



ВЕМ IDE — прототип редактора БЭМ-проектов

Саша Белянский, Яндекс

ВЕМир в Питере, 06 сентября 2014

Саша Белянский (Потапенко)

- › belyanskii@yandex-team.ru
- › [@belyanskii](#)
- › github.com/belyanskii
- › Разрабатываю интерфейсы в Яндексе



О чем?

- › Особенности БЭМ — взгляд со стороны редактора
- › Требования для работы с файловой системой
- › BEM IDE — посмотрим как работает и из чего сделана
- › Итог и наши планы на будущее
- › Вопросы
- › Секретное объявление :)

Еще один редактор? o_o

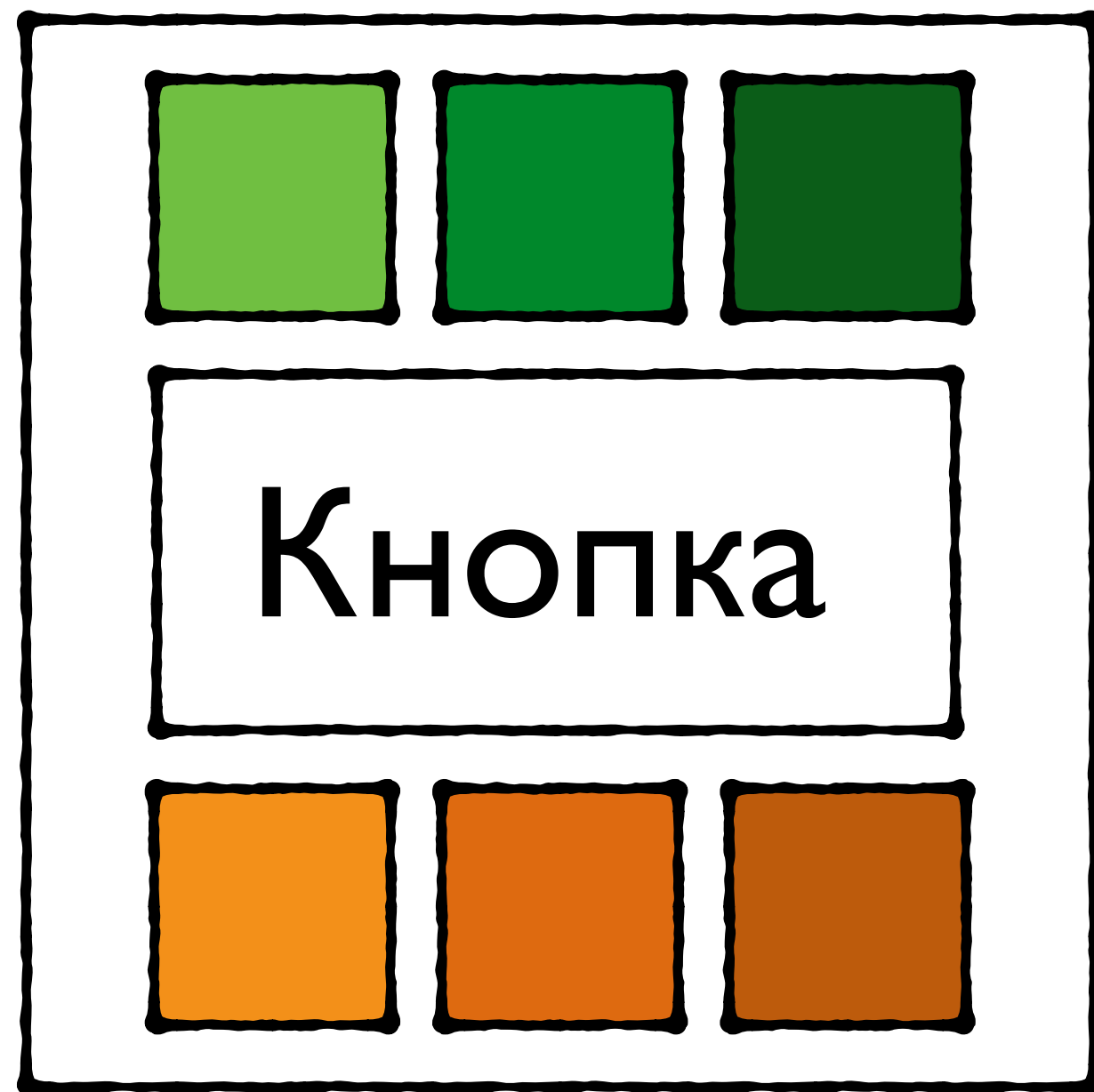
Особенности БЭМ

Взгляд со стороны редактора

Независимость блоков

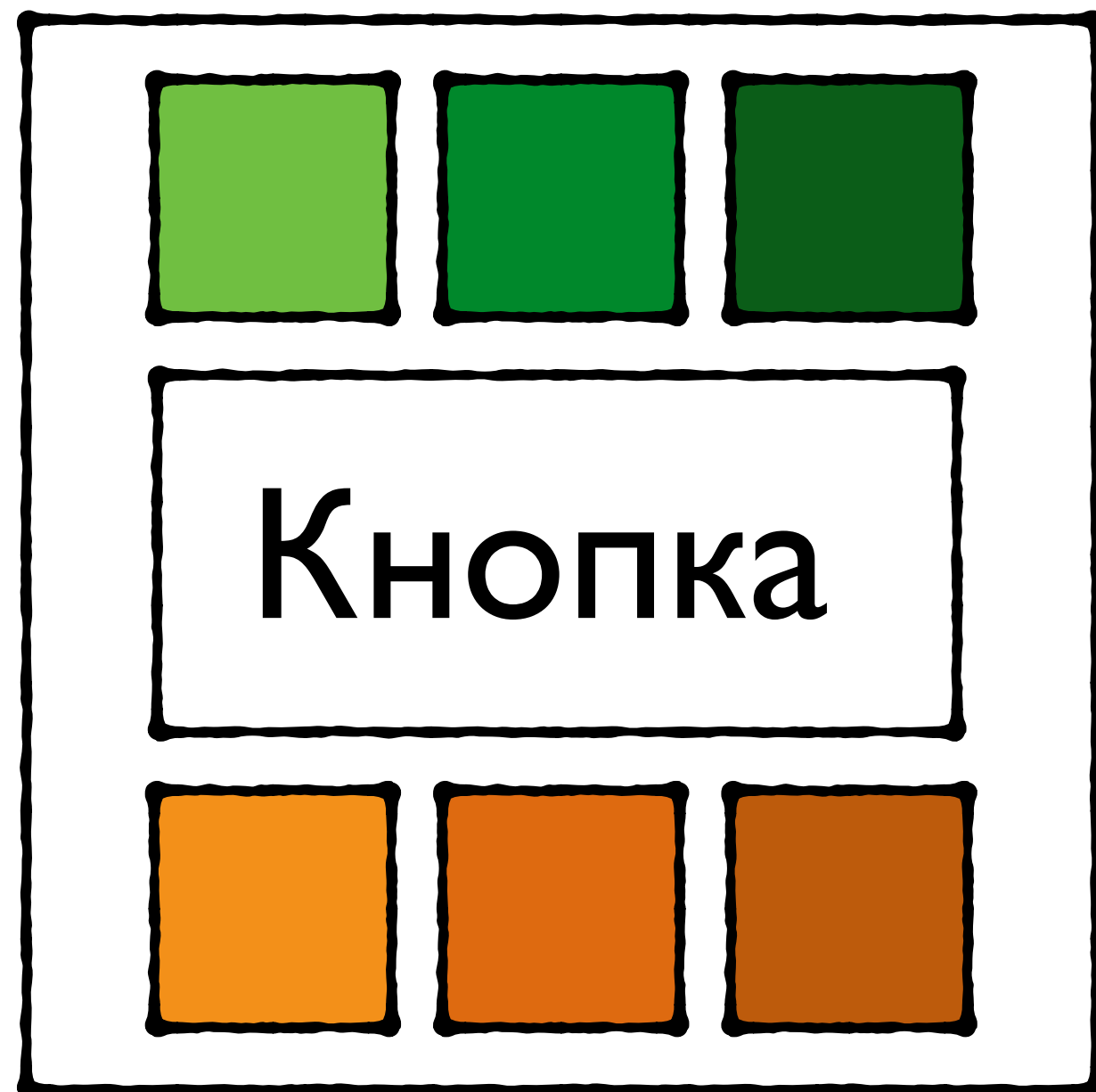


Независимость блоков



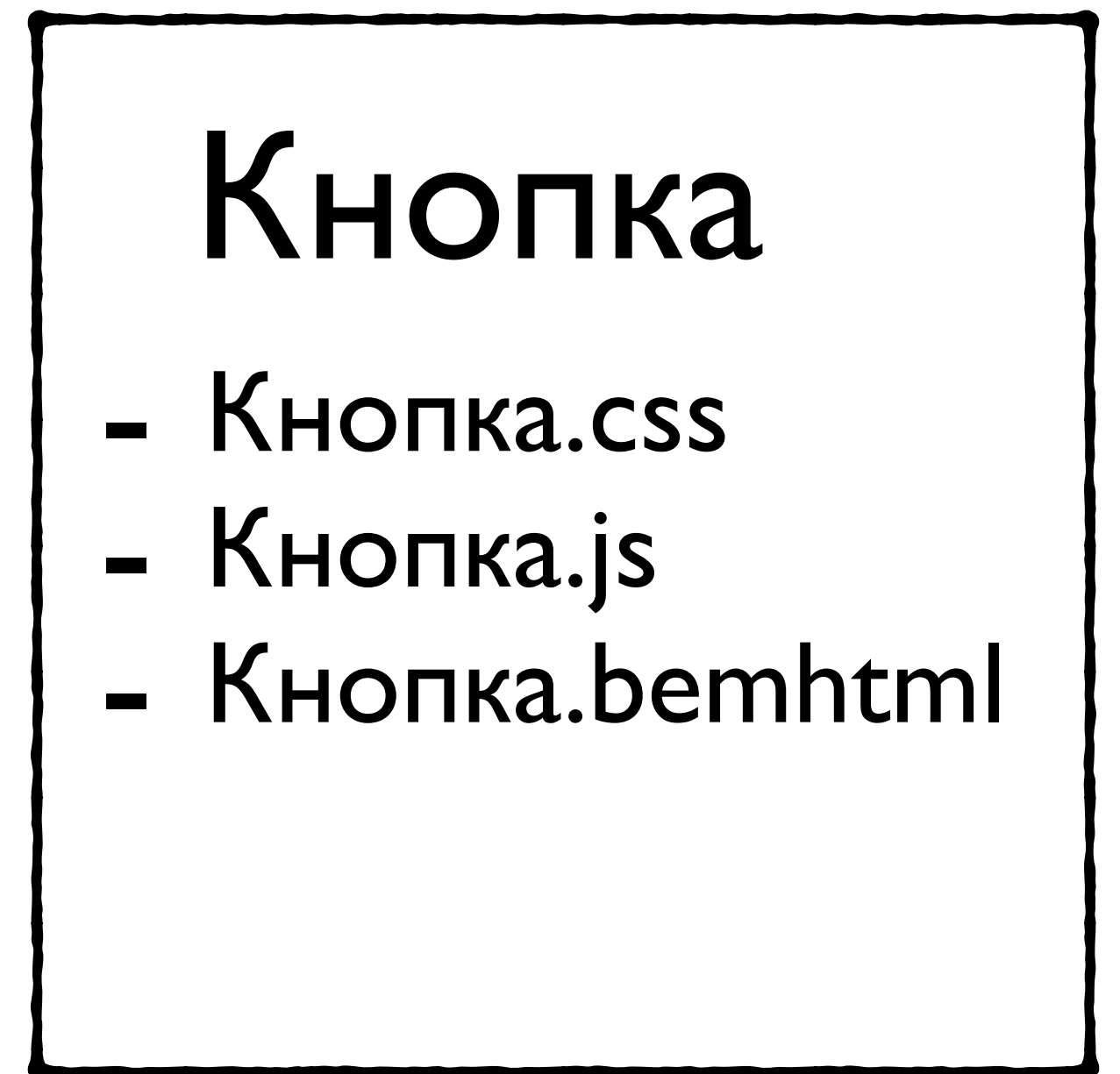
— Интерфейс

Независимость блоков



— Интерфейс

Файловая система —



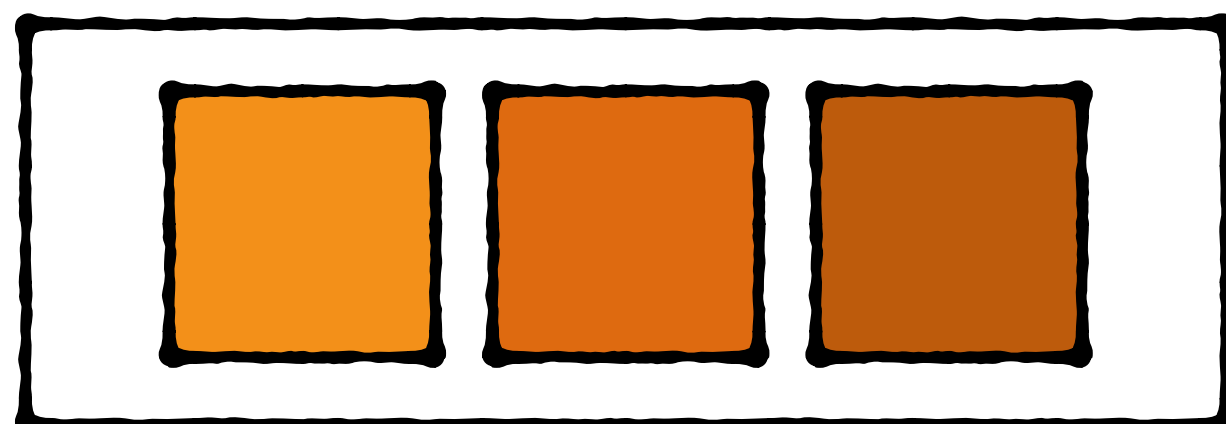
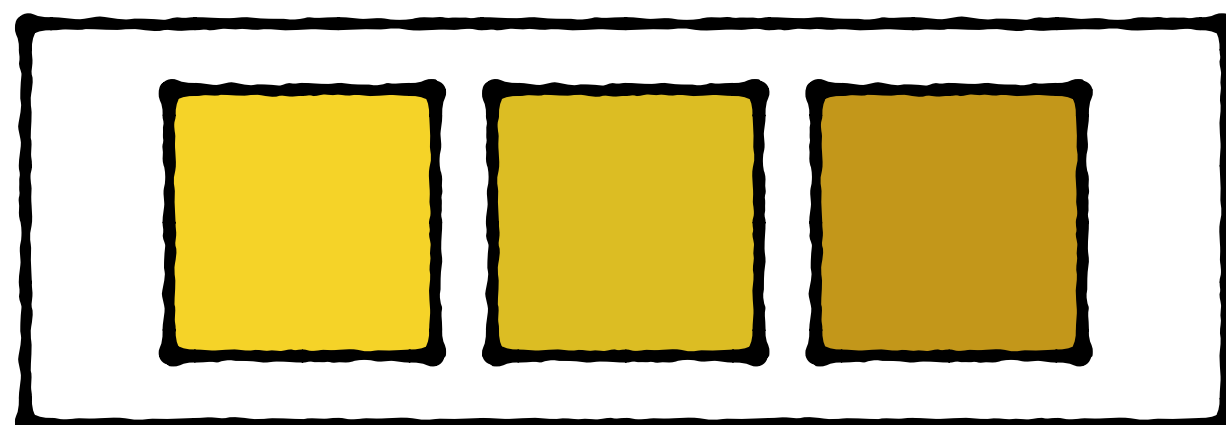
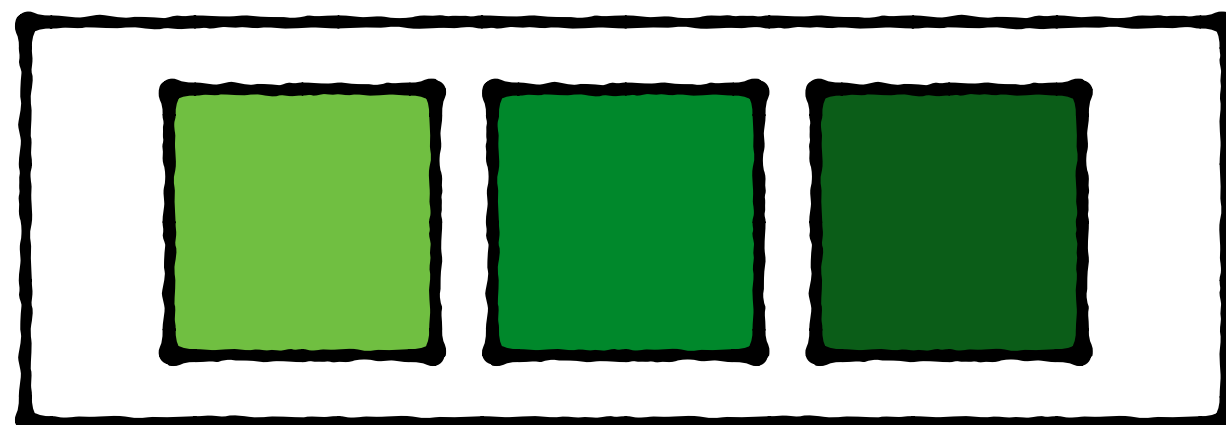


