



Миксы во вселенной БЭМ

Кир Белевич

Руководитель группы мобильных поисковых
интерфейсов

Я.Субботник, Москва, 8 сентября 2012 года

Миксы это...



Миксы это...



**смешивание ВЕМ-сущностей
на одном ДОМ-узле**

Что это значит?

Можно смешивать:

— несколько блоков

b-boy + b-girl

Можно смешивать:

— несколько блоков

b-boy + b-girl

— блок и элемент этого же блока

b-girl + b-girl__heart

Можно смешивать:

— несколько блоков

b-boy + b-girl

— блок и элемент этого же блока

b-girl + b-girl__heart

— блок и элемент другого блока

b-boy + b-girl__heart

Можно смешивать:

— элементы разных блоков

b-boy__heart + b-girl__heart

Можно смешивать:

— элементы разных блоков

b-boy__heart + b-girl__heart

— блок с модификатором и другой блок

b-girl + b-girl_type_hot + b-boy

Можно смешивать:

— элементы разных блоков

b-boy__heart + b-girl__heart

— блок с модификатором и другой блок

b-girl + b-girl_type_hot + b-boy

— модификаторы разных блоков

b-boy + b-boy_type_hot + b-girl + b-girl_type_hot

Зачем?

Зачем?

1. Создание одного блока на основе другого

- повторное использование уже написанных шаблонов, стилей и скриптов
- избавление от копипаста

- повторное использование уже написанных шаблонов, стилей и скриптов
- избавление от копипаста
- прозрачная и наглядная файловая структура

- повторное использование уже написанных шаблонов, стилей и скриптов
- избавление от копипаста
- прозрачная и наглядная файловая структура
- отдельная полноценная сущность

BEMJSON

```
{
  block: "b-menu",
  mods: { layout: "horiz" },
  mix: [{
    block: "b-my-category-list"
  }],
  content: [
    { elem: "item", ... },
    { elem: "item", ... }
  ]
}
```


HTML

```
<ul class="
  b-menu
  b-menu_layout_horiz
  b-my-category-list
">
  <li class="b-menu__item">..</li>
  <li class="b-menu__item">..</li>
  ..
</ul>
```

CSS

```
.b-my-category-list  
{  
  background: linear-gradient(...);  
}
```

CSS

```
.b-my-category-list  
{  
  background: linear-gradient(...);  
}  
  
.b-my-category-list .b-menu__item  
{  
  padding: 10px;  
}
```

JS: i-bem.js

```
BEM.DOM.decl("b-my-category-list", {  
  ...  
});
```

```
this.findBlockOn([elem], block);
```

```
this.findBlocksOn([elem], block);
```

```
this.liveInitOnBlockInit(blockName,
```

Файловая структура

b-menu

```
|— __item  
|— __layout  
|— b-menu.bemhtml  
|— b-menu.css  
|— b-menu.js
```

b-my-category-list

```
|— __item  
|— __type  
|— b-my-category-list.css  
|— b-my-category-list.js
```



Можно круче

BEMJSON

```
{  
  block: "b-my-category-list",  
  content: [  
    { elem: "item", ... },  
    { elem: "item", ... },  
    ...  
  ]  
}
```

