

# 构建富客户端时代的前端框架

---剖析 kissy 设计思想

承玉@taobao

# 富客户端时代到来



# Google 前端背后的

## ► Closure library

### Who uses Closure Library?

The Closure Library serves as the base JavaScript library for many Google products, including:



Gmail



Maps



Docs



Sites



Books



Reader



Blogger



Calendar



Photos

# 电子商务的复杂交互

您好, 欢迎来淘宝! 请登录 免费注册

我要买 我的淘宝 购物车 收藏夹 搜索 网站导航

淘宝旅行 trip.taobao.com

机票超前购 2人同行送100元现金 免费往返香港机票天天送

淘宝网 地球一小时

宝贝 淘宝商城 店铺 拍卖 淘吧 宝贝分享

Q 搜索 高级搜索 使用帮助

连衣裙 女包 2011新品 针织衫 冬装清仓 单鞋 T恤 鱼嘴鞋 更多

首页 淘宝商城 电器城 无名良品 疯抢 吃喝玩乐 HiTao嗨淘 淘花娱乐

特色中国 VIP专区 消费者保障

淘宝服务 更多

- 淘宝天下
- 彩票 NBA篮彩
- 淘宝旅行 春天花事
- 手机淘宝 一元拍
- 免费试用 抢机器人
- 阿里旺旺
- 淘江湖
- 聚划算 淘宝团购
- 游戏交易 淘女郎
- 壹佰壹店 淘画报
- 支付方式 淘公仔
- 淘宝母婴 时装周
- 品牌店铺 排行榜

7折抢购 Spring男人春季约会装 七天无理由 假一赔三

淘宝商城 名鞋馆 运动馆 家装馆 挑逗味蕾 办公室的那点零食

压轴T恤 闪耀每日风尚

精品家电 包邮6折起

设计不停 设计让我上瘾

全场包邮 速增好感度

Disney FAIRWHALE 骆驼 CAMEL 佰草集 HERBORIST VERO MODA SUPOR 苏泊尔

免费注册 登录 免费开店

公告 论坛 规则 安全保障 公益

- 国际信用卡交易解读!
- 315您的网购贴身宝典
- 支付盾免费送倒计时
- 我的保障, 我有话说

手机充值 游戏快充 淘宝旅行 保险

国内机票 国际机票 酒店 旅游

出发城市: H杭州 到达城市: 请选择 出发日期: 2011 年 3 月 21 日 查看折扣价

淘宝购物 快捷支付 立即参与

# Taobao 前端背后的

## ► KISSY



小巧灵活、简洁实用、愉悦编码、快乐开发

[源码](#) | [博客](#)

### 快速入门

KISSY是由淘宝前端攻城师们发起的一个开源项目。

KISSY提供了设计模式库、脚本库和最佳开发流程等工具，致力于减少重复开发，提高开发效率。

我们为你准备了一份快速入门教程，教你从最基本的地方开始[熟悉可爱的 KISSY](#) ^o^。

### API文档 & 范例

#### Seed

种子

- [kissy](#)
- [lang](#)
- [loader](#) [详细](#)

#### Core [作弊条](#)

核心文件

- [ua](#)
- [dom](#)
- [event](#)

#### Components [作弊条](#)

UI组件

- [cssreset](#)
- [cssgrids](#)
- [csscommon](#)
- [sizzle](#)
- [template](#)
- [datalazyload](#)
- [flash](#)
- [switchable](#)

### 编码最佳实践

[通用约定](#)

[HTML 编码规范](#)

[CSS 编码规范](#)

[JavaScript 语言规范](#)

[JavaScript 编码风格](#)

[KISSY 组件开发规范](#)

### 常用工具

[Closure Compiler](#): JS压缩

[YUI Compressor](#): CSS压缩

[JsDoc Toolkit](#): 文档生成

[Closure Linter](#): 检查校验

[Ant](#): 自动配置、打包

[iDebug](#): 页面调试

[CSSEmbed](#): 图片转DataURI

[ConvertZ](#): 繁简转换

[PNGOut](#): PNG优化

### 开发计划

[KISSY的新闻、更新、发布公告](#)

