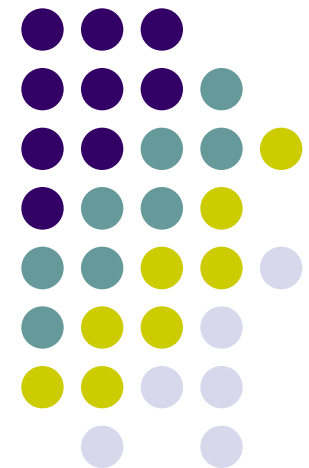
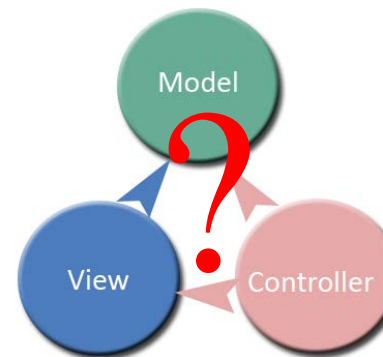


# Javascript ООП

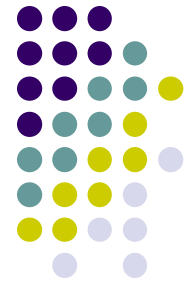
## Паттерны





# Данные в DOM

- ⑩ Храните данные в DOM
  - там, где это удобно
- ⑩ Склейка DOM <-> JS
  - MVVM
  - [Knockout.JS](#)
  - [Backbone Synapse](#)



# Behaviors

- ⑩ Делегирование + CSS + data-\*

*Sergey Miroshnyk*  
*smiroshnick@gmail.com*  
*0677223233*

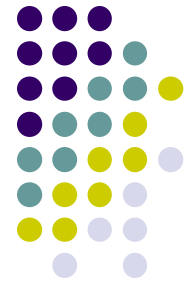


# Не плодить лишних объектов

- ⑩ Один объект-менеджер
  - Делегирование
  - Объект «по требованию»
  - Данные с сервера в разметке

*smiroshnick@gmail.com*

*0677223233*



# Flyweight (приспособленец)

## 10 Context-Menu (Flyweight)

*Sergey Miroshnyk*  
*smiroshnick@gmail.com*  
*0677223233*



# IoC

```
var Person = Class.extend({  
  inject: ["ajaxSubmitter"]  
  init: function(isDancing){  
    this.dancing = isDancing;  
    this.ajaxSubmitter = new AjaxSubmitter(args);  
  },  
  dance: function(){  
    return this.dancing;  
  }  
});
```

- 1) Decoupling for testing
- 2) Разные объекты (имена в конфиге), класс один



# Late binding

## 10 Раннее связывание

- что не так?
- как поправить?

1. Прототипное наследование

2. Позднее связывание



# Events & Plugins VS OOP

## 10 Расширение через events

- Плагины вместо наследования
- Минусы
  - Нельзя управлять вызовом родителя
  - Нельзя вернуть значение
    - Можно записать в объект события, но плагинов много...
  - Код управления зависимостями
  - Сложнее рефакторинг
  - Нужна документация, что можно делать и что нельзя
- Плюсы
  - Гибкость

## 10 Plugin architecture

- [Andy Osmani text video](#)
- [Nicholas Zakas — Scalable JavaScript Application Architecture](#)





# Функции → виджеты → данные

## ⑩ MV

- Фреймворки
  - Backbone.js
    - <http://peepcode.com/products/backbone-js>
  - [JavaScriptMVC](#)
  - Ext.JS 4
- Все изменения идут через модель
- Почитать
  - [MVC Architecture for JavaScript Applications](#)

## ⑩ + MVVM

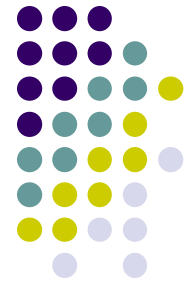
- Backbone Synapse



# Модуль / Синглтон

## 10 Объявление

```
var MODULE = (function () {  
    var my = {},  
        privateVar = 1  
    function privateMethod() {  
        // ...  
    }  
    my.moduleProperty = 1;  
    my.moduleMethod = function () {  
        // ...  
    }  
    return my  
})();
```



# Модуль

## ⑩ Расширение

```
var MODULE = (function (my) {  
    my.anotherMethod = function () {  
        // НОВЫЙ МЕТОД  
    };  
  
    return my;  
})(MODULE))
```

## ⑩ Создание/расширение

```
var MODULE = (function (my) {  
    // ДОБАВИТЬ ВОЗМОЖНОСТИ  
  
    return my;  
})(MODULE || {}))
```



# Материалы

- ⑩ [JavaScript-Module-Pattern-In-Depth](#)
- ⑩ Книги
  - Pro JavaScript Design Patterns
  - Object-Oriented Javascript (Stoyan Stefanov)
- ⑩ [<http://www.slideshare.net/nzakas/scalable-javascript-application-architecture>](#)