Уровни переопределения



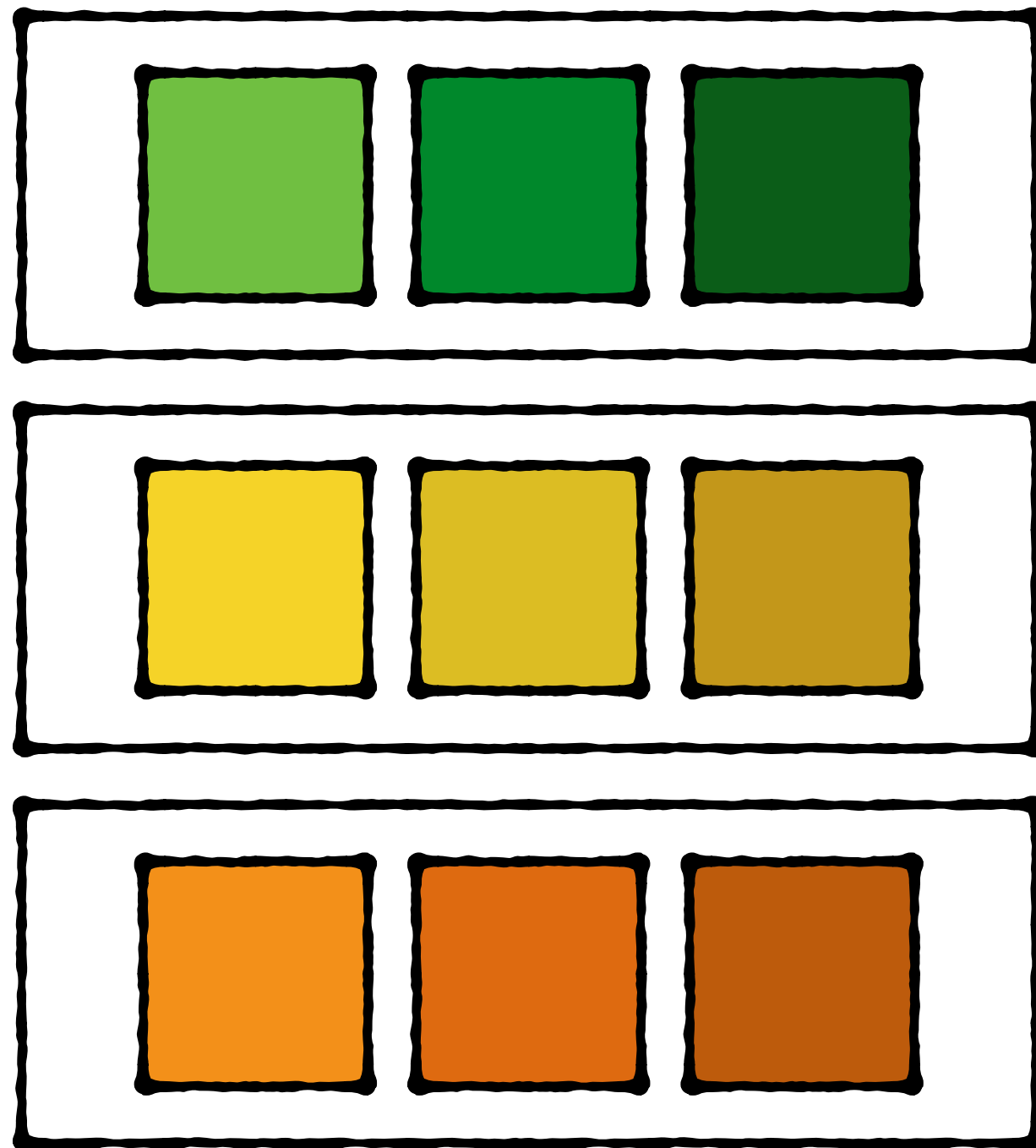
Уровни переопределения

Библиотеки

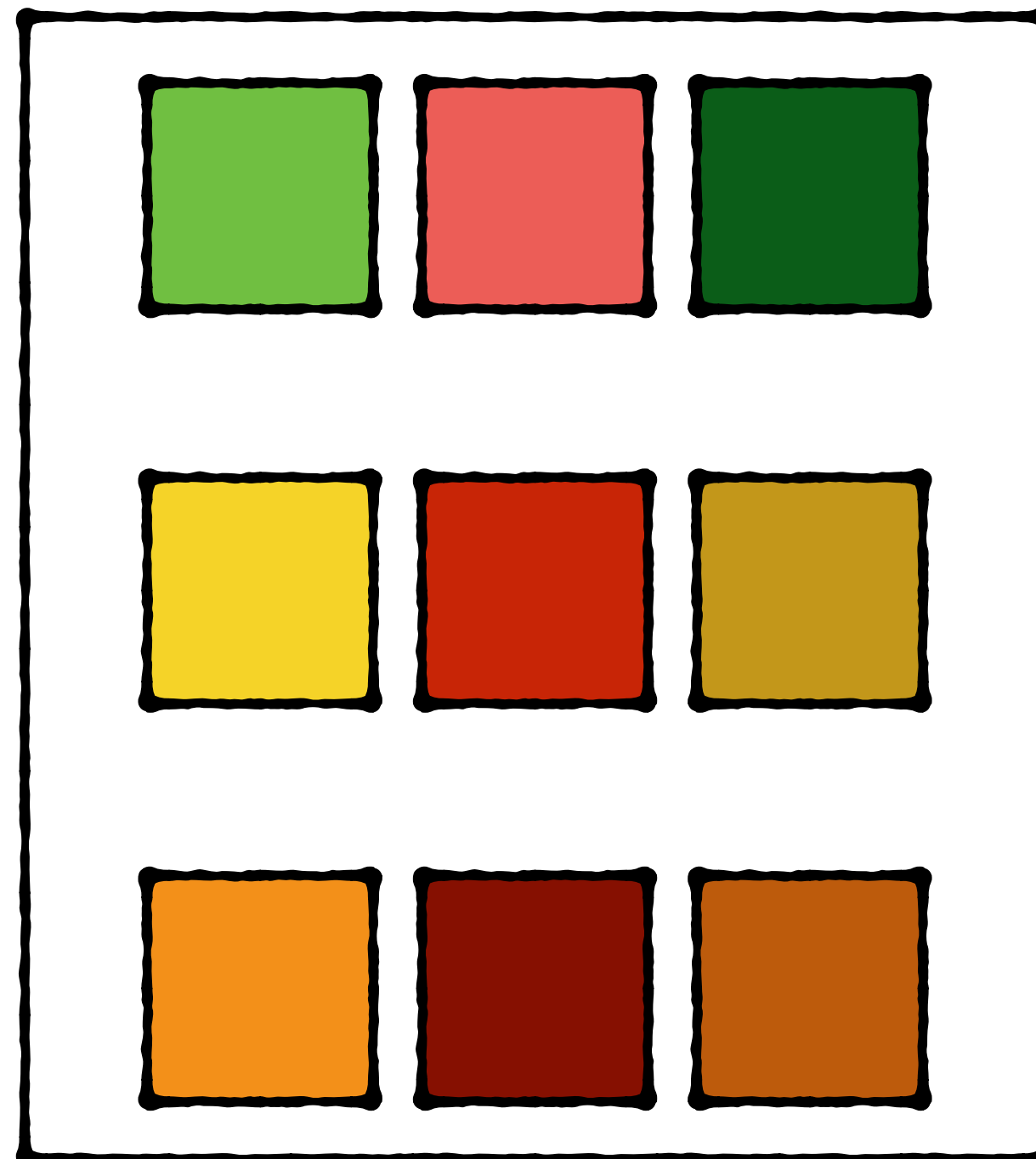


Уровни переопределения

Библиотеки

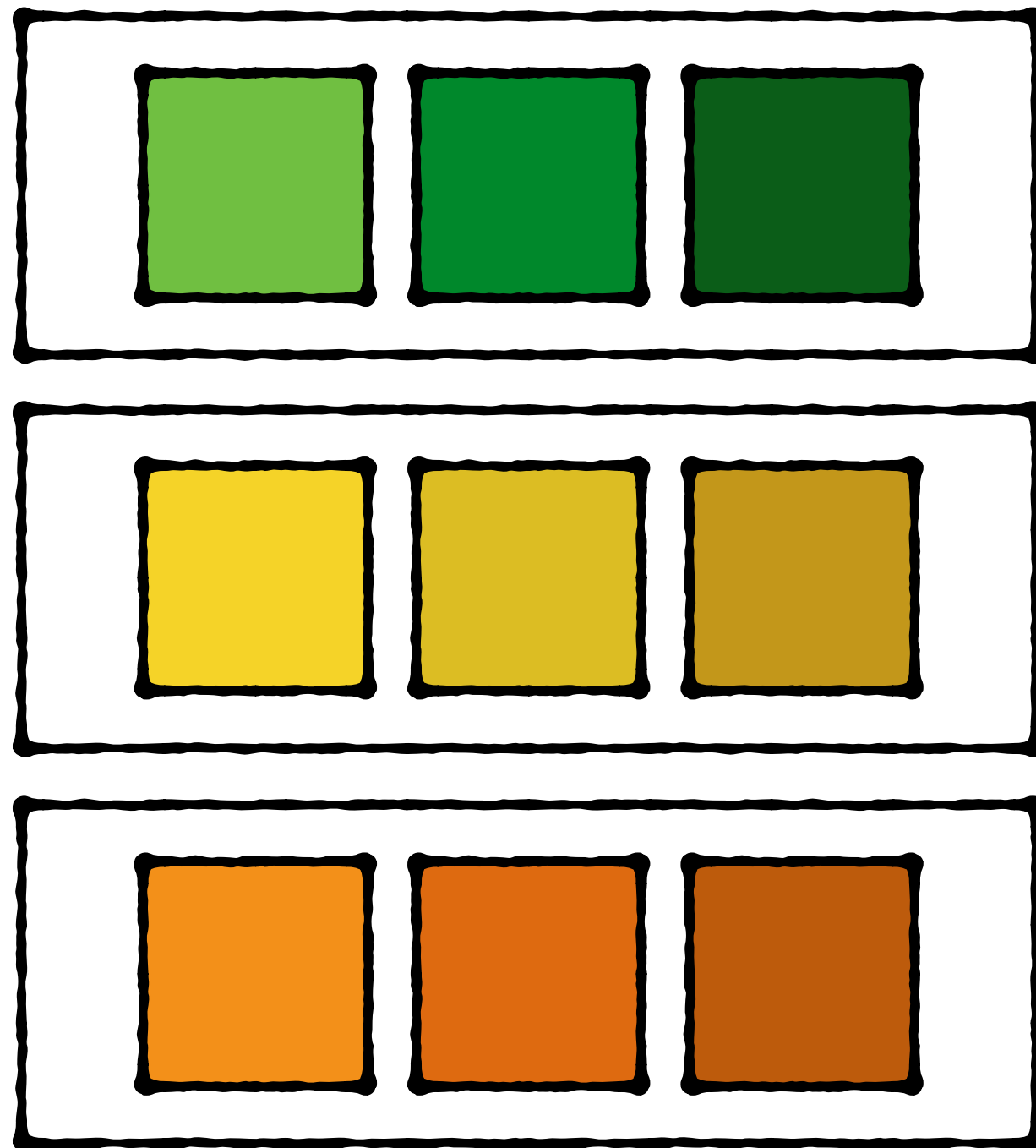


Проект

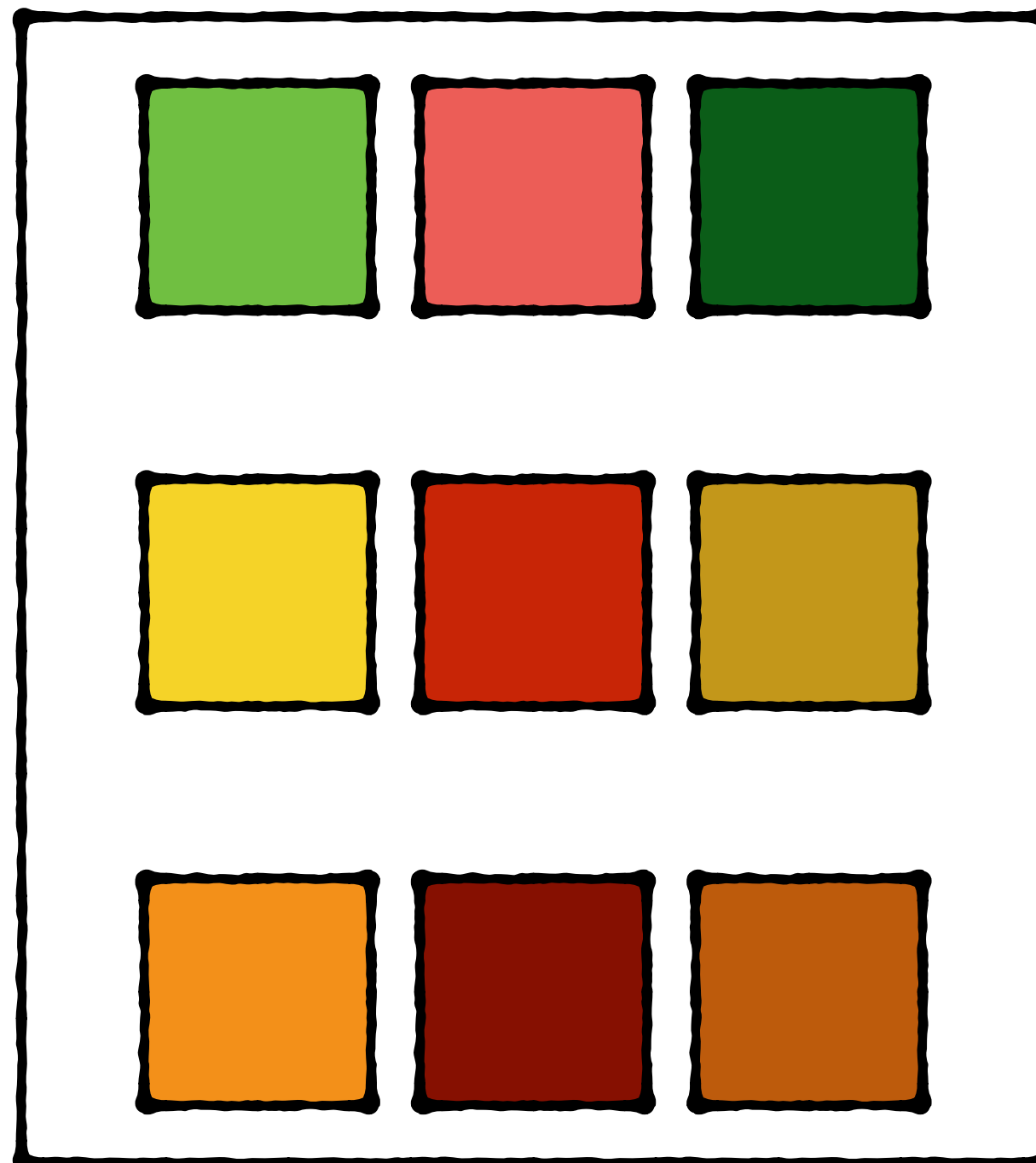


Уровни переопределения

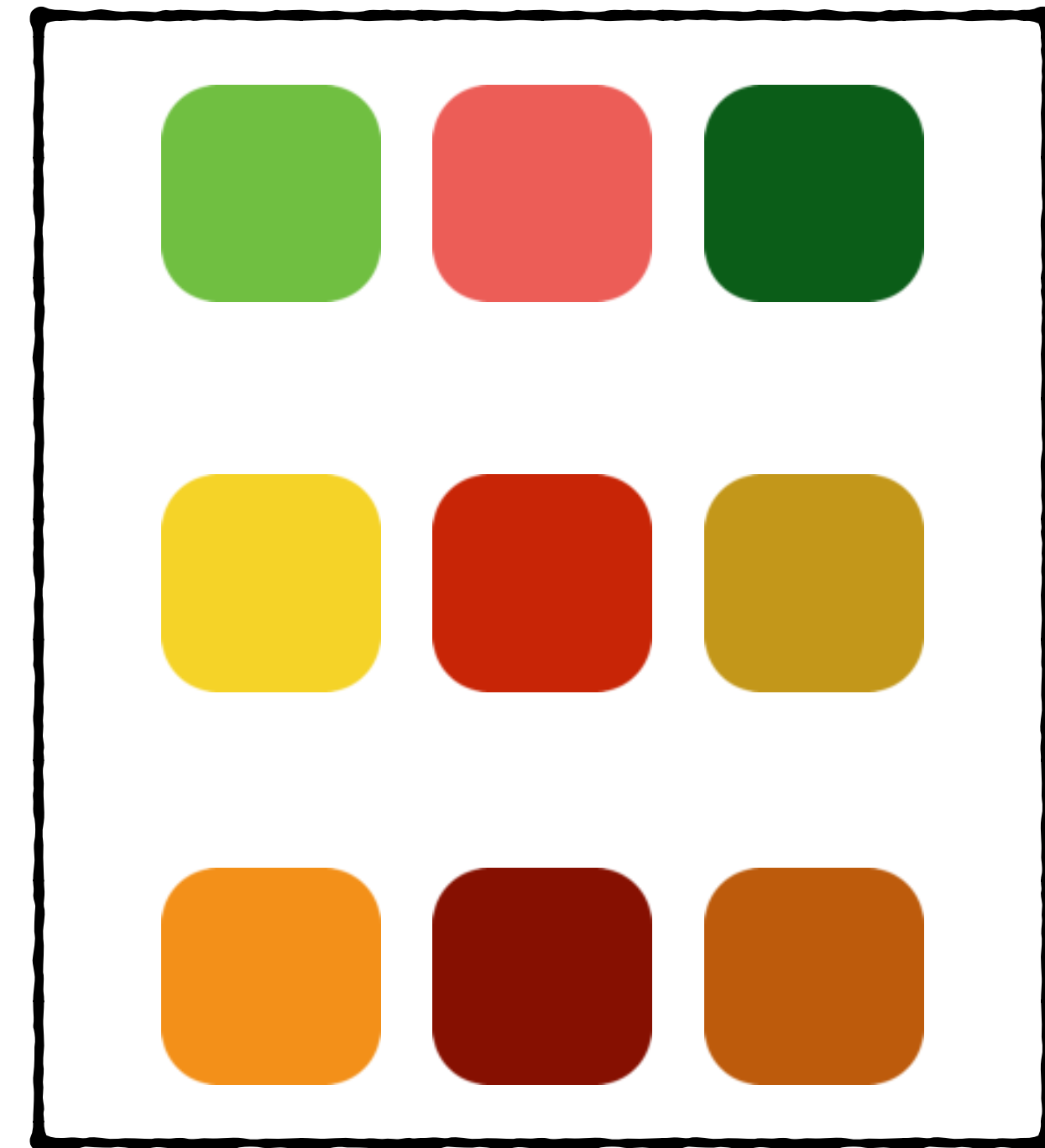
Библиотеки



Проект — desktop



Проект — touch



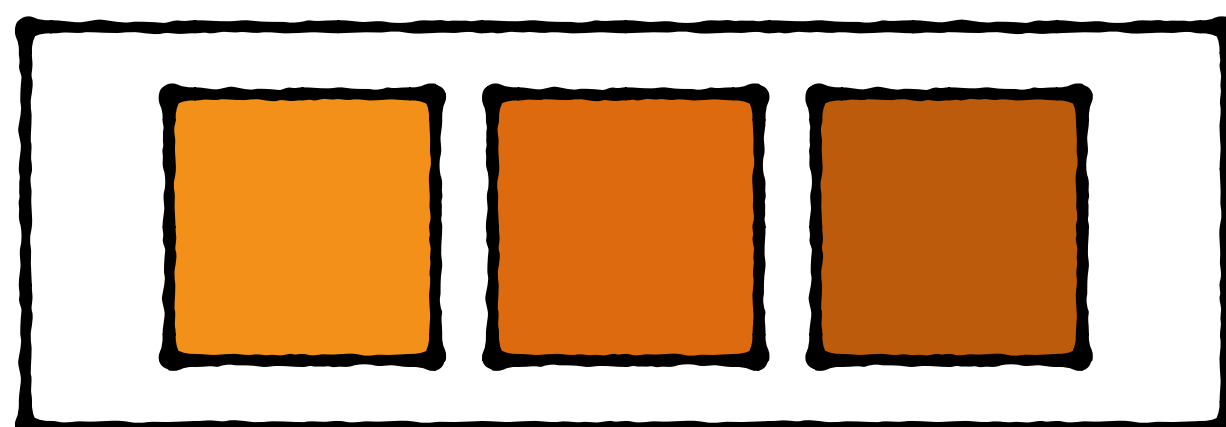
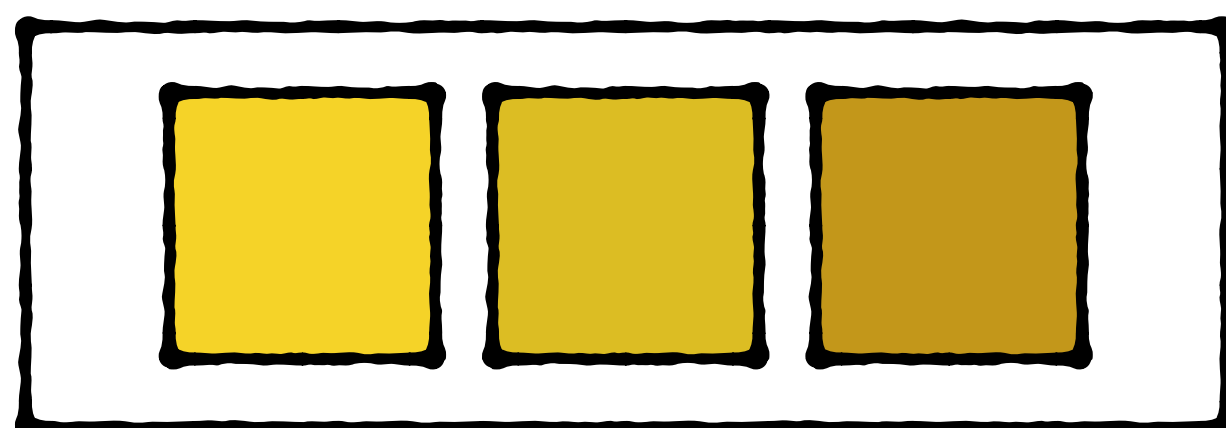
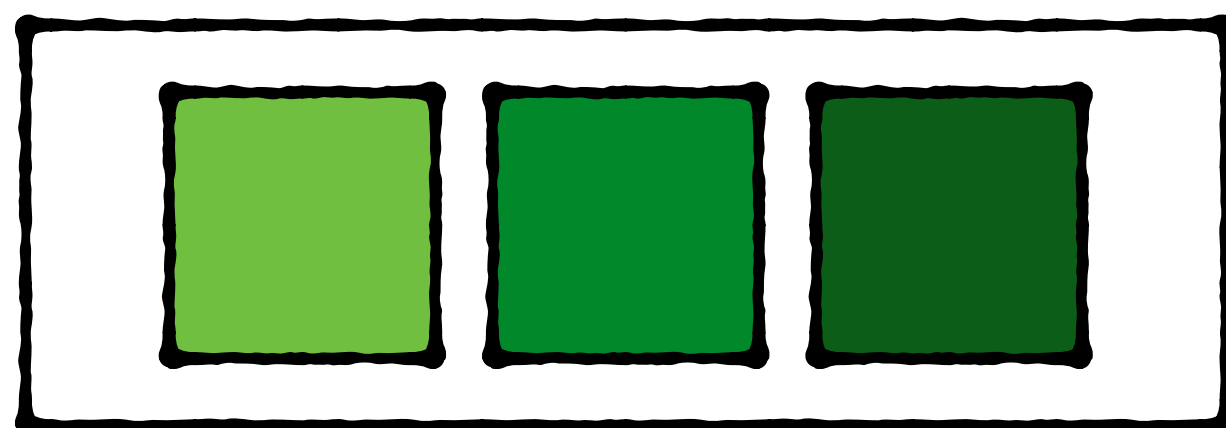