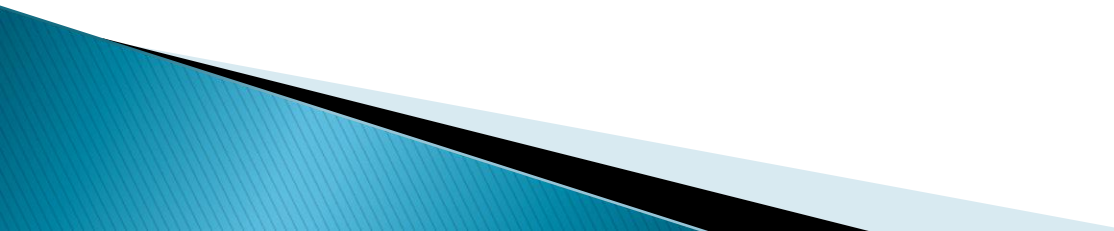
### 加入我们

旺旺群: 29676575

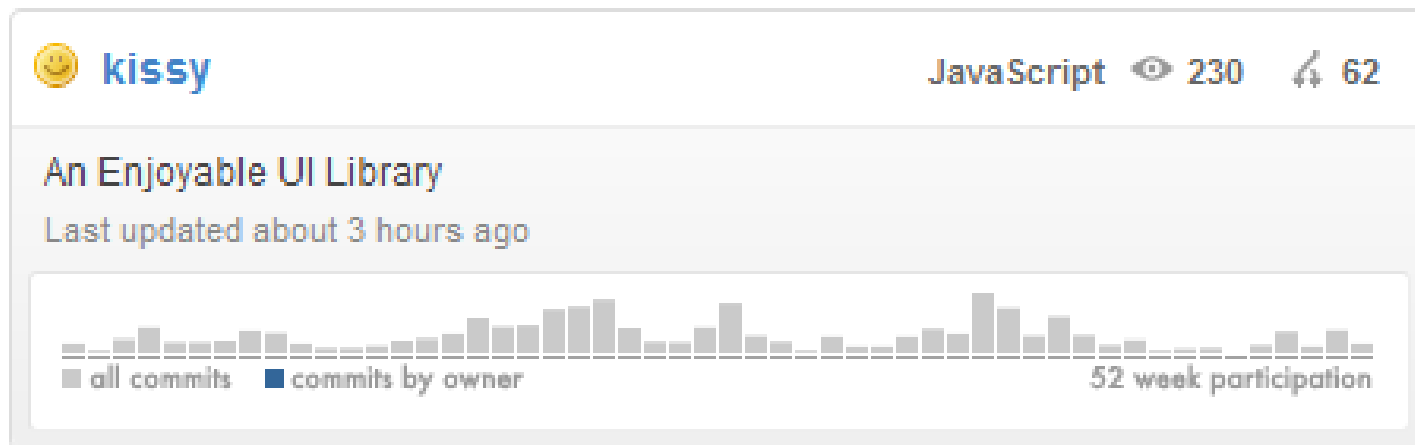
邮箱: [kissyteam@gmail.com](mailto:kissyteam@gmail.com)

如果你想贡献你的代码，你可以先了解[KISSY的开发流程](#)。

# 大纲

- ▶ Kissy 总体介绍
  - ▶ Kissy 核心
  - ▶ Kissy 模块化机制
  - ▶ Kissy 组件模型
  - ▶ Kissy 相关工具与第三方框架
  - ▶ Kissy 展望
- 