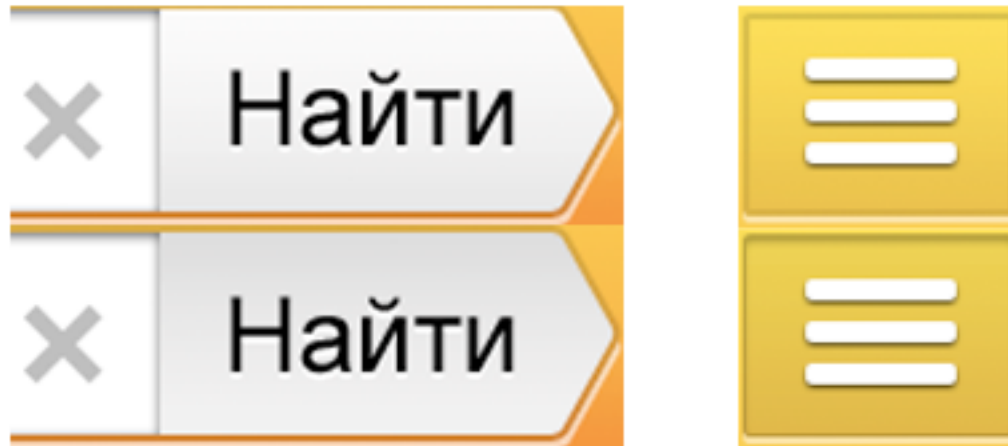

BEMHTML

```
block b-my-category-list, default: {  
  applyCtx({  
    block: 'b-menu',  
    mods: { layout: 'horiz' },  
    mix: { block: this.block },  
    content: this.ctx.content  
  })  
}
```

Зачем?

**2. Расширение
функциональности одного
блока с помощью другого**

Например, нажатое состояние



HTML

```
<button class="
  b-form-button
  i-pressed-controller
">BANG!</button>
```

```
<button class="
  b-form-button
  i-pressed-controller
  i-pressed-controller_pressed_yes
">BANG!</button>
```

CSS

```
.b-form-button.i-pressed-  
controller_pressed_yes  
{  
    background-color: red;  
}
```

Зачем?

3. Связь между абсолютно- независимыми блоками

b-boy



b-girl



b-boy







i-love-controller



```
{  
  block: "b-my-world",
```

```
}
```

```
{  
  block: "b-my-world",  
  mix: [{ block: "i-love-controller" }],
```

```
}
```

```
{
  block: "b-my-world",
  mix: [{ block: "i-love-controller" }],
  content: [
    {
      block: "b-boy",

    },

  ],
}
]
```

```
{
  block: "b-my-world",
  mix: [{ block: "i-love-controller" }],
  content: [
    {
      block: "b-boy",
      mix: [{
        block: "i-love-controller",
        elem: "boy"
      }]
    },
  ],
}
]
```

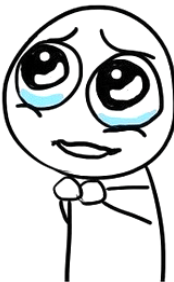
```
{
  block: "b-my-world",
  mix: [{ block: "i-love-controller" }],
  content: [
    {
      block: "b-boy",
      mix: [{
        block: "i-love-controller",
        elem: "boy"
      }]
    }, {
      block: "b-girl",
      mix: [{
        block: "i-love-controller",
        elem: "girl"
      }]
    }
  ]
}
```

JS: i-bem.js

```
BEM.DOM.decl('i-love-controller', {  
  var boy = this.findBlockOn('boy', 'b-boy'),  
      girl = this.findBlockOn('girl', 'b-girl');  
  
});
```


JS: i-bem.js

```
BEM.DOM.decl('i-love-controller', {  
  
  var boy = this.findBlockOn('boy', 'b-boy'),  
      girl = this.findBlockOn('girl', 'b-girl');  
  
  girl.on('loveRequest', function(e, data) {  
    boy.callPhone(data.phone);  
  });  
  
});
```



А? Чё?

Итого

- один блок на основе другого
 - повторное использование шаблонов, стилей и скриптов
 - избавление от копипаста
 - прозрачная и наглядная файловая структура

Итого

- один блок на основе другого
 - повторное использование шаблонов, стилей и скриптов
 - избавление от копипаста
 - прозрачная и наглядная файловая структура
- расширение функционала одного блока другим

Итого

- один блок на основе другого
 - повторное использование шаблонов, стилей и скриптов
 - избавление от копипаста
 - прозрачная и наглядная файловая структура
- расширение функционала одного блока другим
- связь абсолютно-независимых блоков



**ОБОЖАЮ
ИСПОЛЬЗОВАТЬ
МИКСЫ...**

Мои выступления



clck.ru/1wWjV



Кир Белевич

Руководитель группы мобильных
поисковых интерфейсов

deepsweet@yandex-team.ru

[@deepsweet](#)