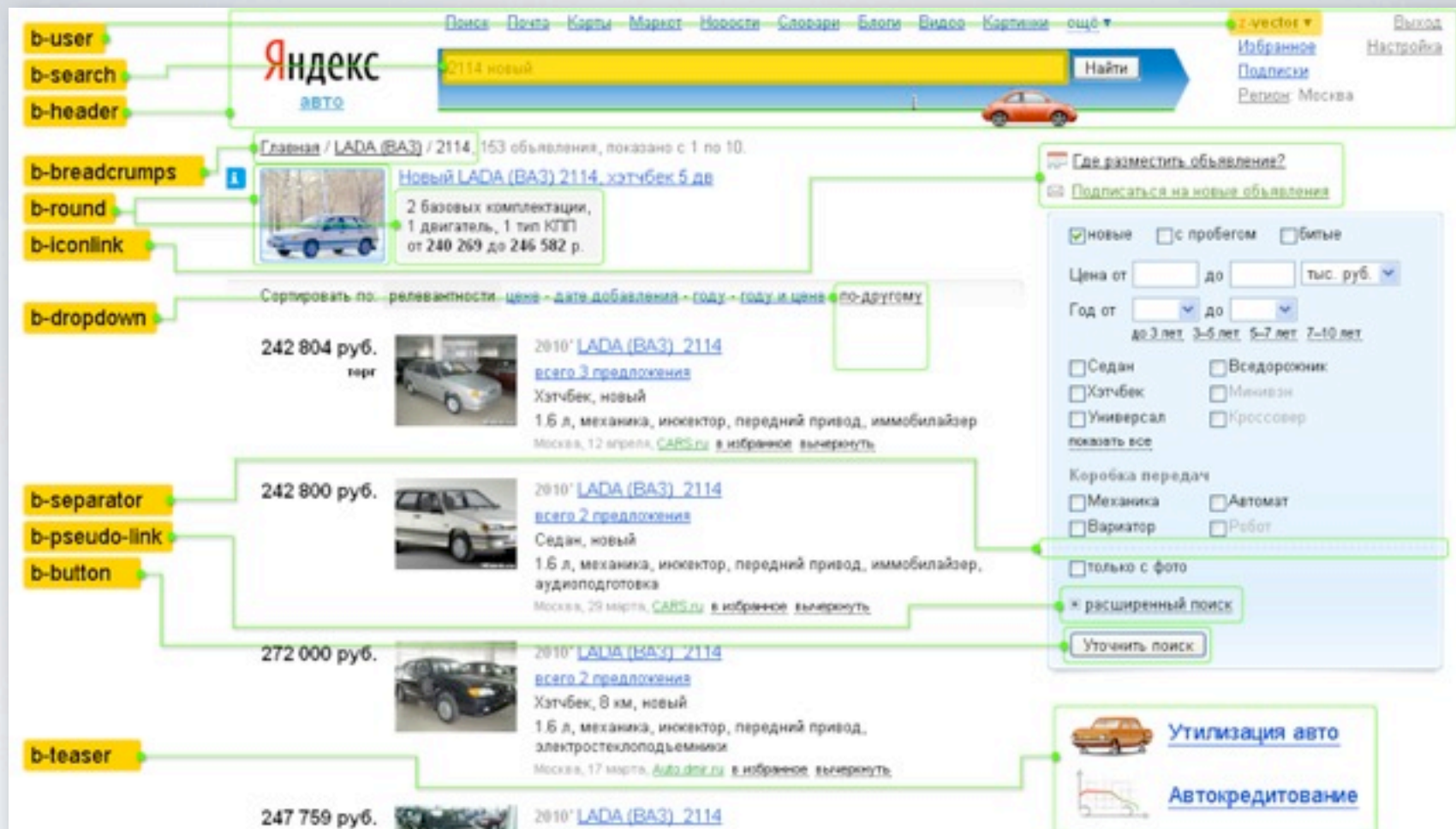
Блоки на странице меняют дизайн и своё расположение, появляются и исчезают.

Сегодня некий блок существует в единственном числе. Завтра оказывается, что их уже много.



БЭМ!