# Kissy 总体介绍



► Kissy on github



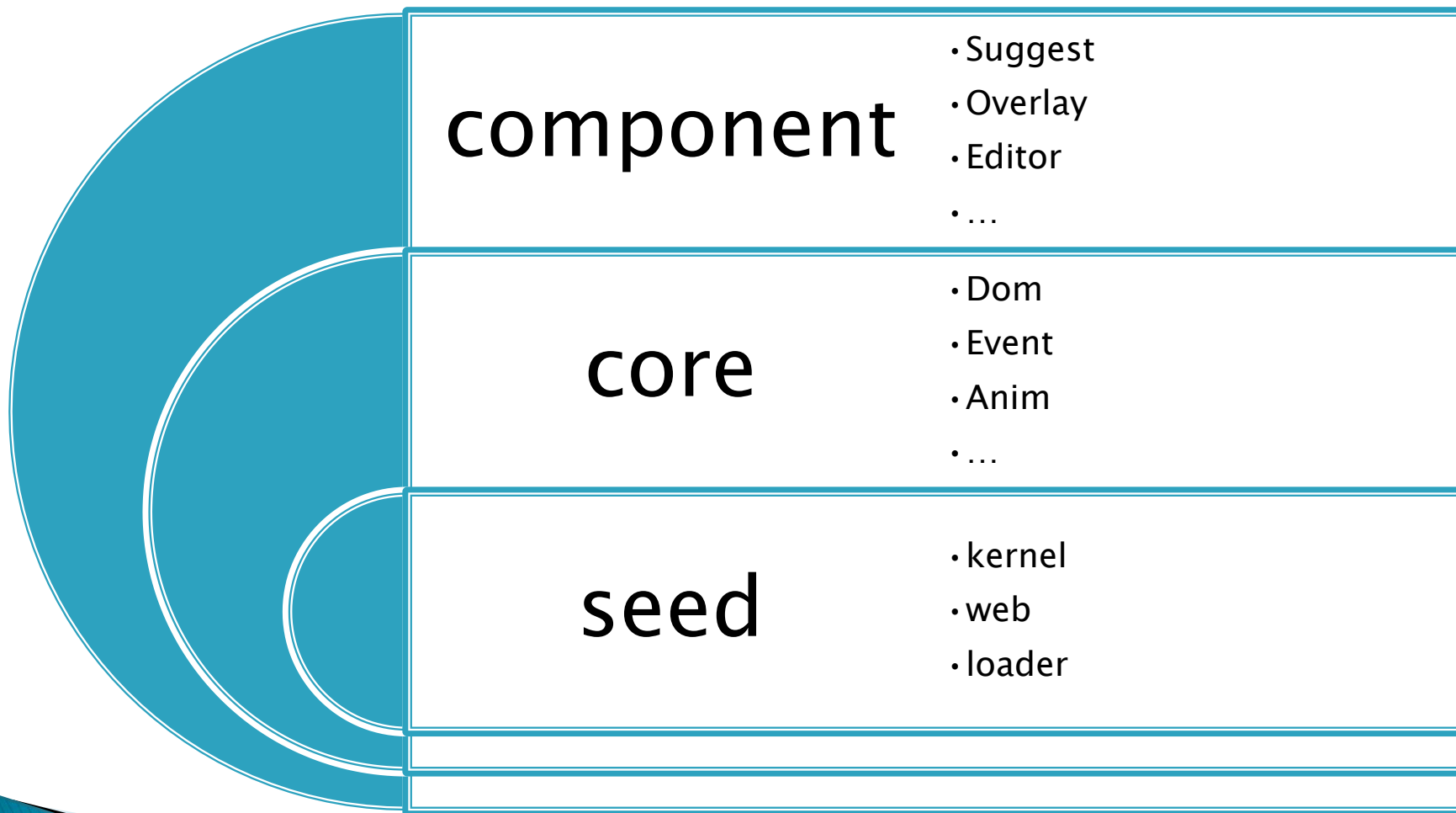
# 名字来源



小巧灵活，简洁实用