БЭМ-дерево



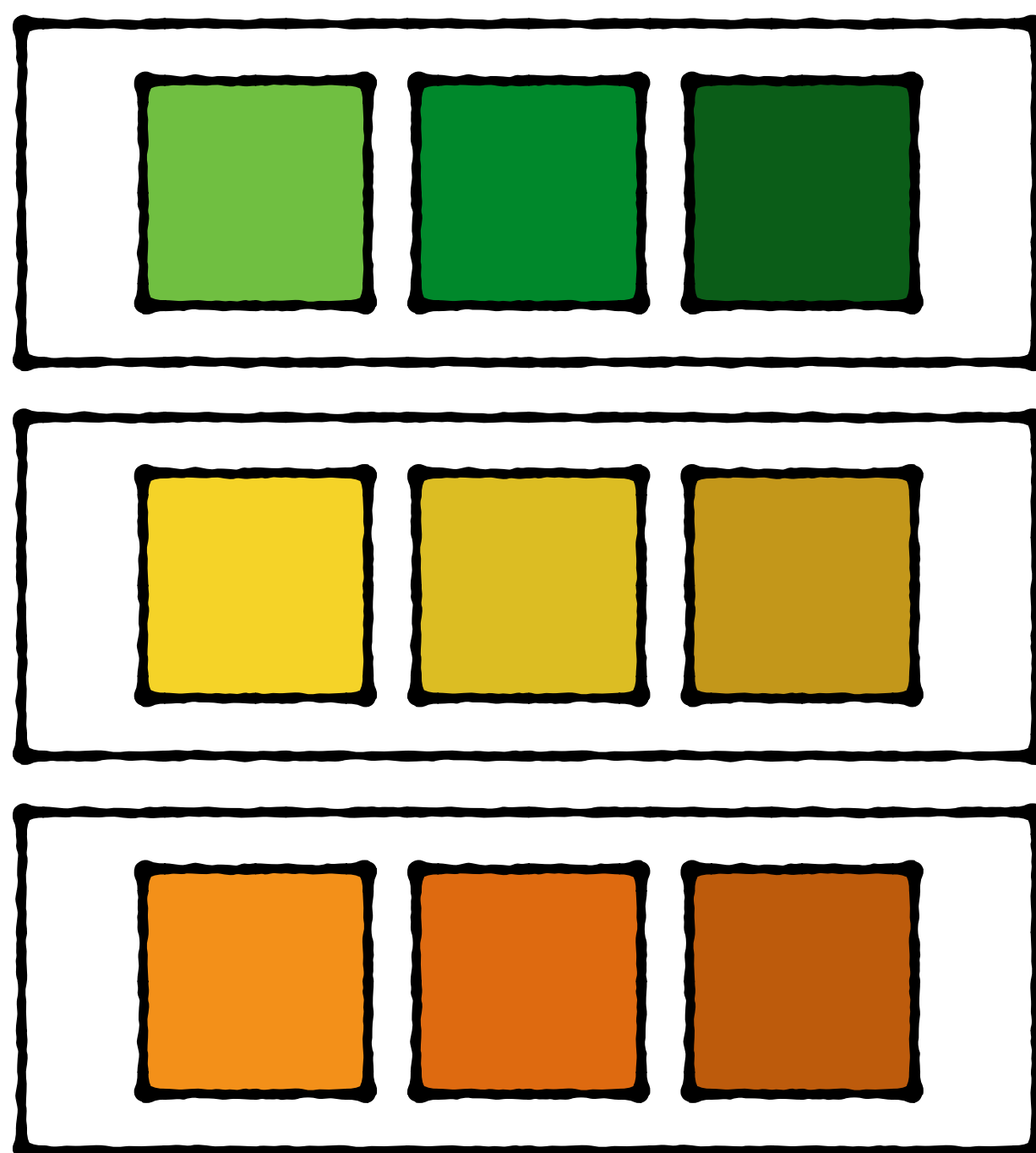
БЭМ-дерево

Библиотеки

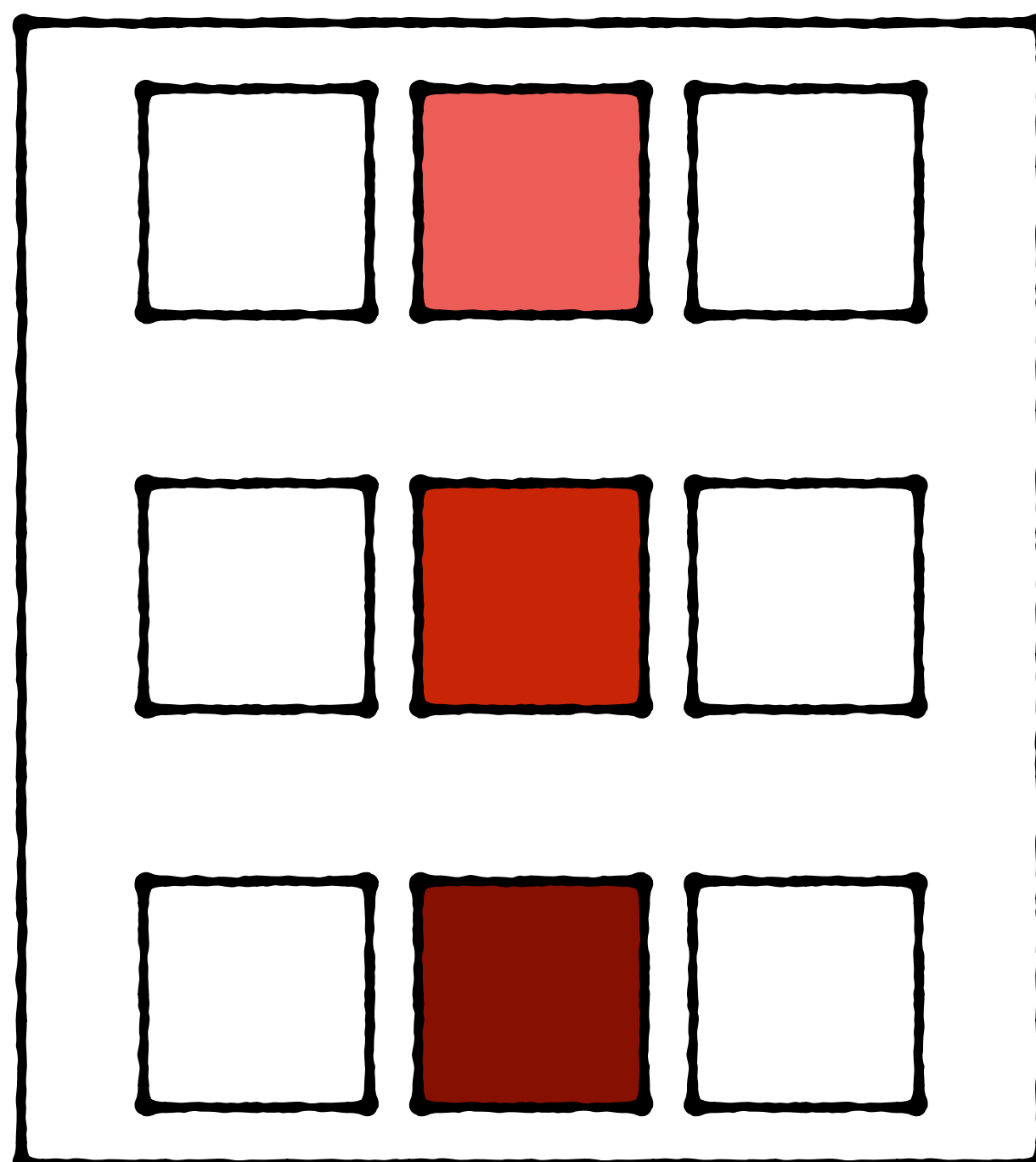


БЭМ-дерево

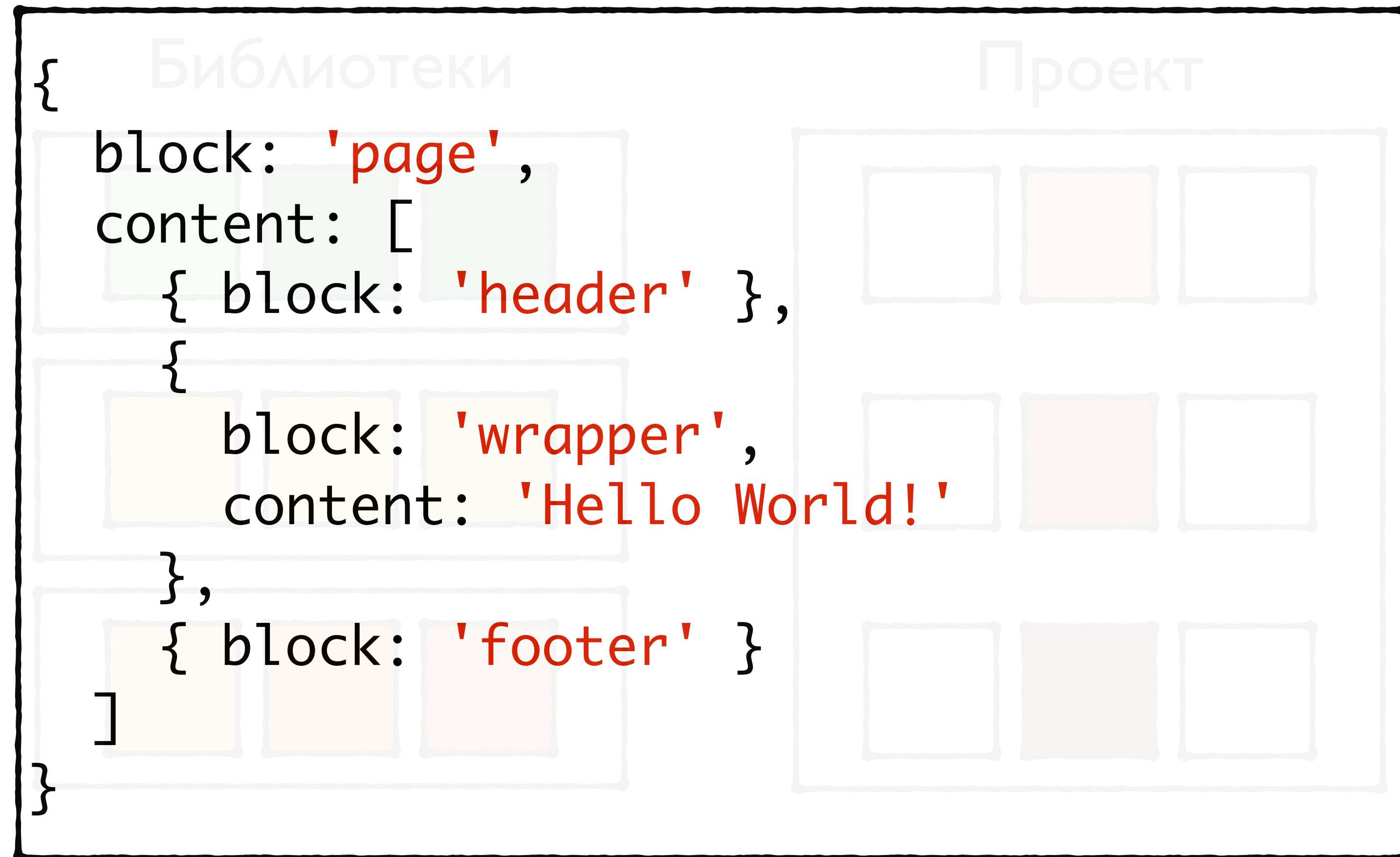
Библиотеки



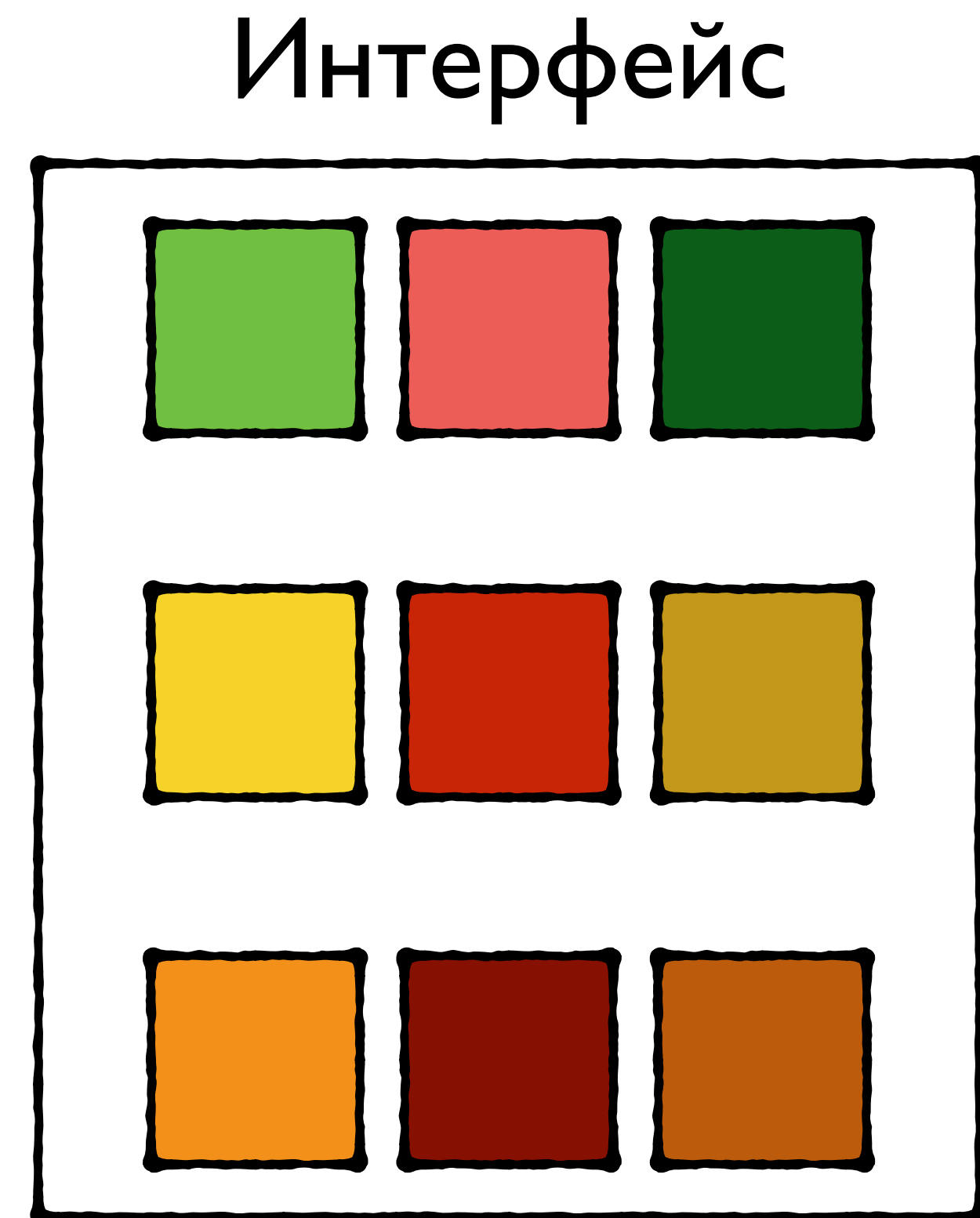
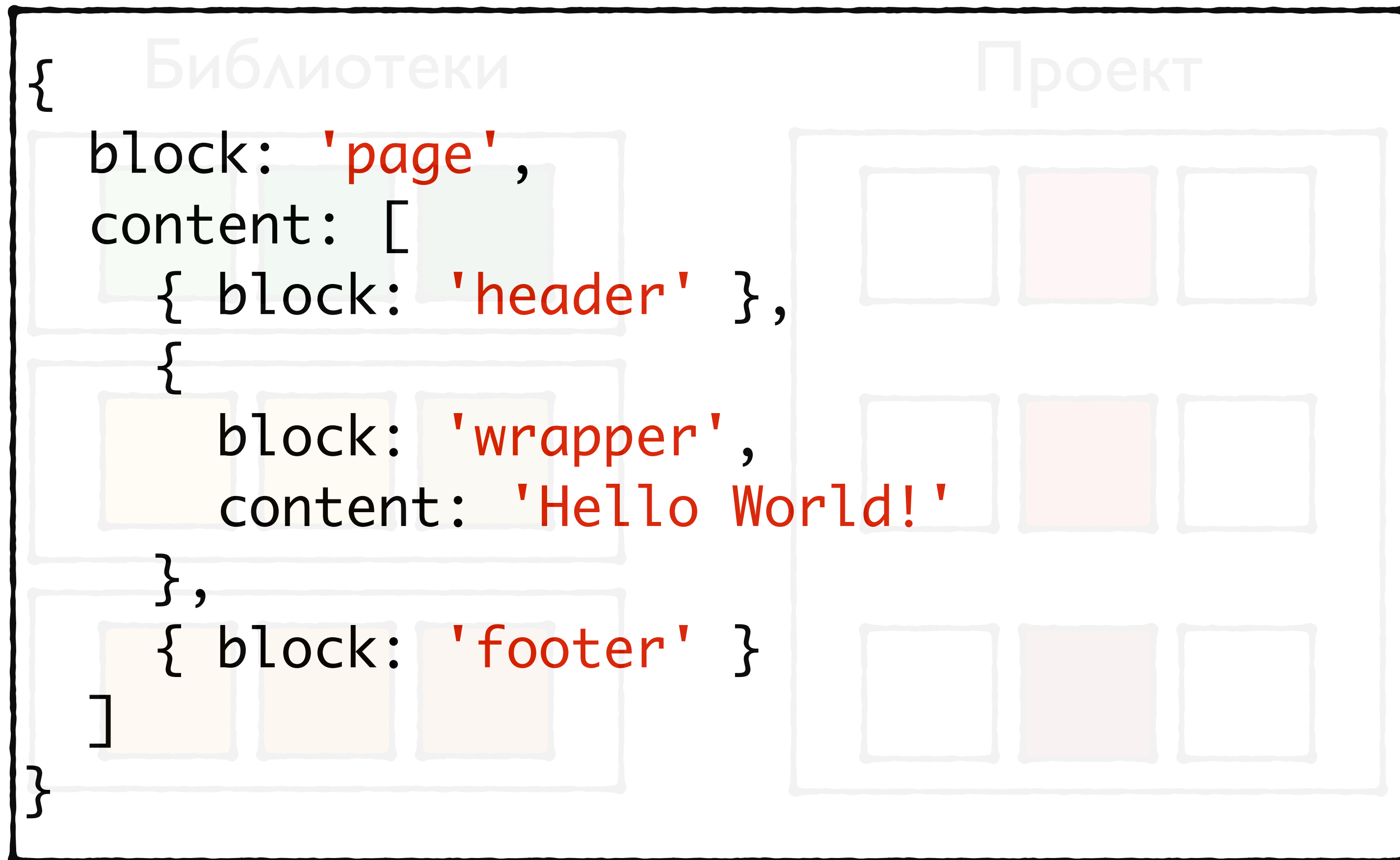
Проект



БЭМ-дерево



БЭМ-дерево




```

1  /**
2   * @module input
3   */
4  modules.define('input', ['i-bem__dom',
5
6  /**
7   * @exports
8   * @class input
9   * @augments base-control
10  * @bem
11  */
12  provide(BEMDOM.decl({ block : this.name,
13    onSetMod : {
14      'js' : {
15        'init' : function() {
16          this.__base.apply(this, arguments);
17          this._val = this.elem('input').val();
18        }
19      }
20    },
21  },
22
23  /**
24   * Returns control value
25   * @returns {String}
26   * @override
27   */
28  getVal : function() {
29    return this._val;
30  },
31
32  /**
33   * Sets control value
34   * @param {String} val value
35   * @param {Object} [data] additional data
36   * @returns {this}
37   */
38  setVal : function(val, data) {
39    val = String(val);
40
41    if(this._val !== val) {
42      this._val = val;
43
44      var control = this.elem('input');
45      control.val() !== val && control.val(val);
46
47      this.emit('change', data);
48    }
49
50    return this;

```

```

1  module.exports = function(bh) {
2
3    bh.match('input', function(ctx, json) {
4      ctx.tag('span')
5
6      .js(true)
7
8      .param('id', ctx.generateId());
9
10     .tParam('_input', json);
11
12     var content = ctx.content();
13     if(typeof content === 'undefined') {
14       content = [{ elem : 'control' }];
15       // NOTE: не вынесли в отдельный блок
16       json.label && content.unshift({ elem : 'label', content : ctx.content(content) });
17     }
18   });
19
20   };

```

```

1  modules.define('input', ['tick', 'idle',
2
3  var instances = [],
4  boundToTick,
5  bindToTick = function() {
6    boundToTick = true;
7    tick
8      .on('tick', update)
9      .start();
10  idle
11    .on({
12      idle : function() {
13        tick.un('tick', update);
14      },
15      wakeup : function() {
16        tick.on('tick', update);
17      }
18    })
19    .start();
20  },
21  update = function() {
22    var instance, i = 0;
23    while(instance = instances[i++]) {
24      instance.setVal(instance.elem('input').val());
25    }
26  };
27
28  provide(Input.decl({
29    onSetMod : {
30      'js' : {
31        'init' : function() {
32          this.__base.apply(this, arguments);
33
34          boundToTick || bindToTick();
35
36          // сохраняем индекс в массиве
37          this._instanceIndex = instances.length;
38        },
39
40        'update' : function() {
41          this.__base.apply(this, arguments);
42
43          // удаляем из общего массива
44          instances.splice(this._instanceIndex, 1);
45          // понижаем _instanceIndex
46          var i = this._instanceIndex;
47          while(instance = instances[i]) {
48            instance._instanceIndex--;
49          }
50        },

```

```

1  /** @module base-control */
2
3  modules.define(
4    'base-control',
5    ['i-bem__dom', 'dom', 'next-tick',
6    function(provide, BEMDOM, dom, nextTick) {
7
8    /**
9     * @exports
10    * @class base-control
11    * @abstract
12    * @bem
13    */
14    provide(BEMDOM.decl(this.name, /** @lends BaseControl.prototype */
15      beforeSetMod : {
16        'focused' : {
17          'true' : function() {
18            return !this.hasMod('disabled');
19          }
20        }
21      },
22
23      onSetMod : {
24        'js' : {
25          'init' : function() {
26            this._focused = dom.hasClass(this.elem('input'), 'focused');
27            this._focused?
28              // if control is focused
29              this.setMod('focused')
30              // if block already has focused
31              this.hasMod('focused') :
32
33              this._tabIndex = this._instanceIndex;
34            if(this.hasMod('disabled')) {
35              this.elem('input').disabled = true;
36            }
37          },
38
39          'focused' : {
40            'true' : function() {
41              this._focused || this.setMod('focused');
42            },
43
44            'false' : function() {
45              this._focused && this.setMod('focused');
46            }
47          },
48
49          'disabled' : {
50            '*' : function(modName, modValue) {

```


Что мы имеем?