# БЭМ!

## Блок Элемент Модификатор

# БЛОК ЭЛЕМЕНТ МОДИФИКАТОР

Блоки могут быть вложенными

Много однотипных блоков на странице

Блоки могут располагаться в любом месте

# БЛОК ЭЛЕМЕНТ МОДИФИКАТОР

Не используем ID

Не используем селекторы типа



# НЕ ИСПОЛЬЗУЕМ ID

```
1 | #header {}
```

Со временем на странице может быть много блоков, поэтому мы id не используем для css.

id для быстрого доступа к DOM-узлу из js.

# НЕ ИСПОЛЬЗУЕМ СЕЛЕКТОРЫ ТИПА

```
1 p { color: #ccc }  
2 table.news td { border-bottom: 1px solid #ccc }  
3 .list li { float: left }
```

Скорее всего, цвет текста должен быть серым не во всех параграфах.

Или завтра внутри table.news придется положить другую таблицу, у которой ячейки не должны иметь бордера.

Или внутри .list вложим другой список ul, дети которого не должны иметь float.



БЭМ!

# БЛОК

Некий фрагмент DOM-дерева, который выполняет специальные функции на странице.

Содержит информацию о самом себе (плюс о своих детях), и только.

HTML-разметка обеспечивает нужную структуру блока, а CSS-правила обеспечивают внешний вид и его поведение на странице.



# БЛОК

Блок (и его элемент) всегда имеет CSS-класс.

Блок имеет префикс «b-», слова в имени разделяются символом «-». Есть служебные префиксы «i-» и «m-». Об этом позже.

```
1 <h1 class="b-title"></h1>
2
3 <br />
4
5 <span class="b-news"></span>
6
7 <div>
8     <a class="b-link" href=""></a>
9 </div>
```

# БЛОК

Имя класса должно отвечать на вопрос:

«Зачем нужен этот блок?»

```
1 .b-title {}          /* Какой-то заголовок */  
2 .b-menu {}           /* Какое-то меню */  
3 .b-articles-list {}  /* Список каких-то статей */
```

# ЭЛЕМЕНТ

Часть блока, которая отвечает за какую-то отдельную функцию блока.

# ЭЛЕМЕНТ

Если в блоке есть элемент, ему дается имя по схеме

b-имя-блока\_\_имя-элемента-блока

```
1 <div class="b-title">
2   <img class="b-title__icon" src="" />
3   <h1 class="b-title__text"></h1>
4 </div>
```

```
1 .b-title {}
2 .b-title__icon {}
3 .b-title__text {}
```

```
1 <div class="b-menu">
2   <h1 class="b-menu__head"></h1>
3   <ul class="b-menu__list">
4     <li class="b-menu__item"></li>
5     <li class="b-menu__item"></li>
6     <li class="b-menu__item"></li>
7   </ul>
8 </div>
```

```
1 .b-menu {}
2 .b-menu__head {}
3 .b-menu__list {}
4 .b-menu__item {}
```

# МОДИФИКАТОР

Один и тот же блок на разных страницах может выглядеть по-разному. Эту задачу решает модификатор блока.

Модифицировать блоки мы научились тремя способами:

- Постфиксом
- Контекстом
- Файлом



# Модификация ПОСТФИКСОМ

Дополнительный класс, в котором указываем тип модификации по схеме

b-имя-блока\_тип-модификации\_значение-модификации

```
1 <div class="b-title b-title_page_about"></div>
```

```
1 .b-title_page_about {font-size: 130%}
```

# Модификация КОНТЕКСТОМ

Иногда мы помещаем один блок в другой и требуется, чтобы внутренний блок изменил свой вид зависимости от контекста. Снова используем контекстный селектор потомка.

```
1 <div class="b-menu">
2   <div class="b-title">
3     <h1 class="b-title__text"></h1>
4   </div>
5 </div>
```

```
1 .b-menu .b-title { font-size: 130% }
2 .b-menu .b-title__text { color: red }
```