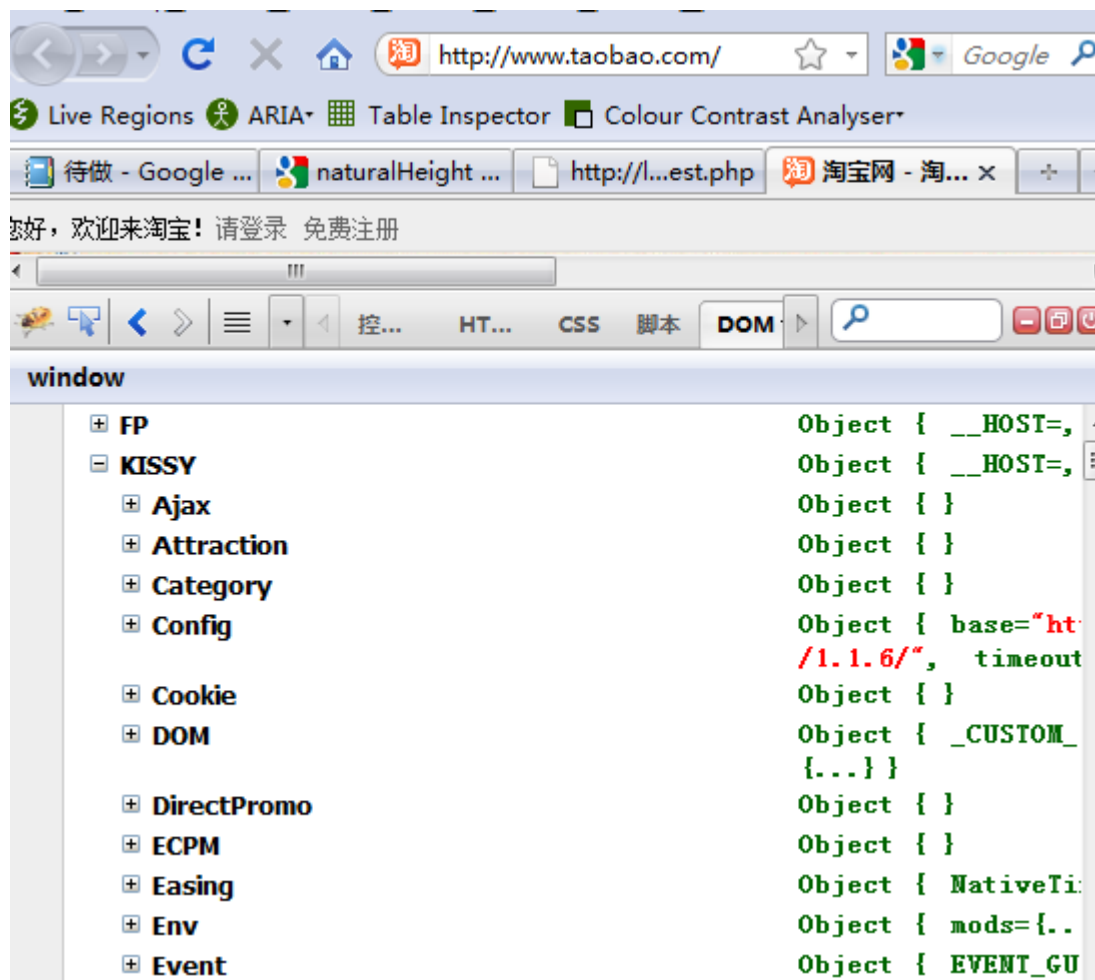


# 总体架构



# 淘宝中的应用

## ► 核心



# 淘宝中的应用

- ▶ 组件:tab,slider,overlay,imagezoom ...