Что мы имеем?

- › Независимость блоков
- › Уровни переопределения
- › БЭМ — Дерево

Требования для работы с файловой системой

Работать с кодом блока со всех уровней

Работать с кодом блока со всех уровней

Уровни

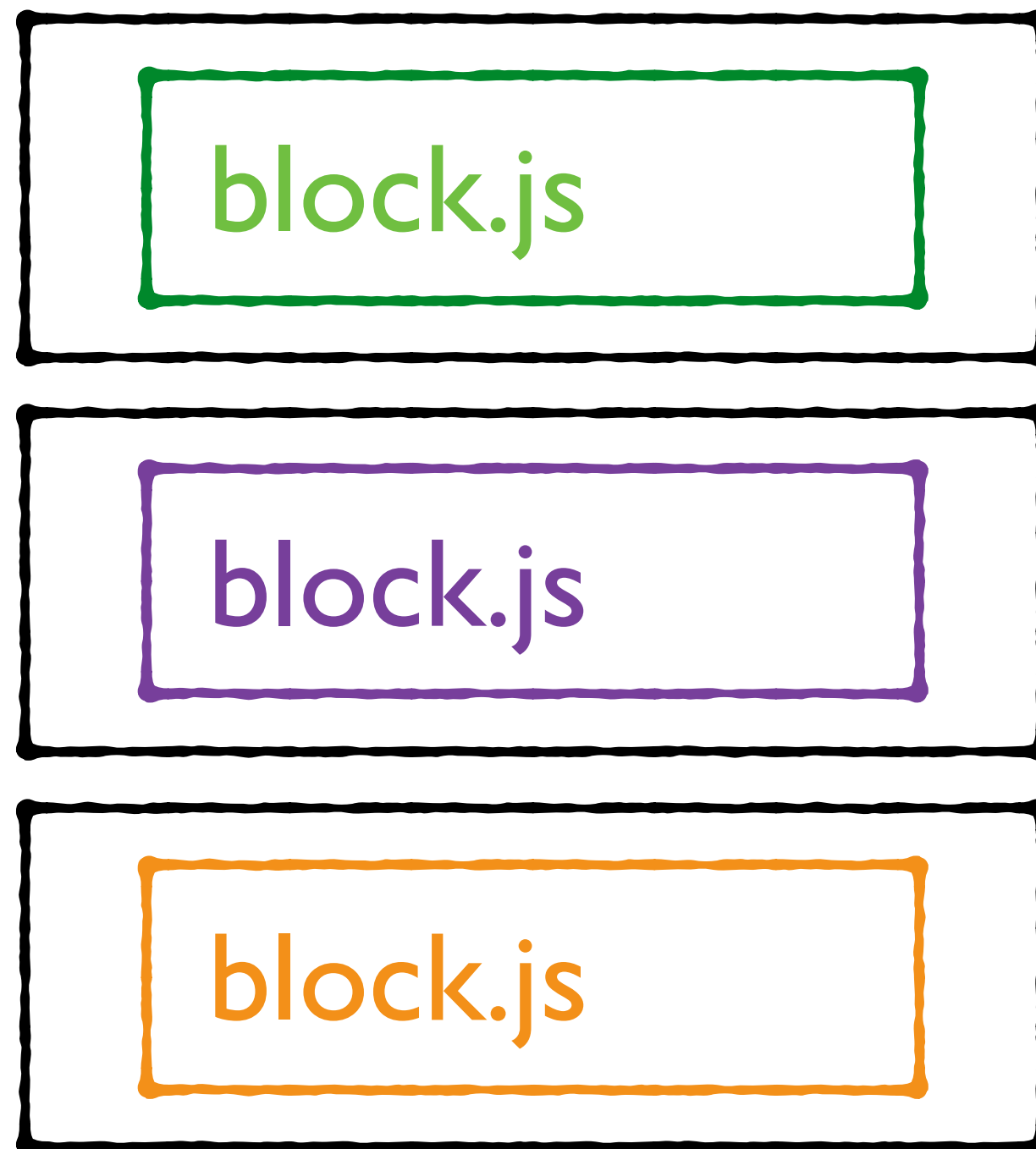
block.js

block.js

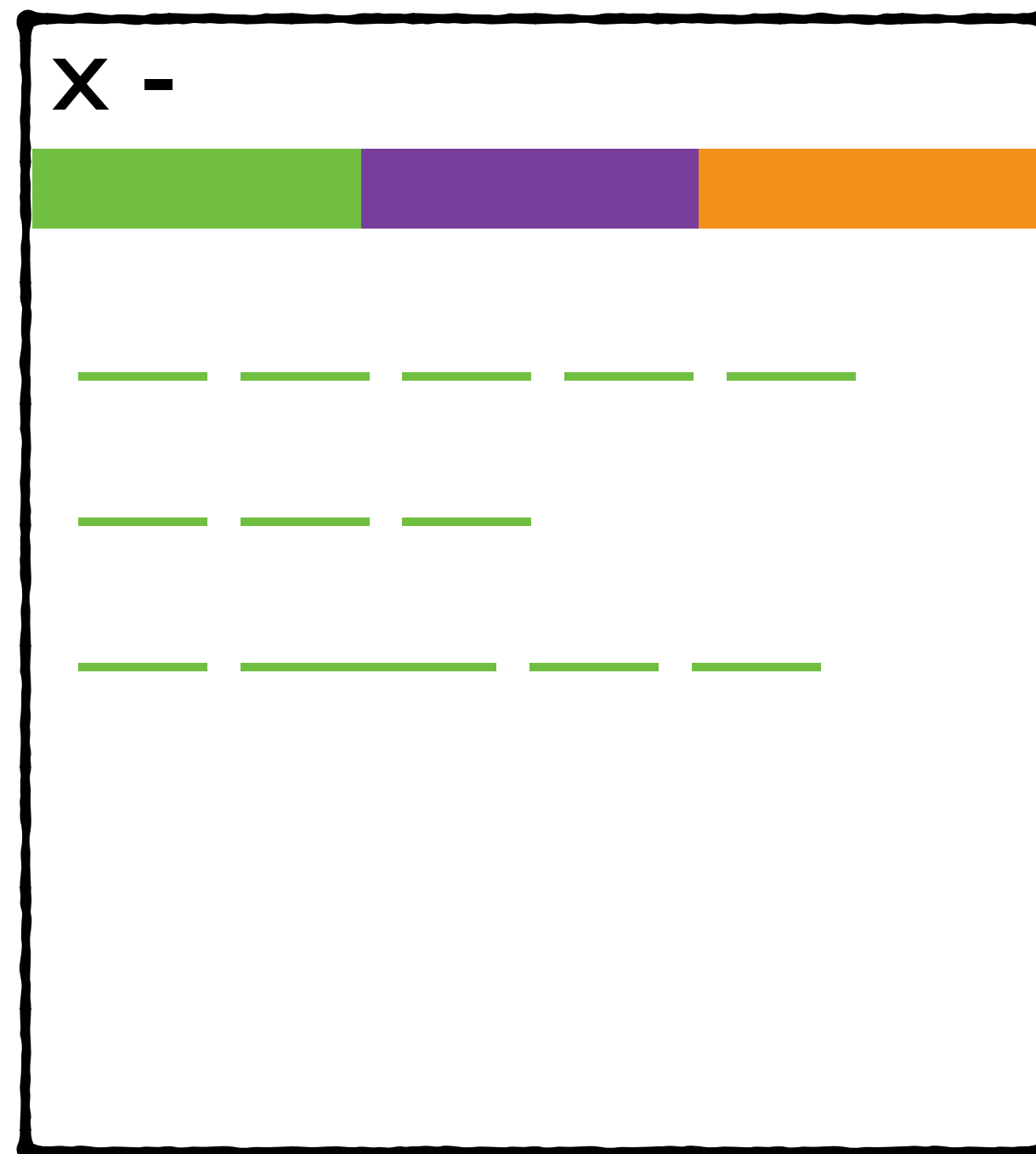
block.js

Работать с кодом блока со всех уровней

Уровни

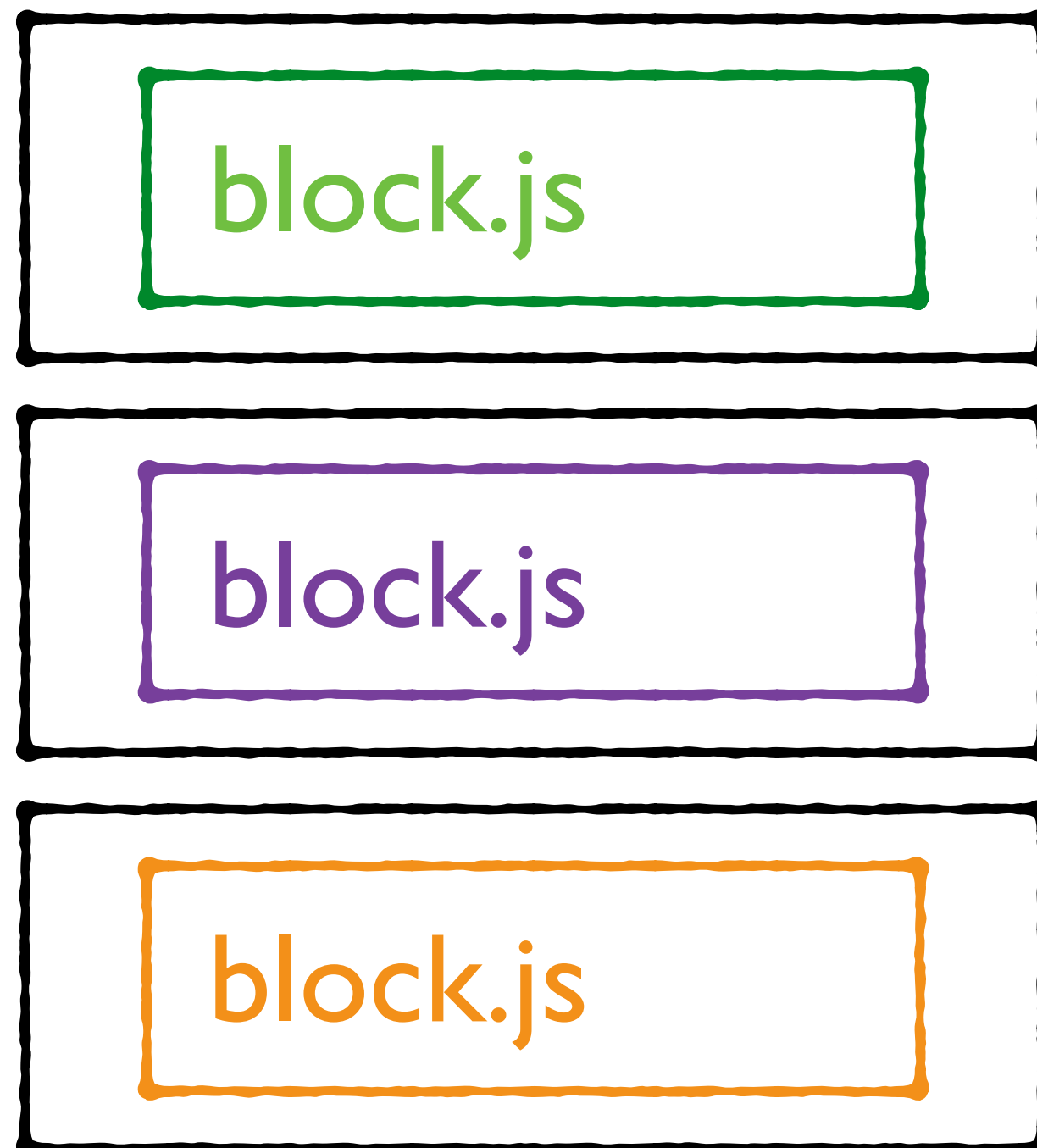


Редактор

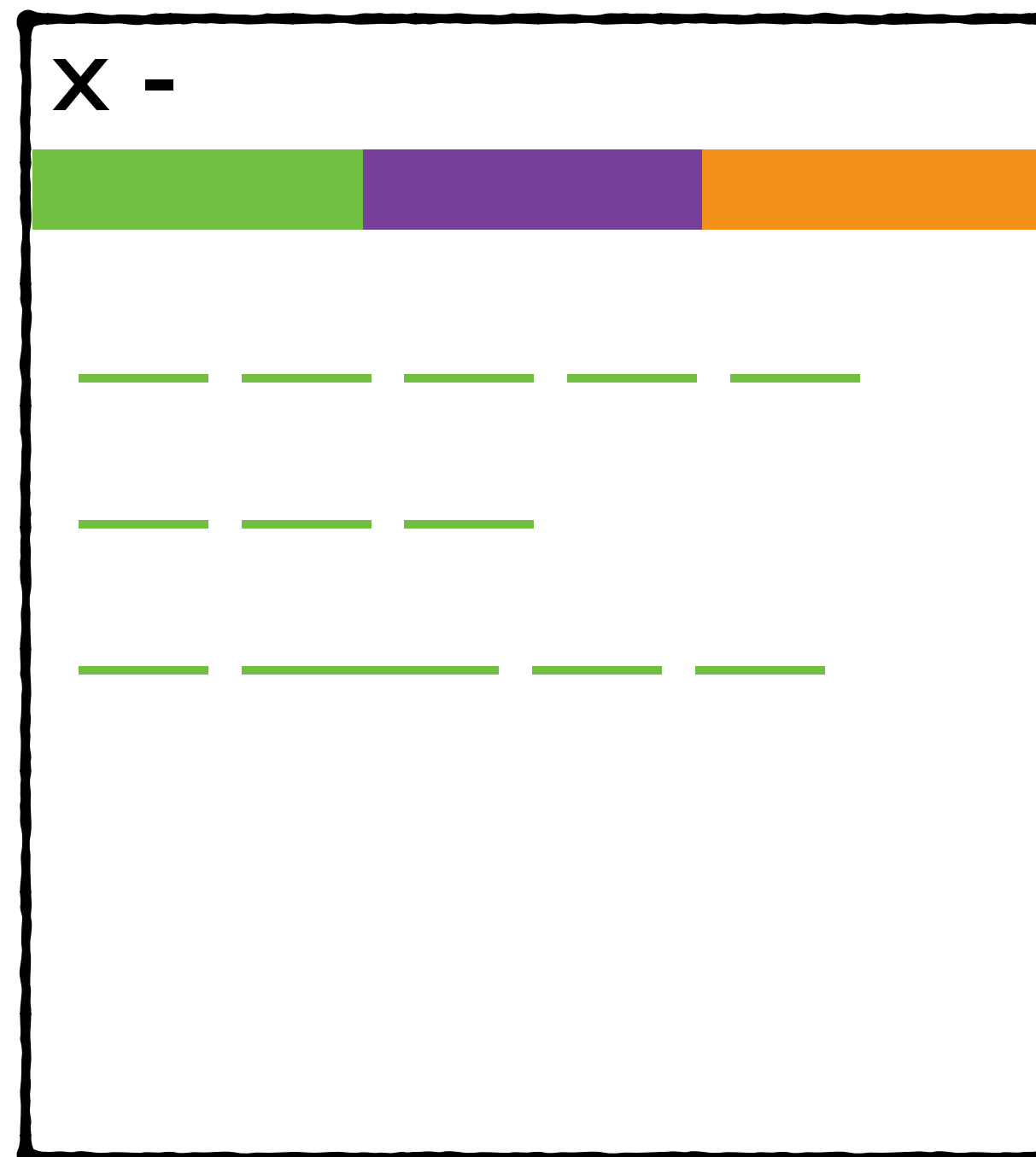


Работать с кодом блока со всех уровней

Уровни



Редактор



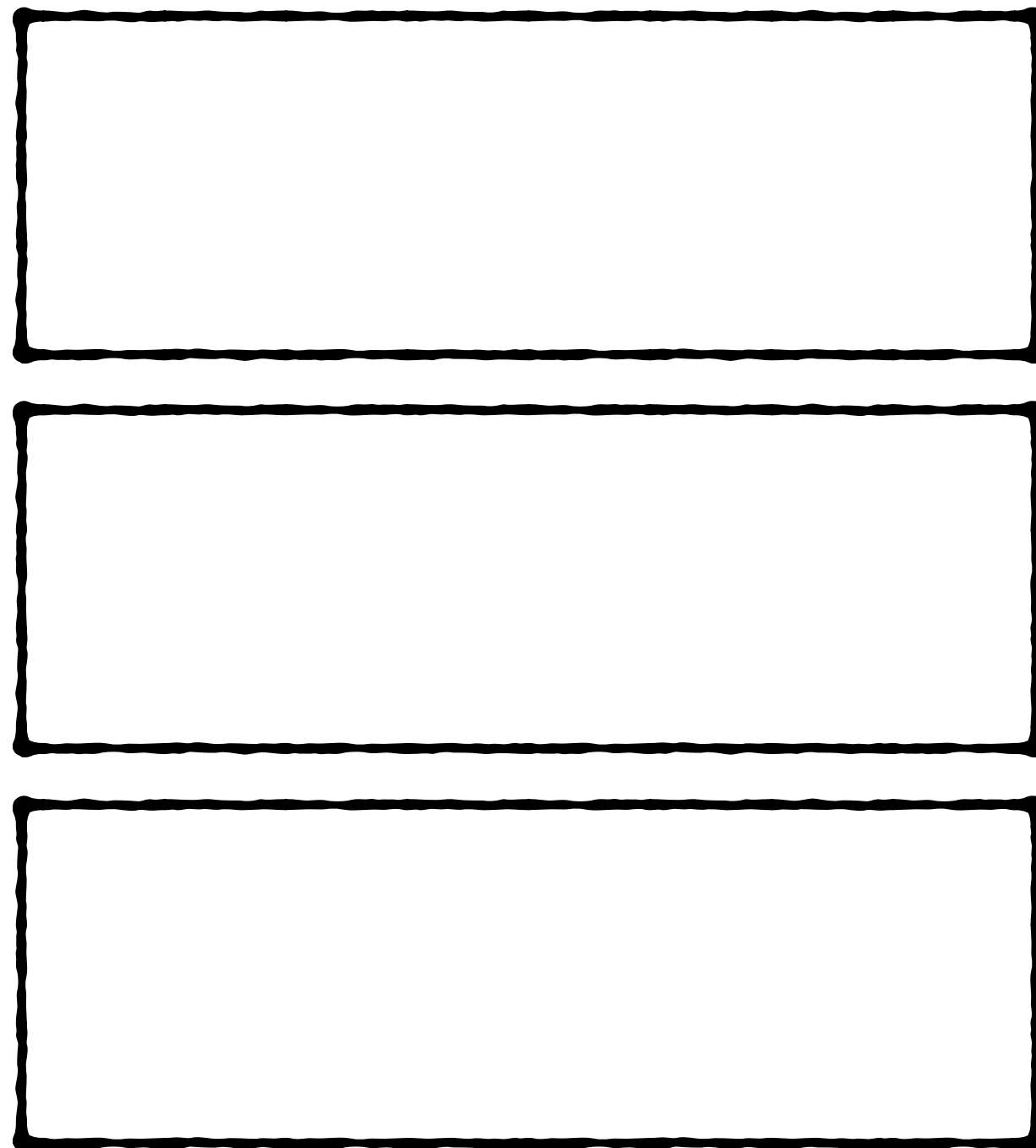