# Модификация КОНТЕКСТОМ

Если от контекста нужно менять много блоков сразу, создаем специальный модификатор с префиксом m-.

```
1 <body class="m-theme-red">
2   <div class="b-title">...</div>
3   <div class="b-article">...</div>
4   <div class="b-menu">...</div>
5   <div class="b-foot">...</div>
6 </body>
```

```
1 .m-theme-red .b-title,
2 .m-theme-red .b-article,
3 .m-theme-red .b-menu,
4 .m-theme-red .b-foot
5 {
6   margin-left: 2em;
7 }
```

# Модификация файлом

Все модификации создаются в дополнительном файле без всяких постфиксов и m-префиксов. Этот файл грузится только на той странице, где нужна модификация.

```
1 <head>
2   ...
3   <link rel="stylesheet" href="index.css"/>
4   <link rel="stylesheet" href="about.css"/>
5 </head>
6 <body>
7   <div class="b-title">
8     <h1 class="b-title__text"></h1>
9   </div>
10  ...
11 </body>
```

# БЭМ

Взаимное отношение этих трёх терминов описывается так:

- Блок состоит из элементов блока,
- Обе эти сущности в HTML-документе записаны с помощью DOM-узлов.
- Все многообразие состояний блока обеспечивается модификациями.



# БЭМ

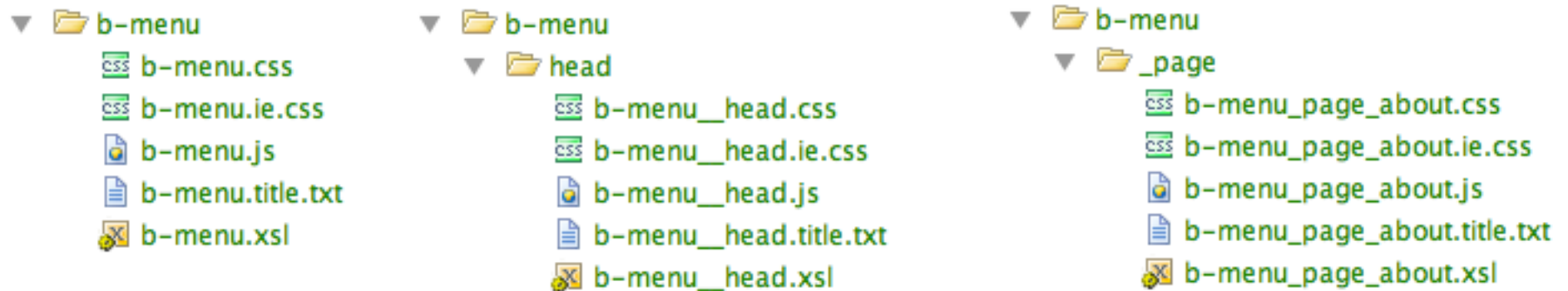
Чего мы добились:

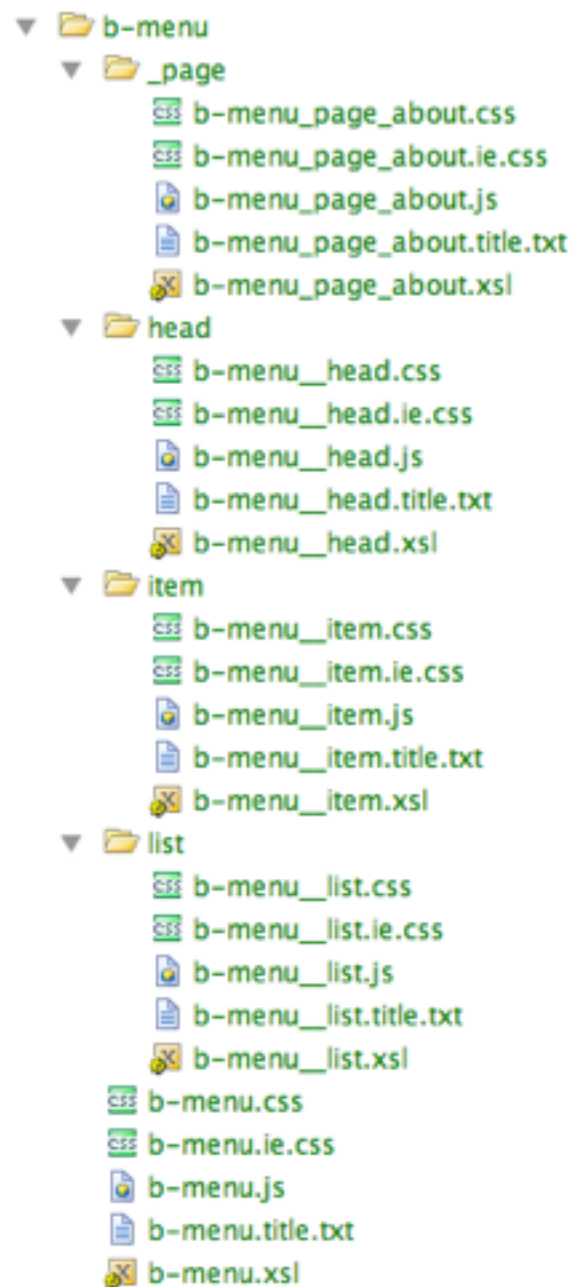
- Селекторы «уникальны» для одного блока — нет конфликтов пересечения стилей с другими блоками.
- Нет селекторов потомков — растёт скорость отрисовки стилей в браузере.
- Мы создали почти уникальные идентификаторы, только они лишены недостатка ID — их можно повторять на странице много раз.