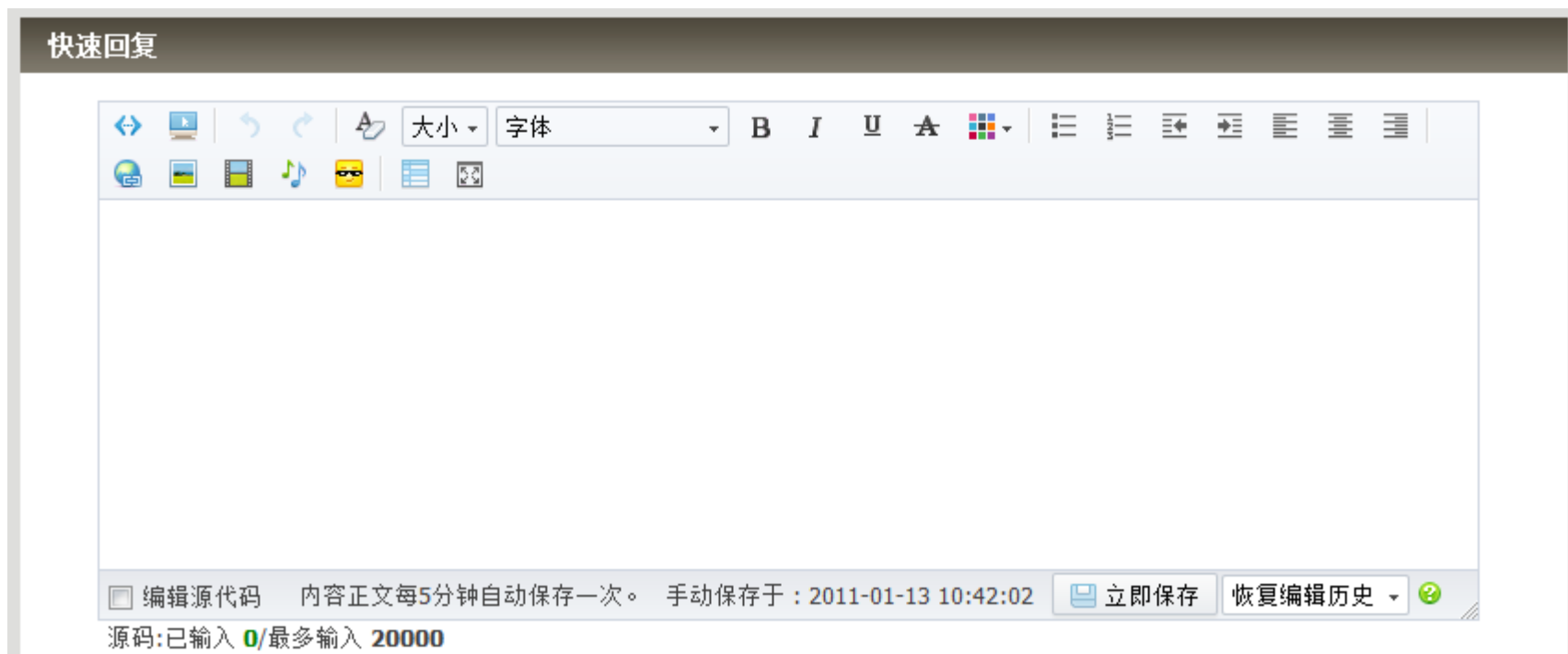
# 淘宝中的应用

## ▶ 自动补全 - suggest



# 淘宝中的应用

## ► Kissy editor : 编辑器



# Kissy 核心

Core

{dom/event/anim/ajax...}

Web.js

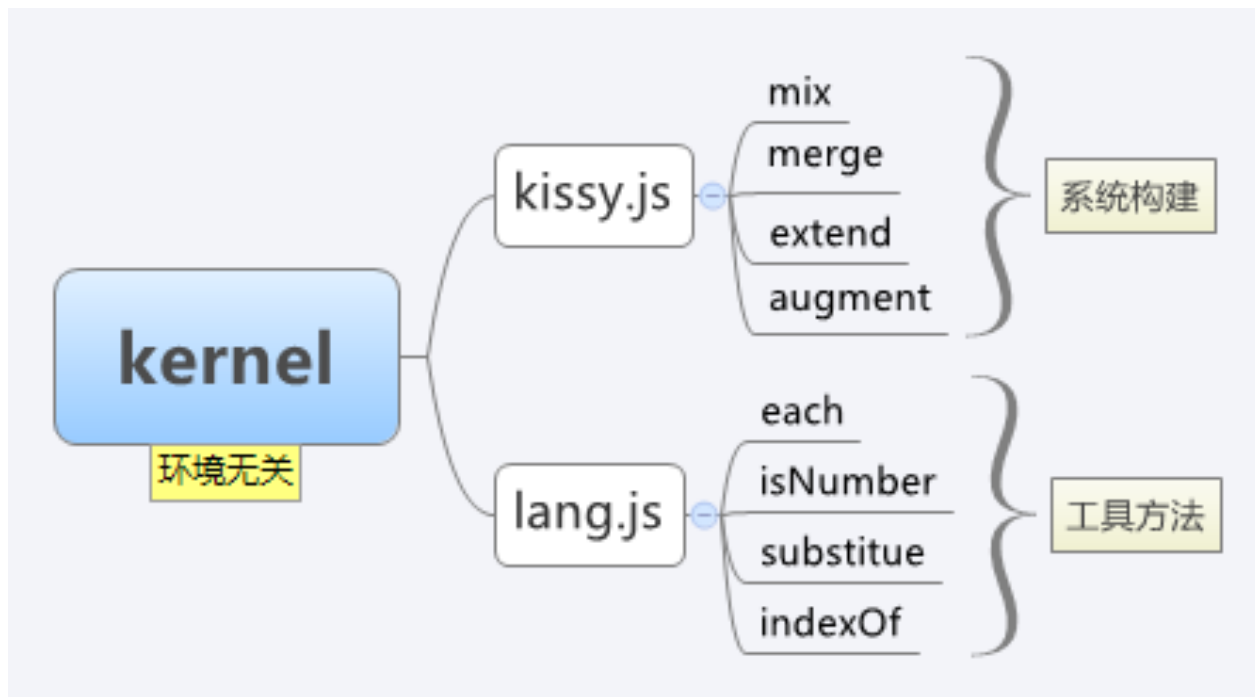
kissy.js

Lang.js