BEM IDE



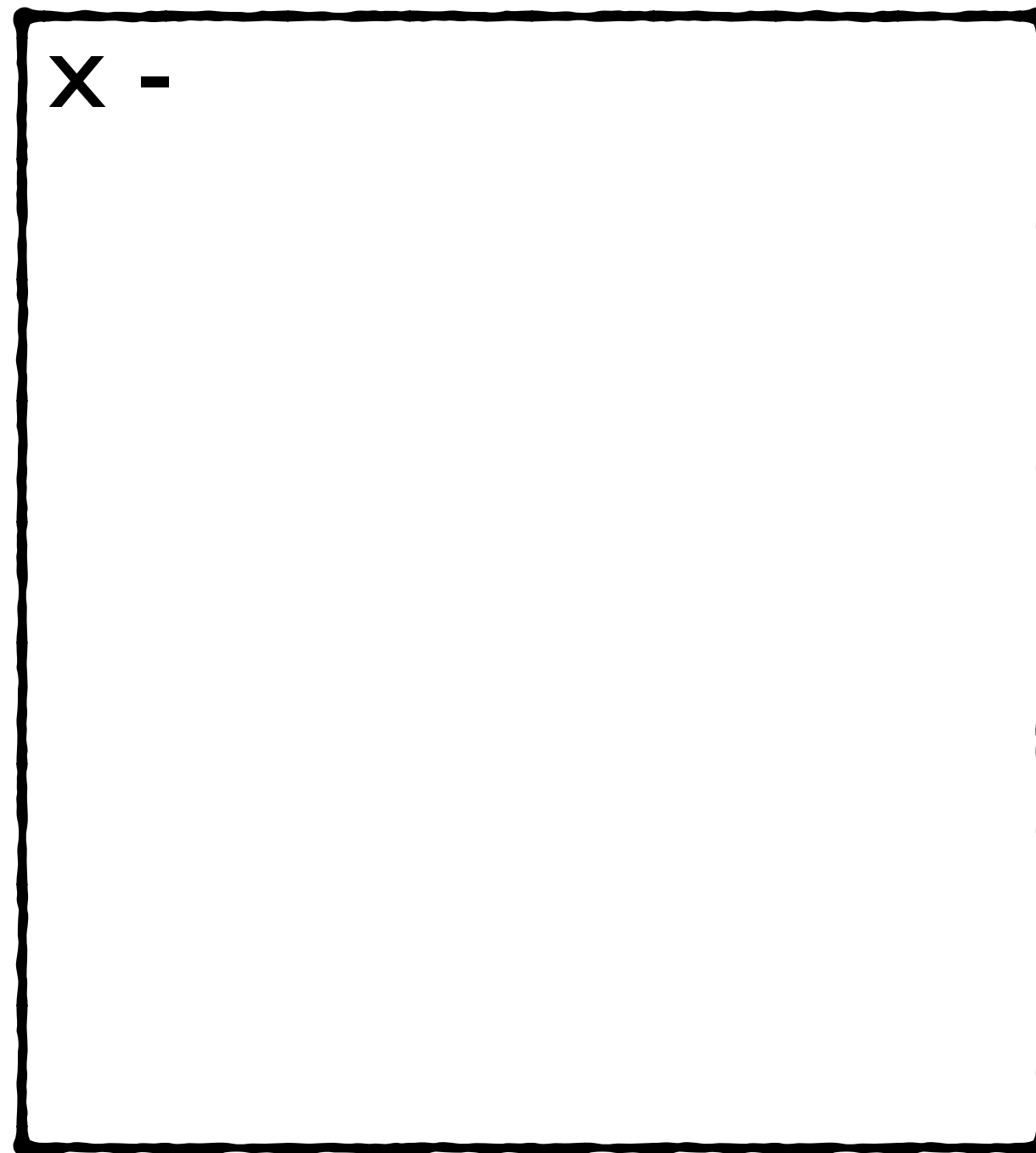
Создавать блоки, а не файлы и папки

Создавать блоки, а не файлы и папки

Уровни

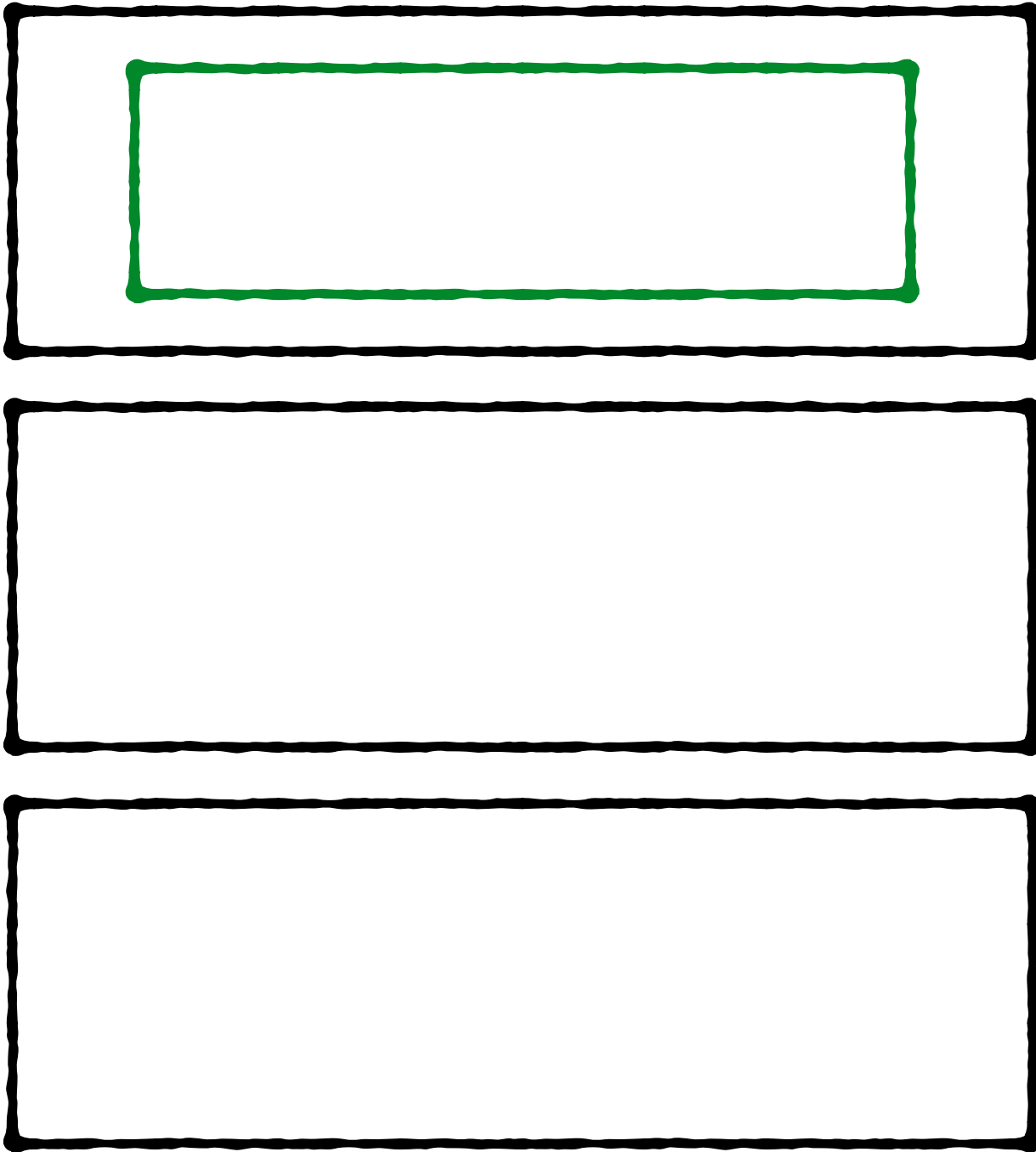


Редактор

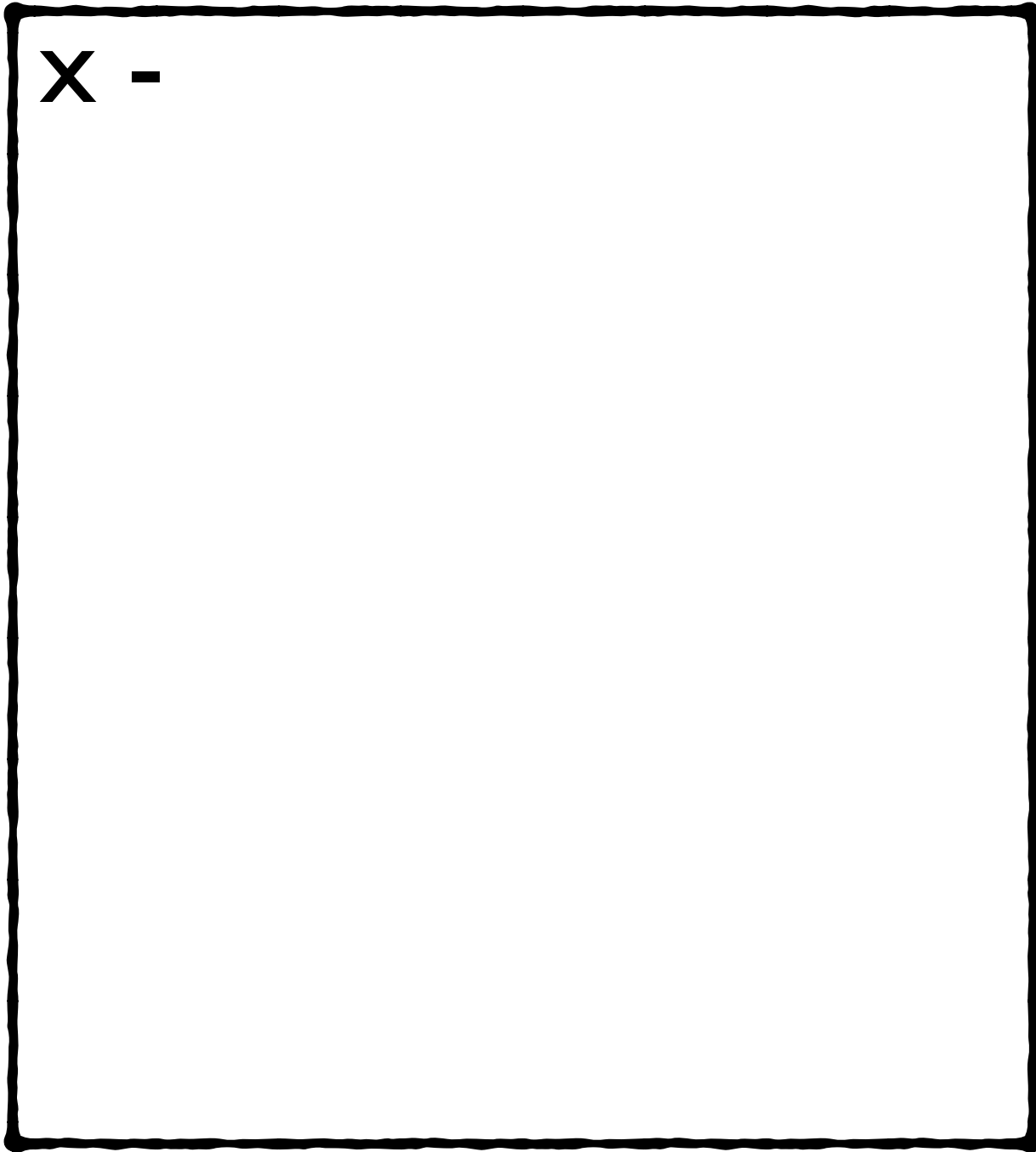


Создавать блоки, а не файлы и папки

Уровни

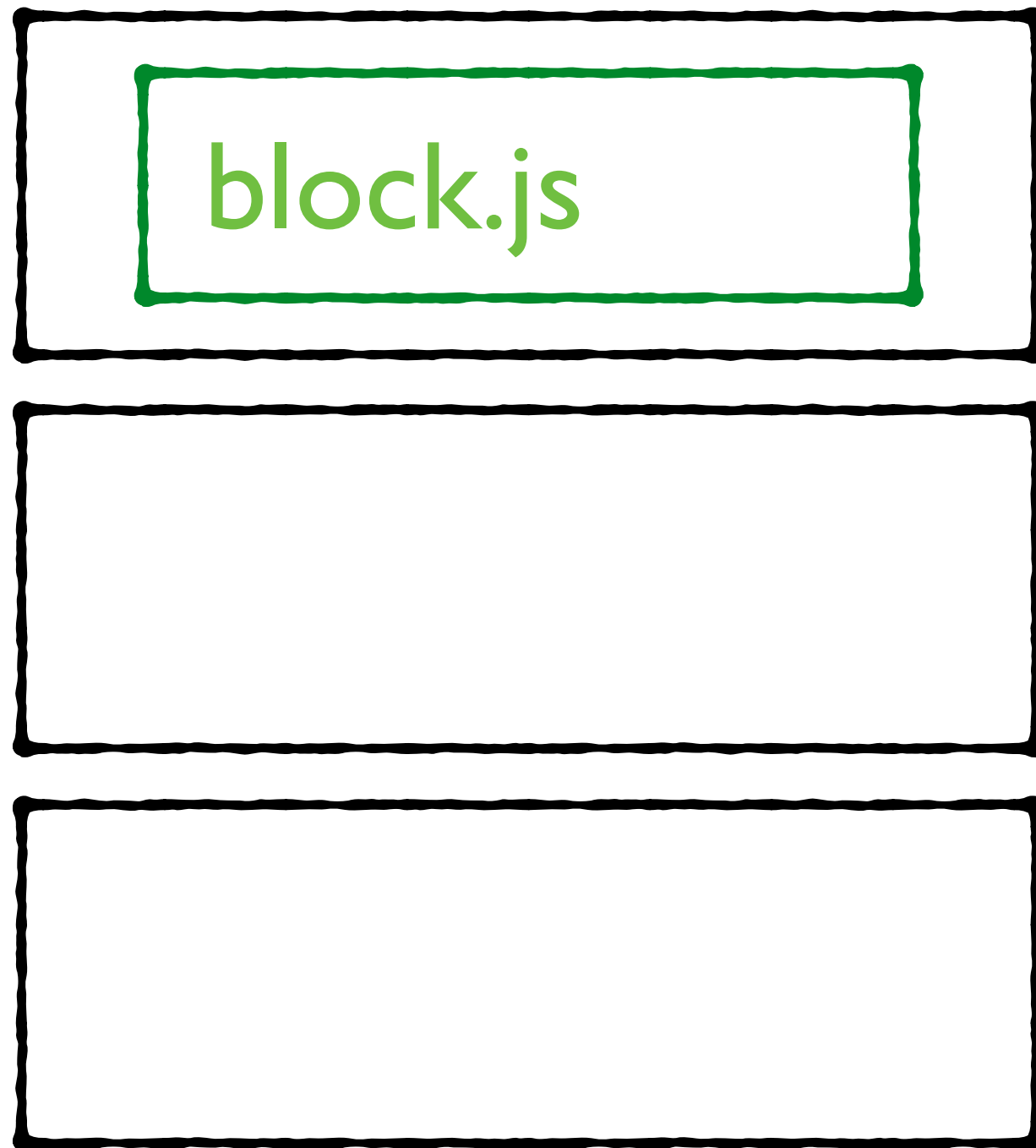


Редактор

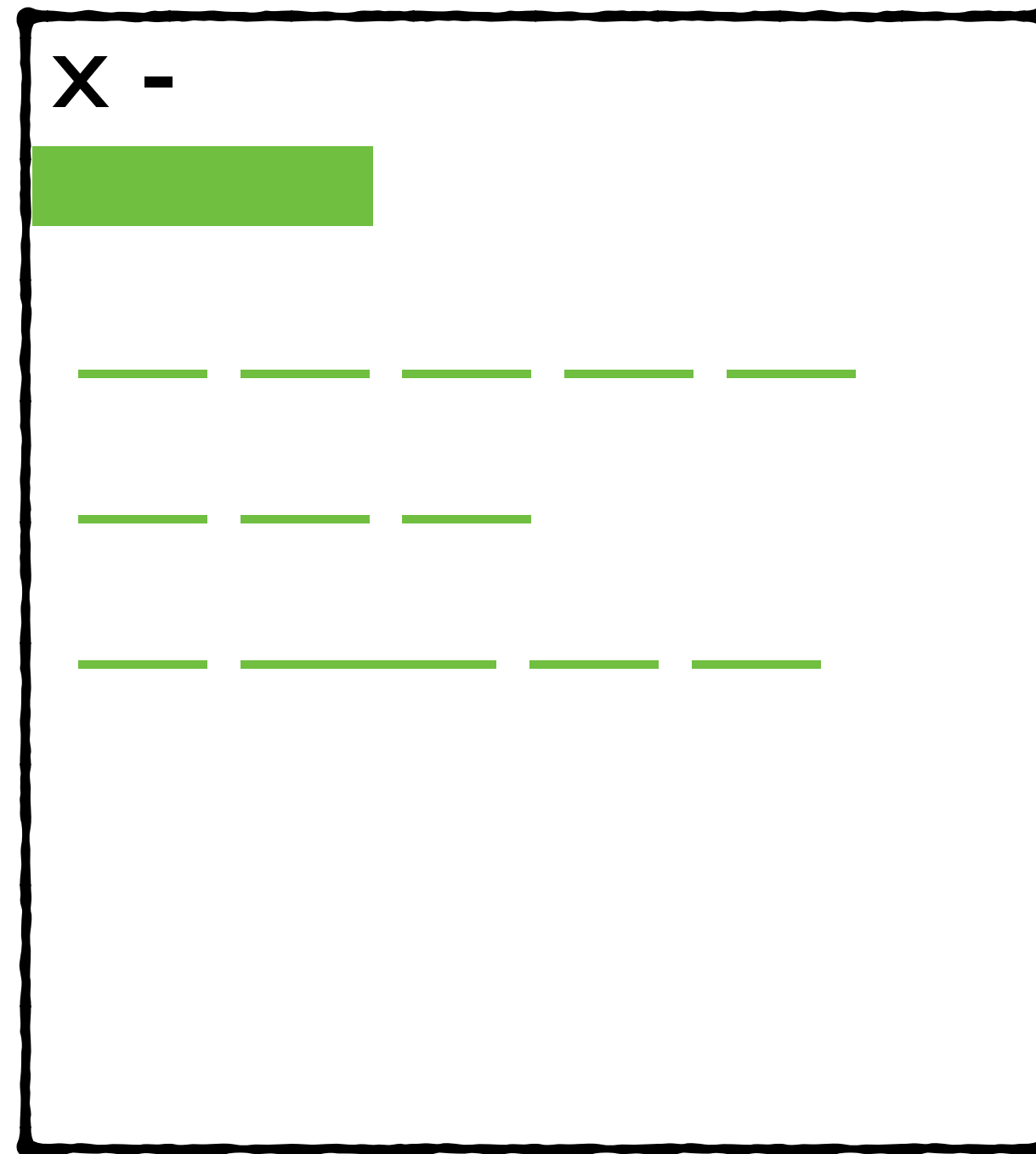


Создавать блоки, а не файлы и папки

Уровни

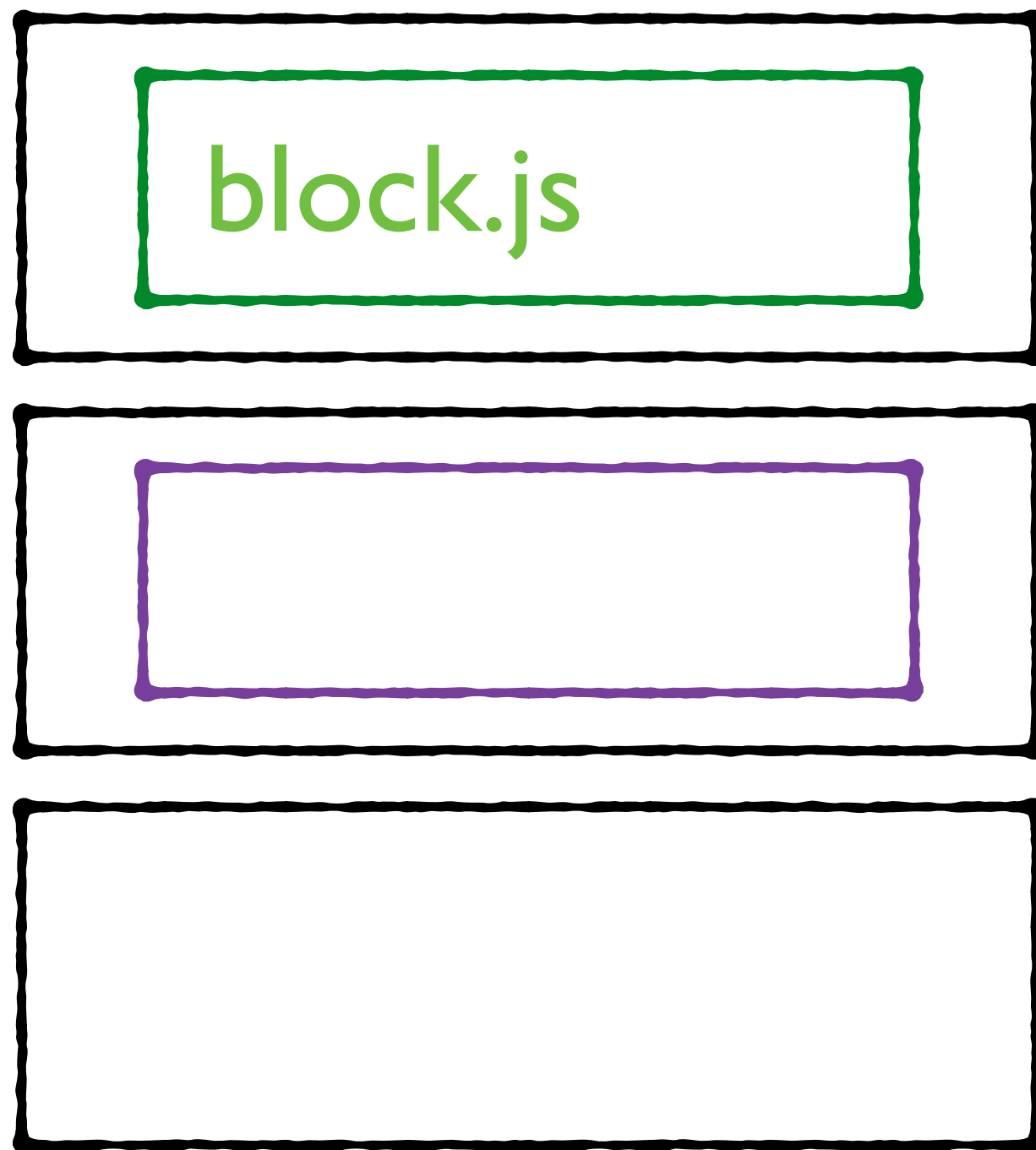


Редактор

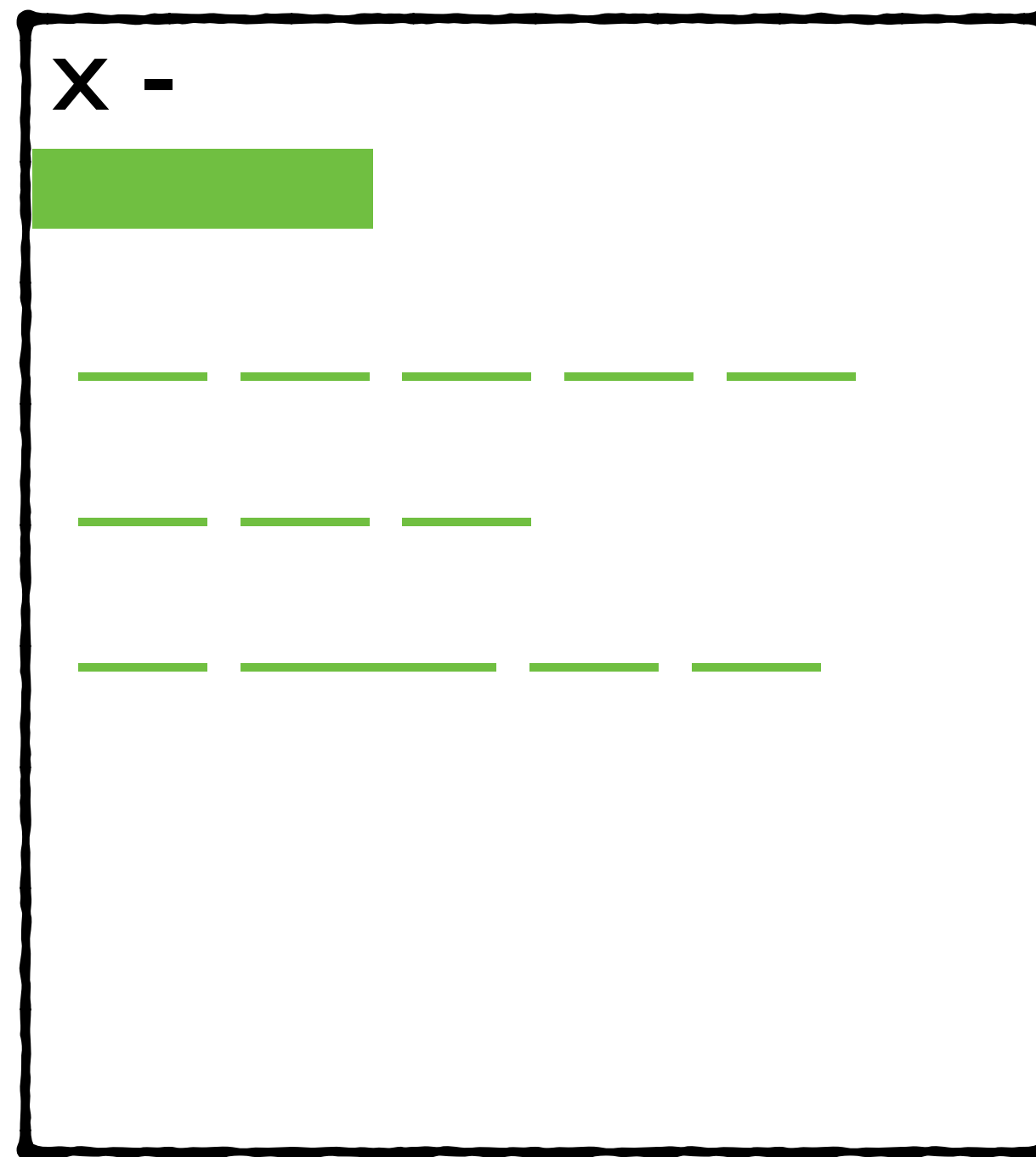


Создавать блоки, а не файлы и папки

Уровни