# БЭМ на файловой системе

# БЭМ на файловой системе

Идея «блок хранит информацию о самом себе»  
продолжается и в файловом размещении блока.

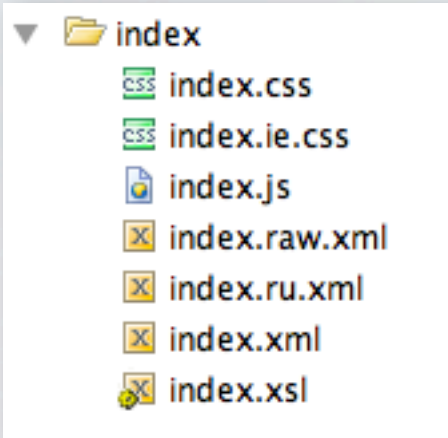




# Структура блока

# Страницы

На странице сервиса нужны только те блоки, которые пользователь должен увидеть. Все блоки на странице не нужны. Эту задачу решает отдельная сущность page.





# Страницы

index.css

```
1 @import url(../../blocks/b-menu/b-menu.css);  
2 @import url(../../blocks/b-menu/head/b-menu__head.css);  
3 @import url(../../blocks/b-menu/_page/b-menu_page_about.css);  
4 @import url(../../blocks/b-menu/_page/b-menu_page_order.css);
```

index.ie.css

```
1 @import url(../../blocks/b-menu/b-menu.ie.css);  
2 @import url(../../blocks/b-menu/head/b-menu__head.ie.css);  
3 @import url(../../blocks/b-menu/_page/b-menu_page_about.ie.css);
```

index.xsl

```
1 <xsl:import href="../../blocks/b-menu/b-menu.xsl"/>  
2 <xsl:import href="../../blocks/b-menu/head/b-menu__head.xsl"/>  
3 <xsl:import href="../../blocks/b-menu/_page/b-menu_page_about.xsl"/>
```

# Страницы

## Подключение css

```
1 <head>
2
3   <!--[if gt IE 7]><!--><link rel="stylesheet" href="index.css"/><!--<![endif]-->
4   <!--[if lt IE 8]><link rel=stylesheet href="index.ie.css"><![endif]-->
5
6   <script type="text/javascript"src="index.js"></script>
7 </head>
```



Такая файловая структура  
позволяет грузить только  
тот код, который нужен.

# БЭМ

# БЭМ

Позволяет писать гибкий и легко изменяемый код.

Задаёт четкую структуру html/css.

Задаёт четкую файловую структуру.

Все это позволяет легко масштабировать и изменять код проекта.



<http://clubs.ya.ru/bem>

# Спасибо. Вопросы?

Вадим Пацев

[basvasilich@yandex.ru](mailto:basvasilich@yandex.ru)

[@basvasilich](#)

<http://basvasilich.moikrug.ru>