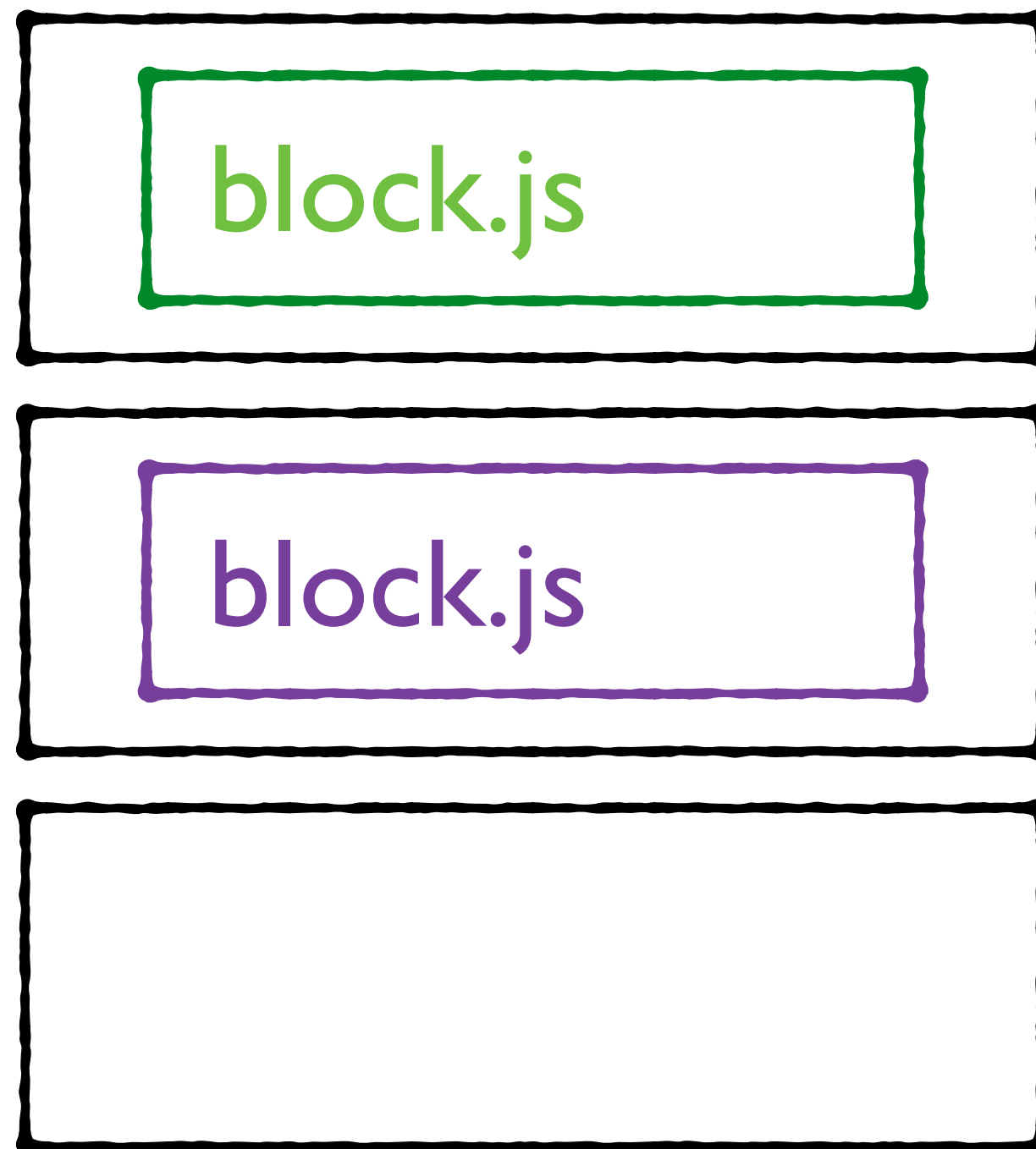


Редактор



Создавать блоки, а не файлы и папки

Уровни

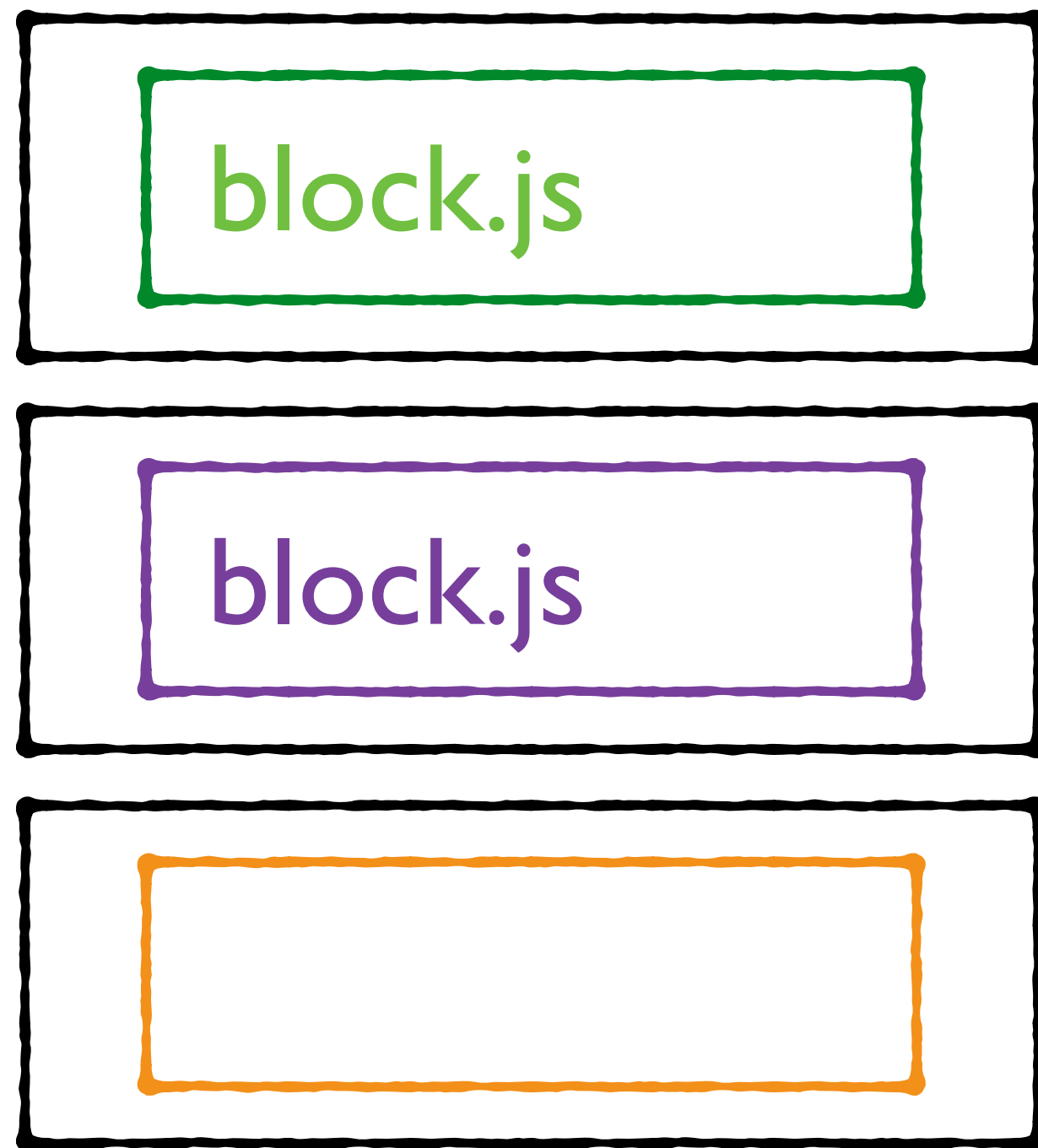


Редактор

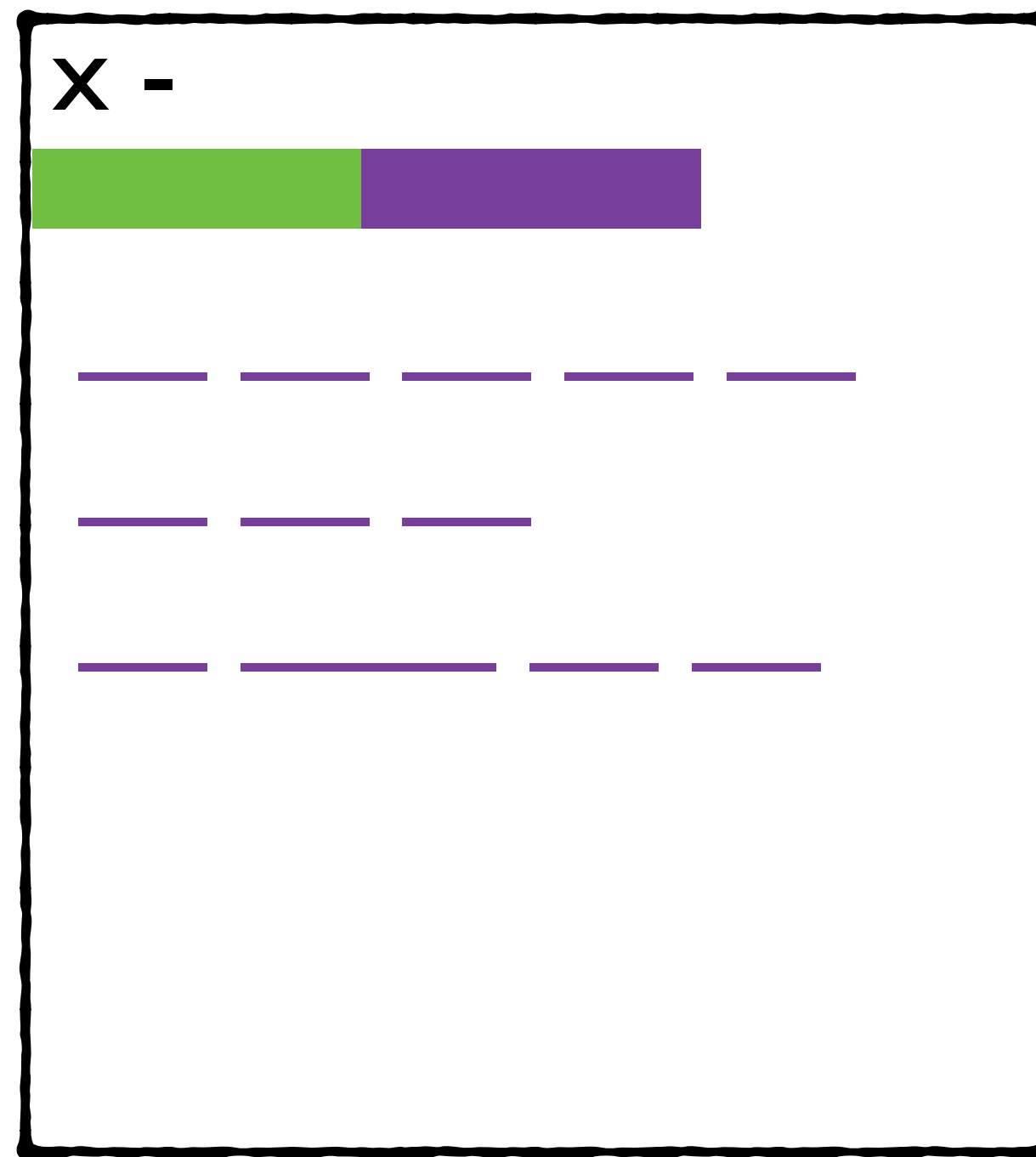


Создавать блоки, а не файлы и папки

Уровни

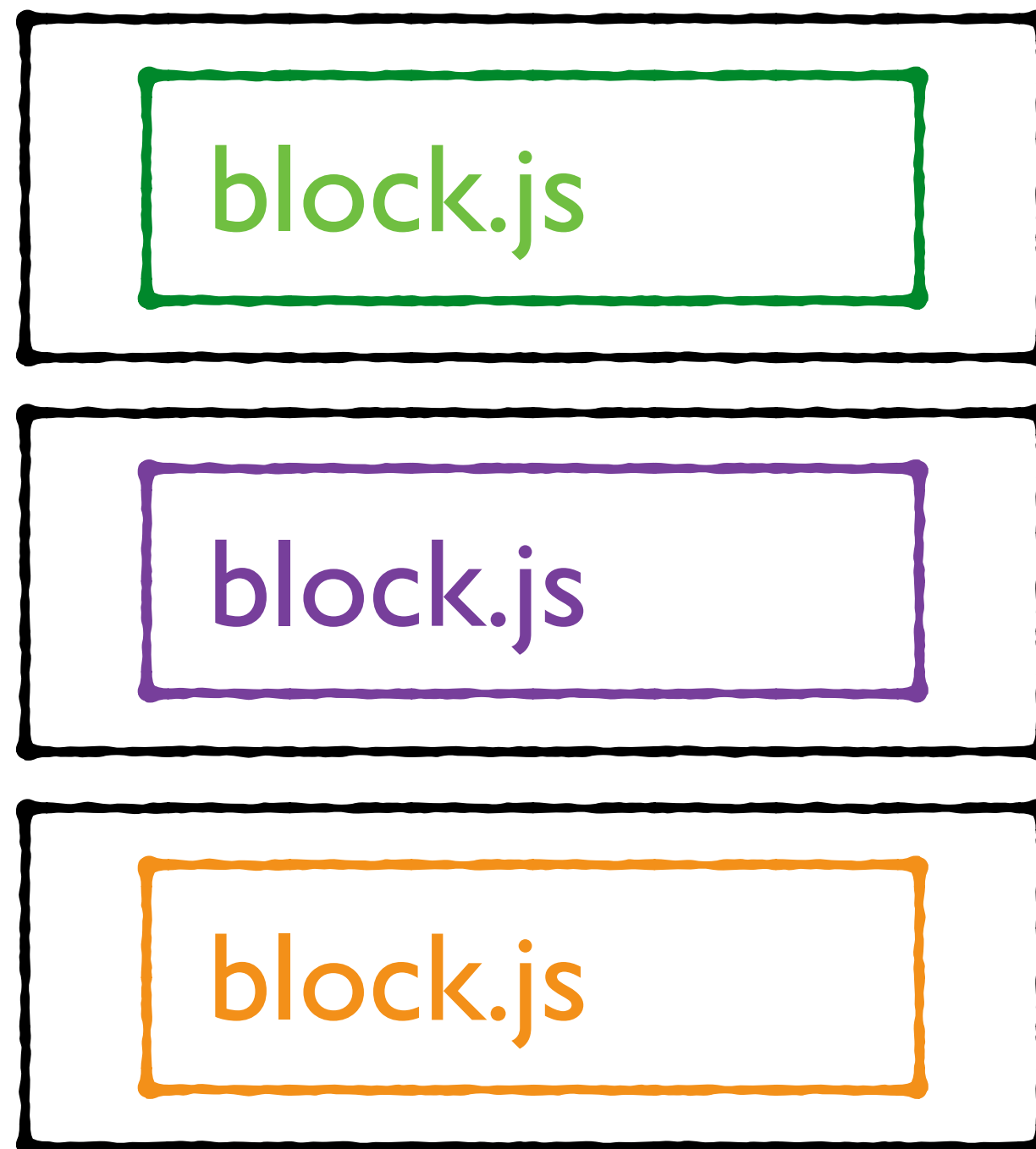


Редактор



Создавать блоки, а не файлы и папки

Уровни

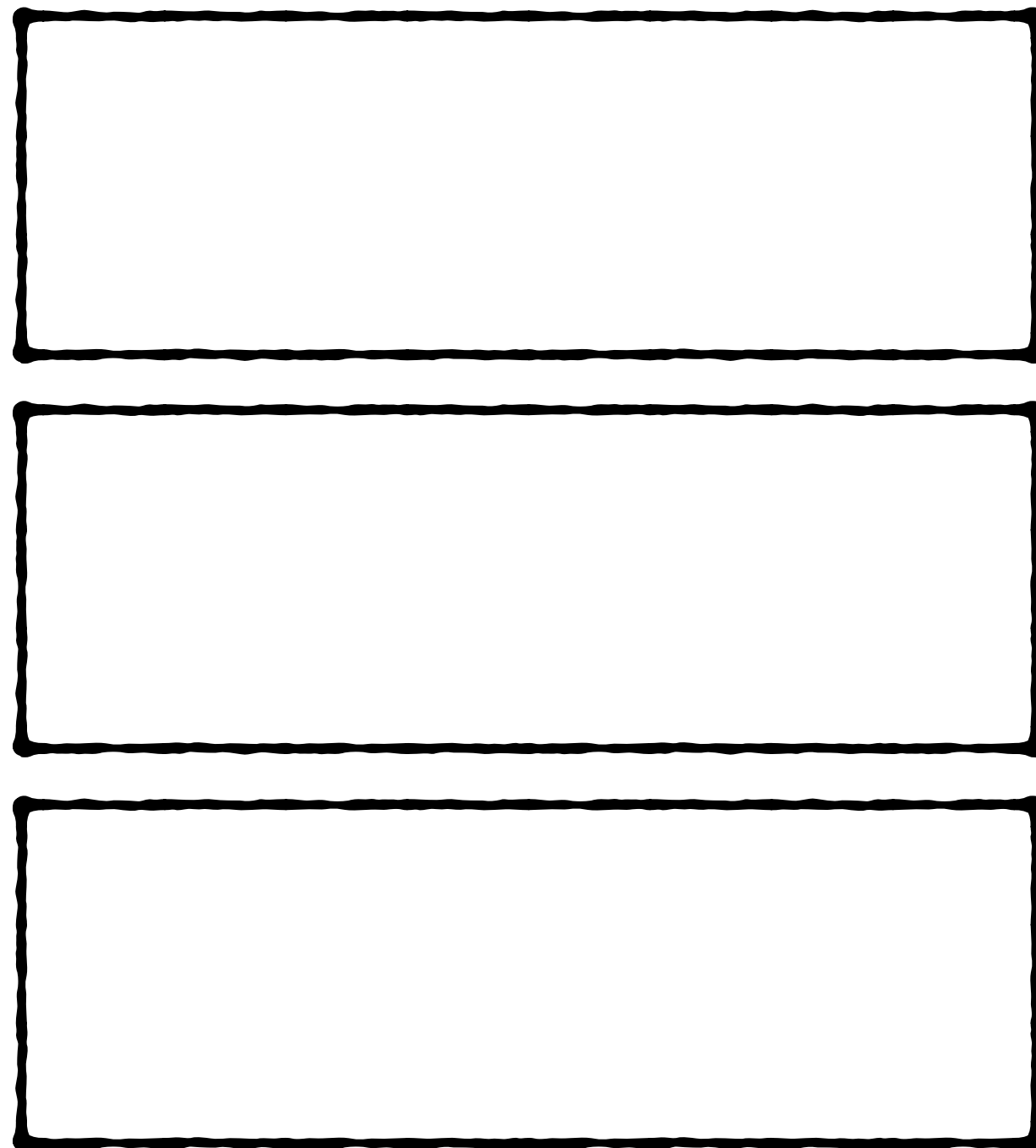


Редактор

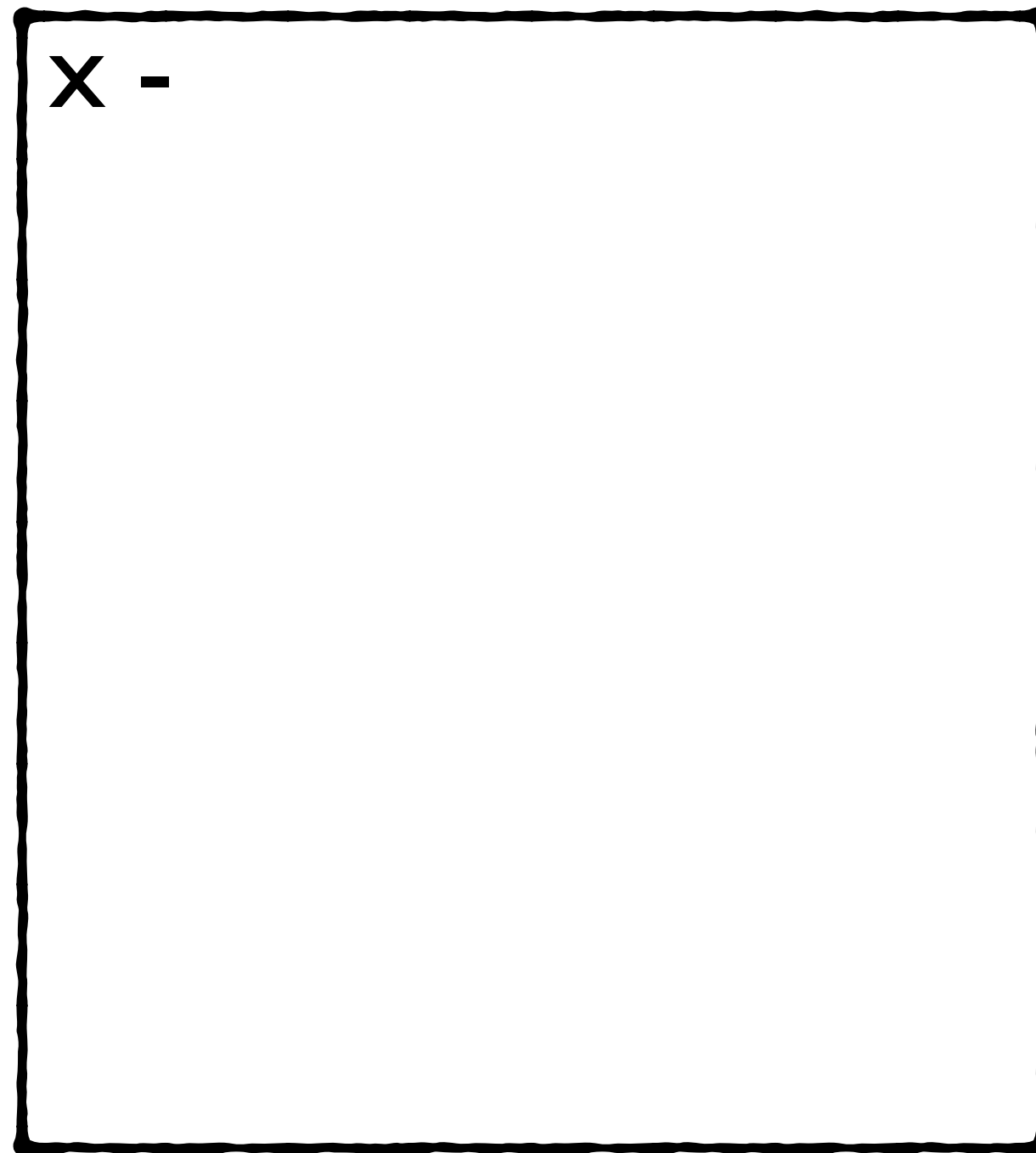


Создавать блоки, а не файлы и папки

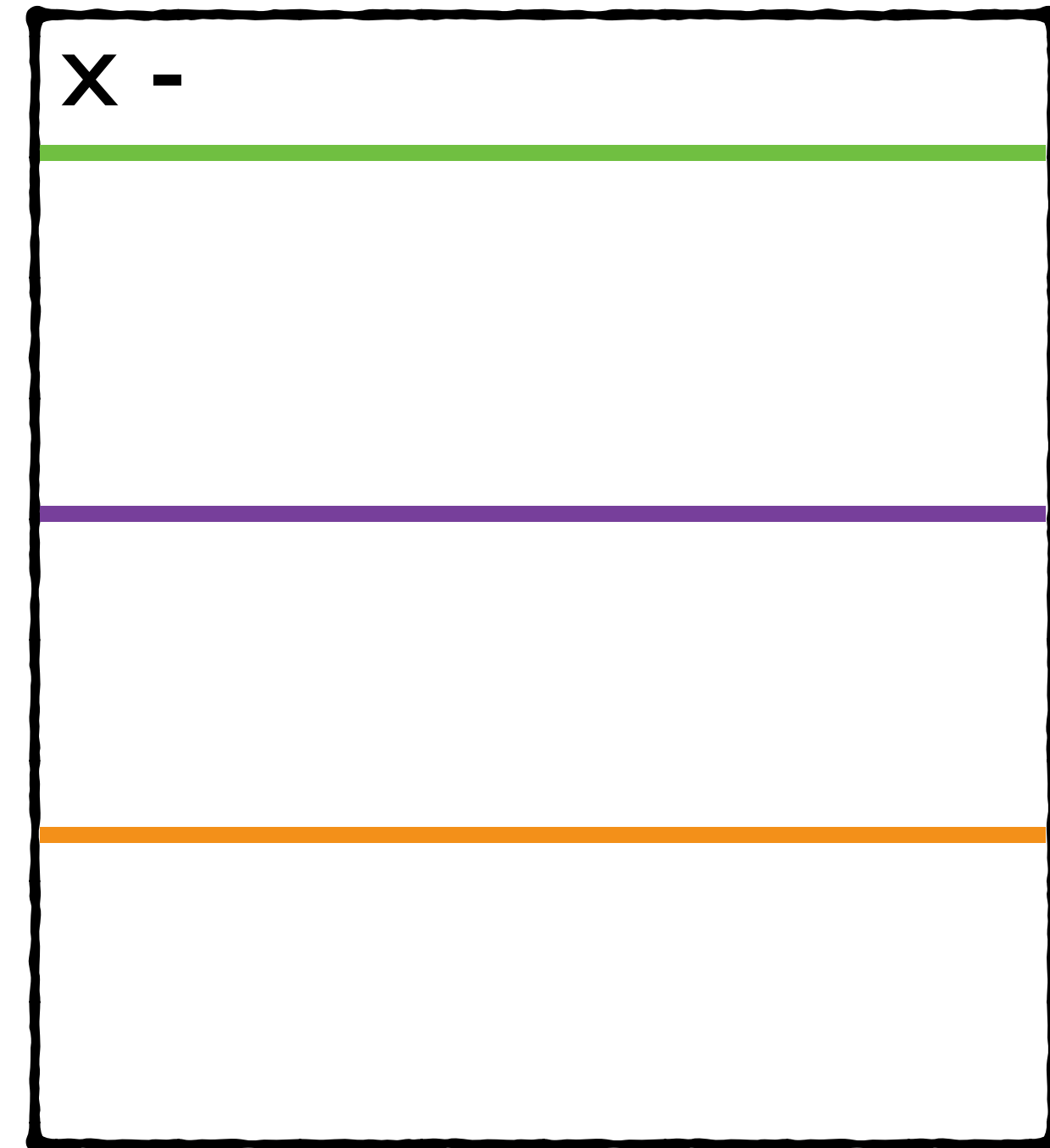
Уровни



Редактор

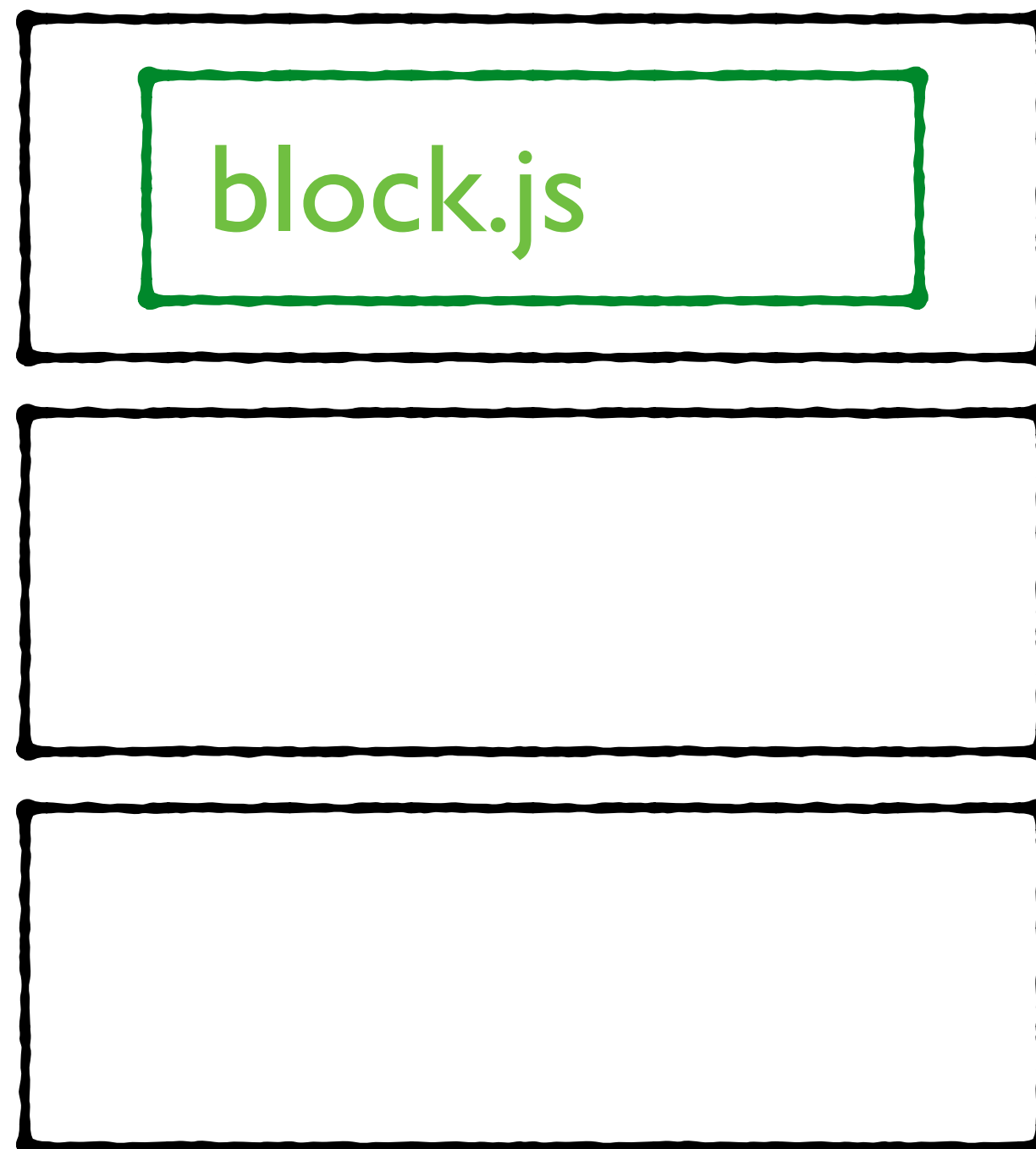


ВЕМ IDE

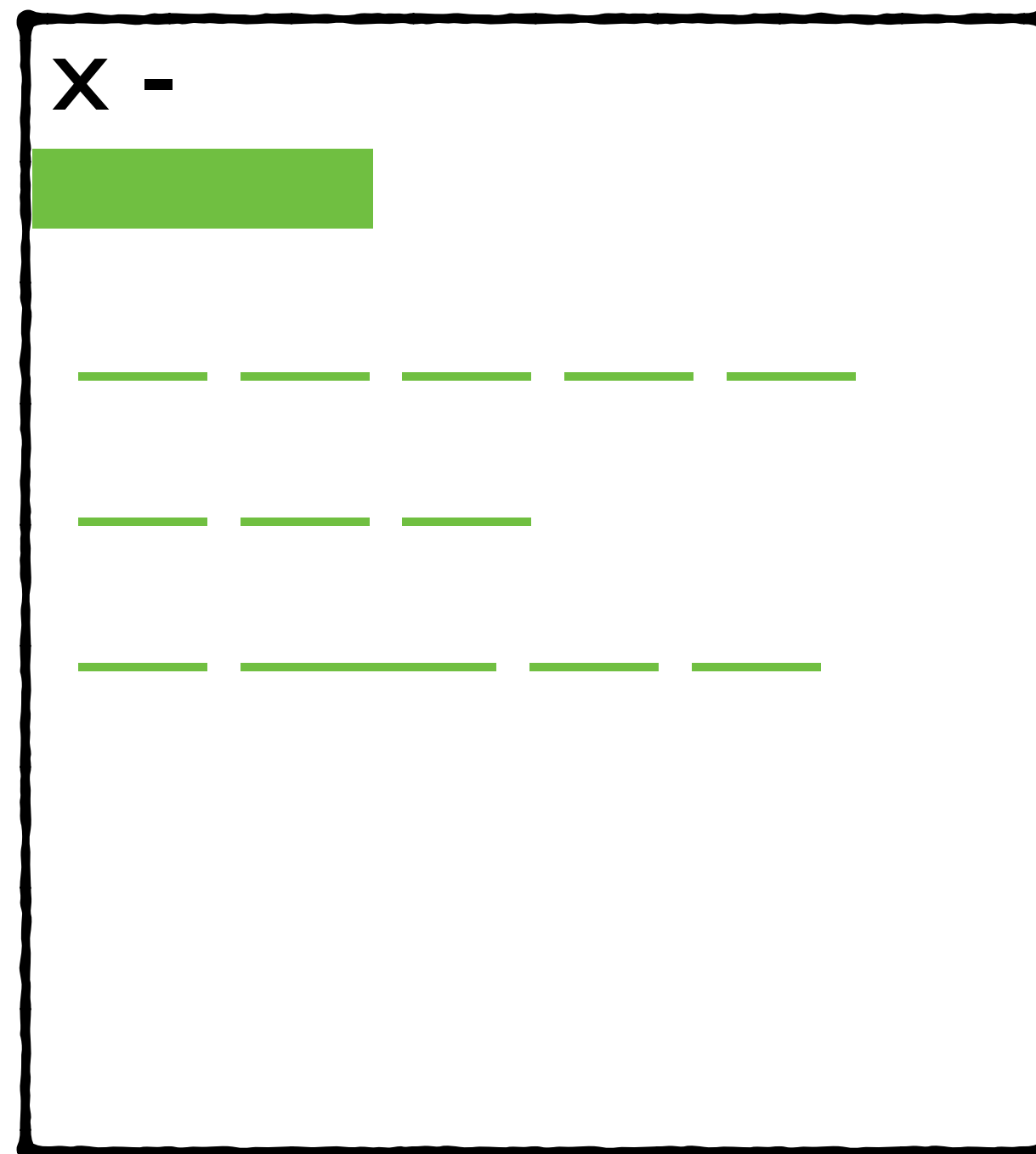


Создавать блоки, а не файлы и папки

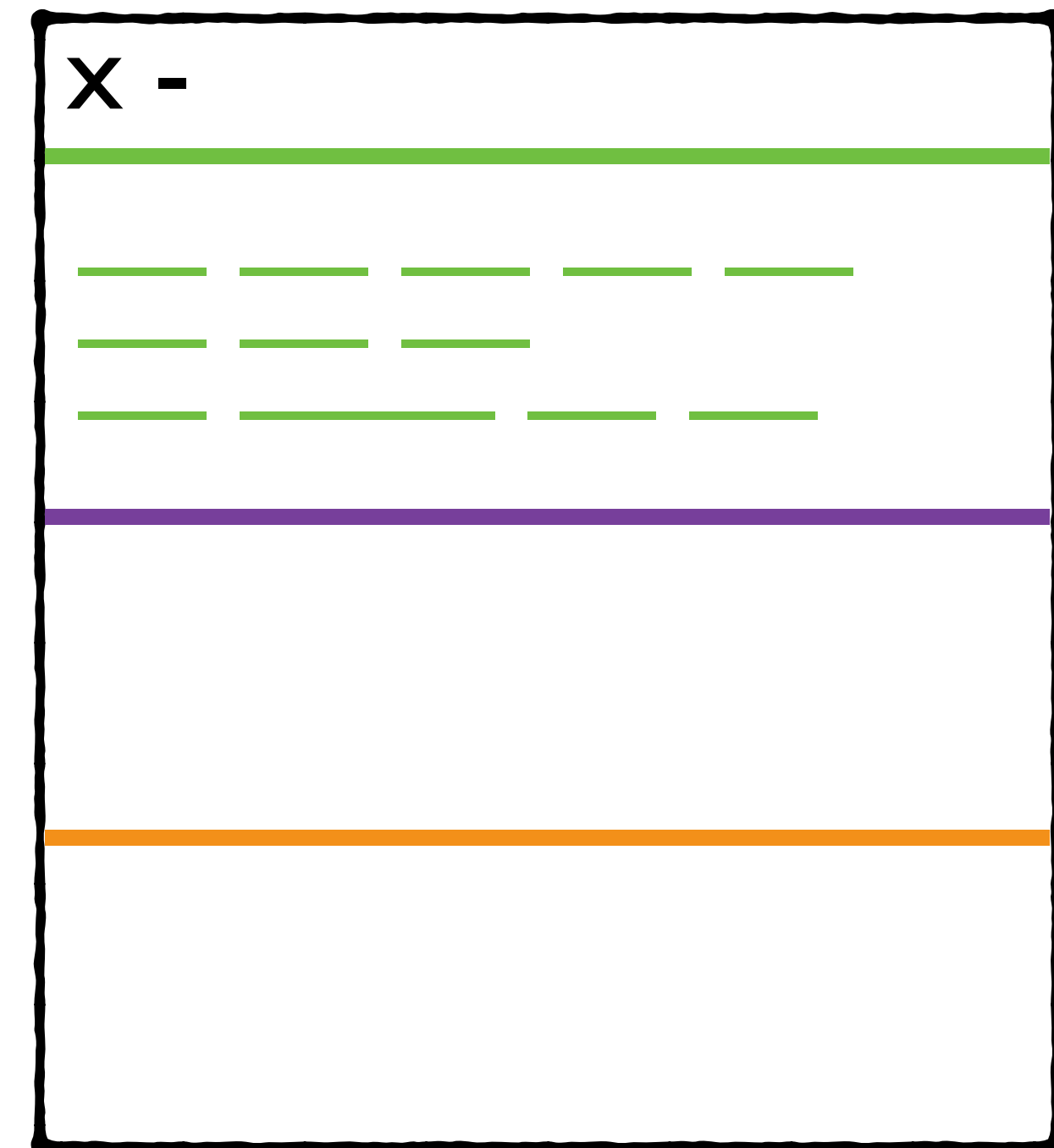
Уровни



Редактор

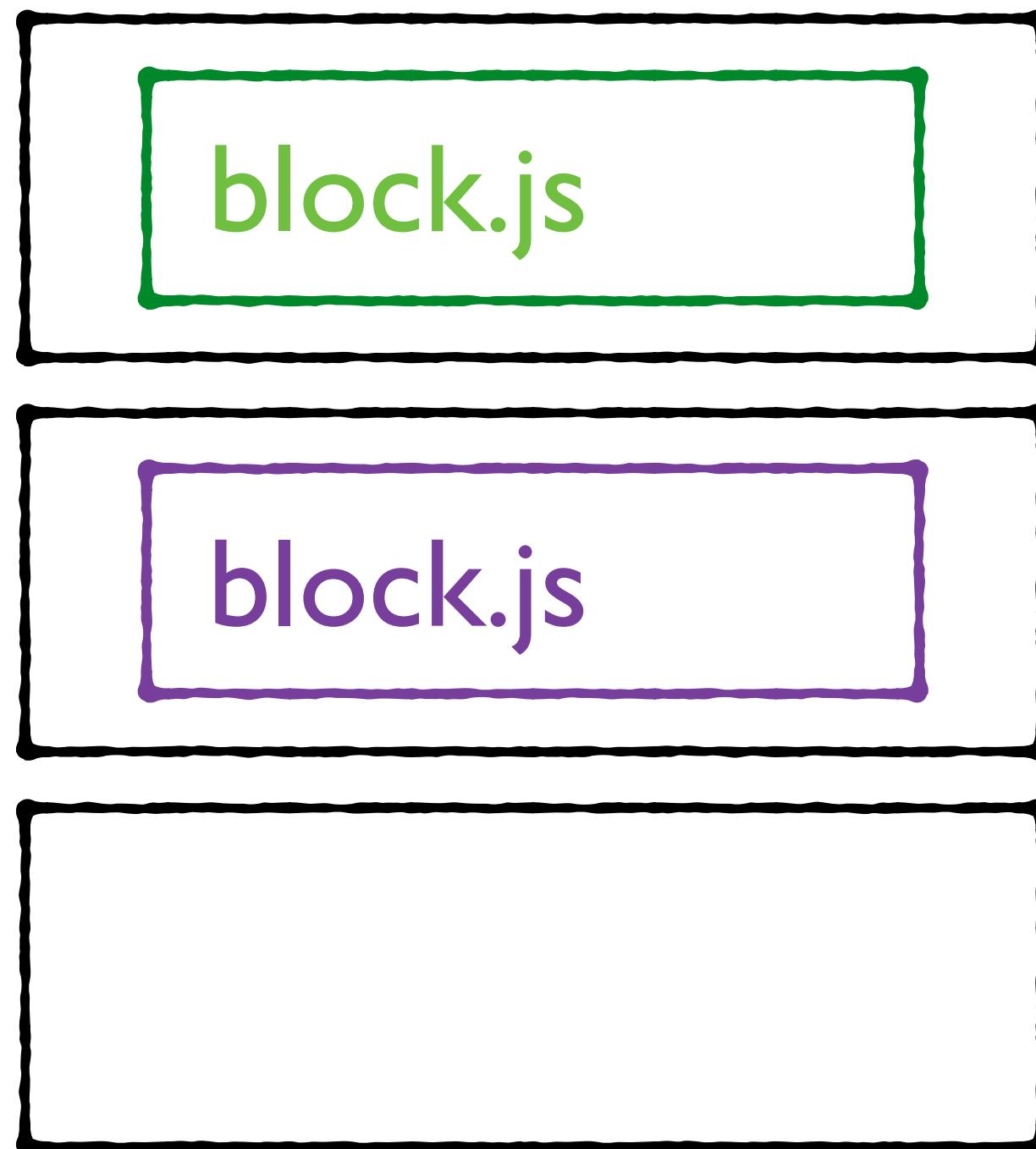


BEM IDE



Создавать блоки, а не файлы и папки

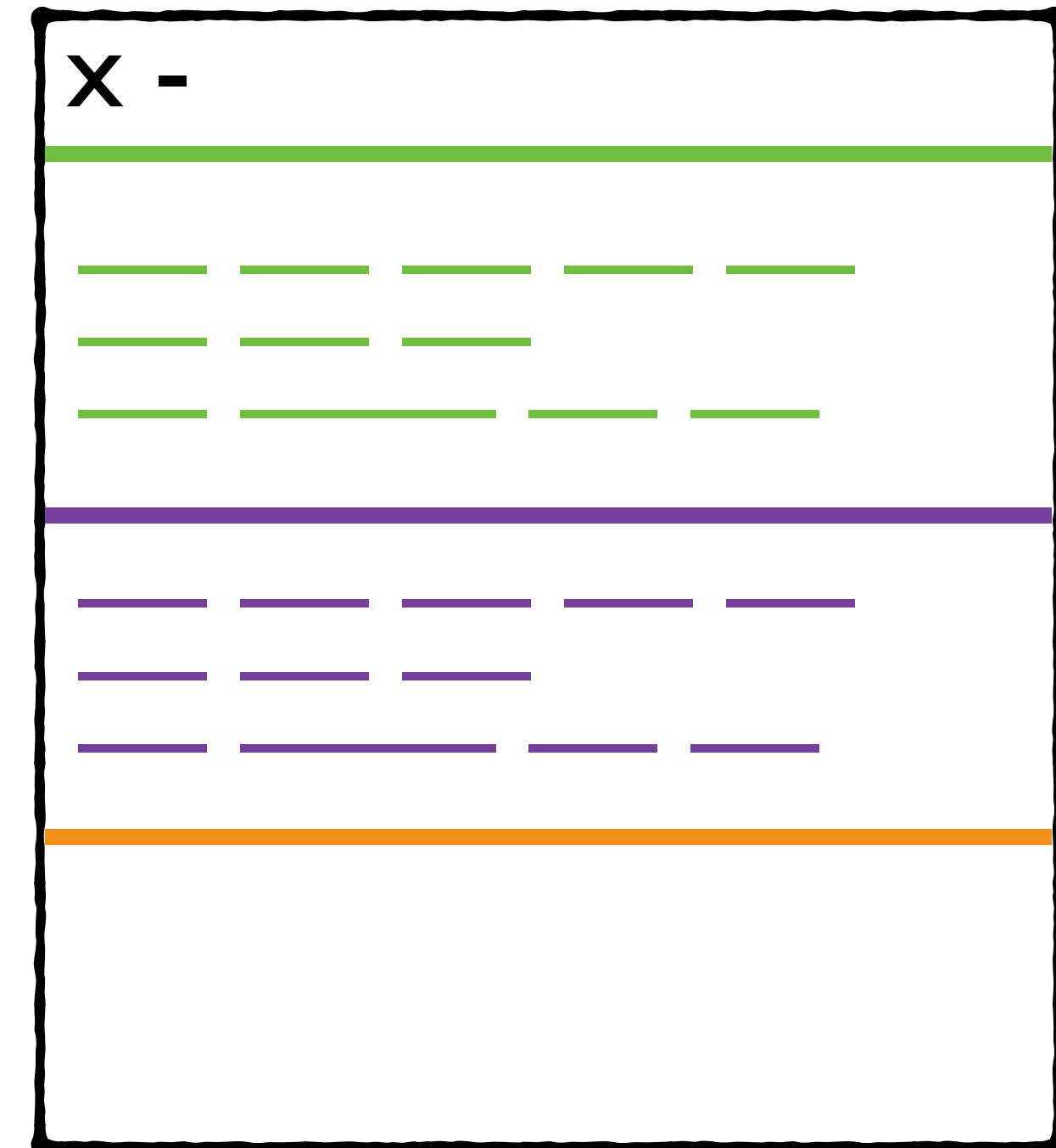
Уровни



Редактор

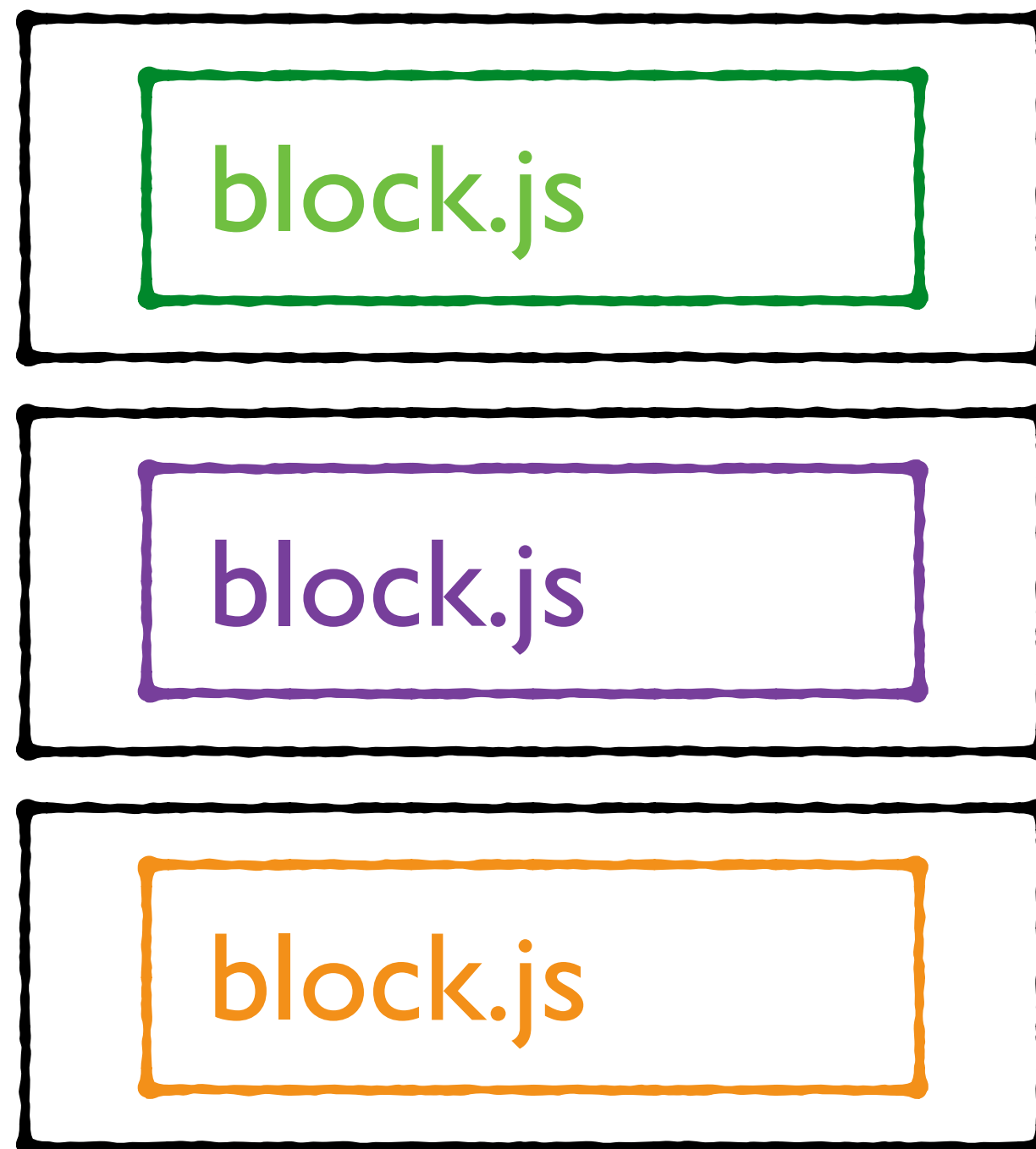


BEM IDE



Создавать блоки, а не файлы и папки

Уровни



Редактор



BEM IDE



Требования для работы с файловой системой

- › Работать с кодом блока со всех уровней
- › Создавать блоки, а не файлы и папки

ВЕМ IDE

Прототип редактора БЭМ проектов

ВЕМ IDE — что под капотом?

ВЕМ IDE — что под капотом?

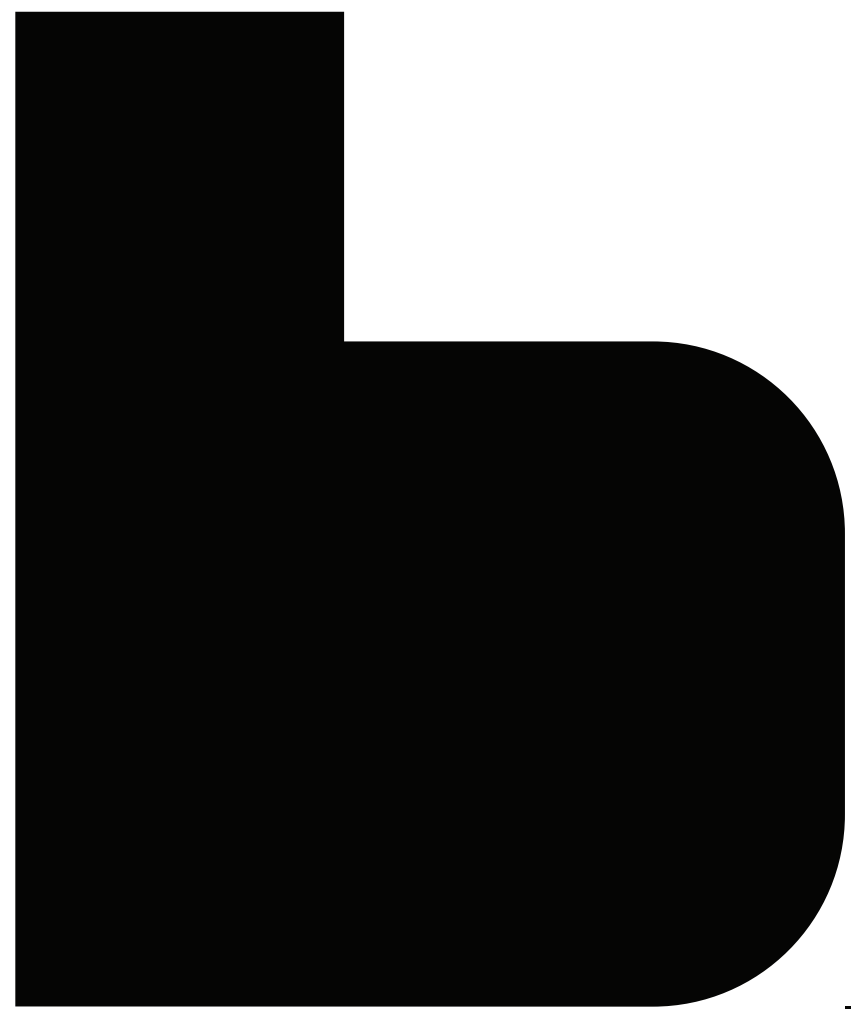
- › Node.js
- › Express
- › `bem-ide-toolkit`, библиотека для работы с БЭМ
- › Интерфейс на БЭМ
- › Библиотеки `bem-core` и `bem-components`

Где взять?

bit.ly/bem-ide

Итог и наши планы на будущее

Вопросы? :)



Спасибо!



bit.ly/spb-feedback

Саша Белянский

Разработчик интерфейсов в
группе разработки интерфейсов
рекламных технологий.

belyanskii@yandex-team.ru

 bem.info

 ru.bem.info/forum
[asktheteam](#)

 twitter.com/bem_ru