Loader.js

seed

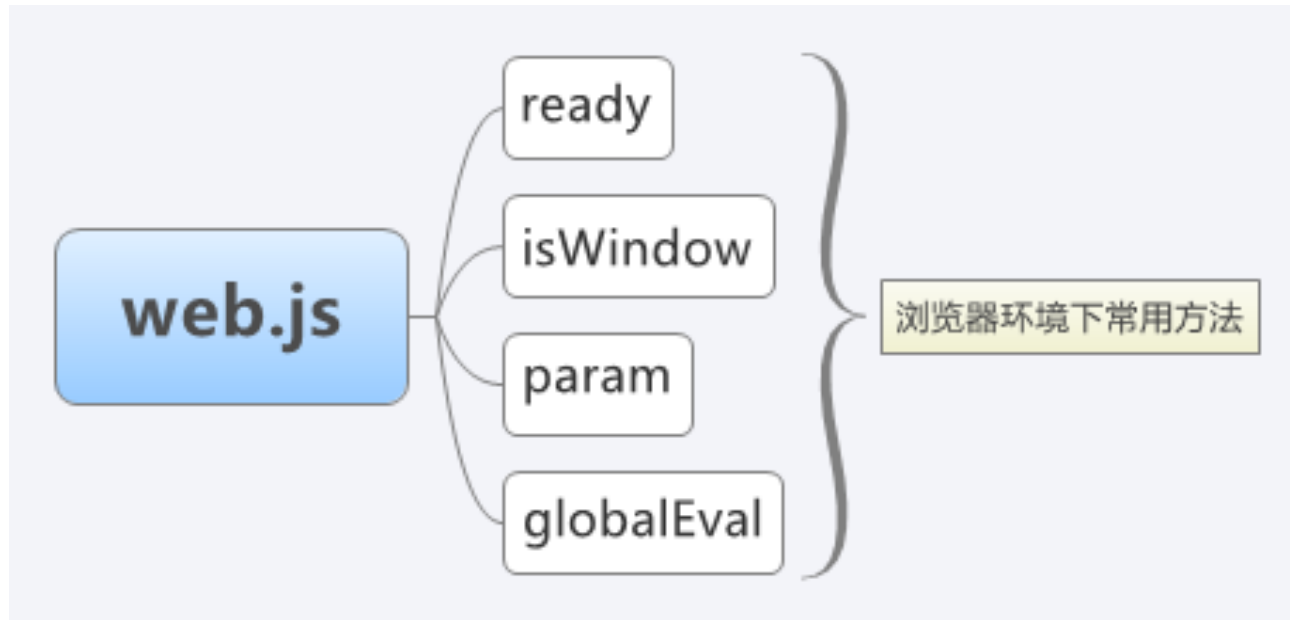
# kernel



独立于环境，可运行于任意宿主环境  
客户端 Browser  
服务器端 Nodejs，命令行自动化测试

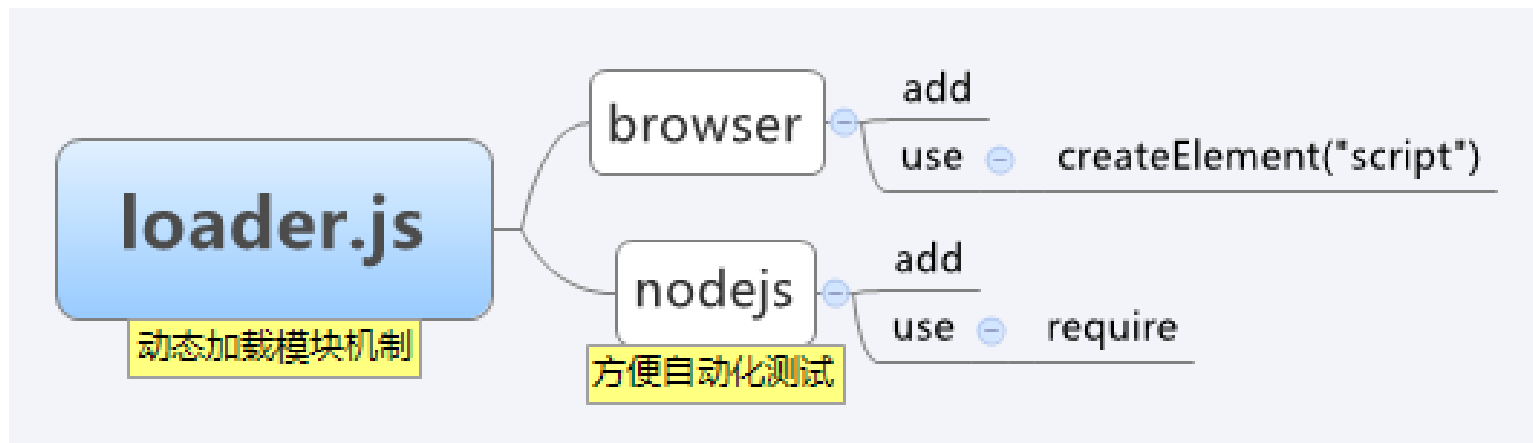


# Web.js



客户端浏览器环境常用方法

# Loader.js



浏览器环境与nodejs环境

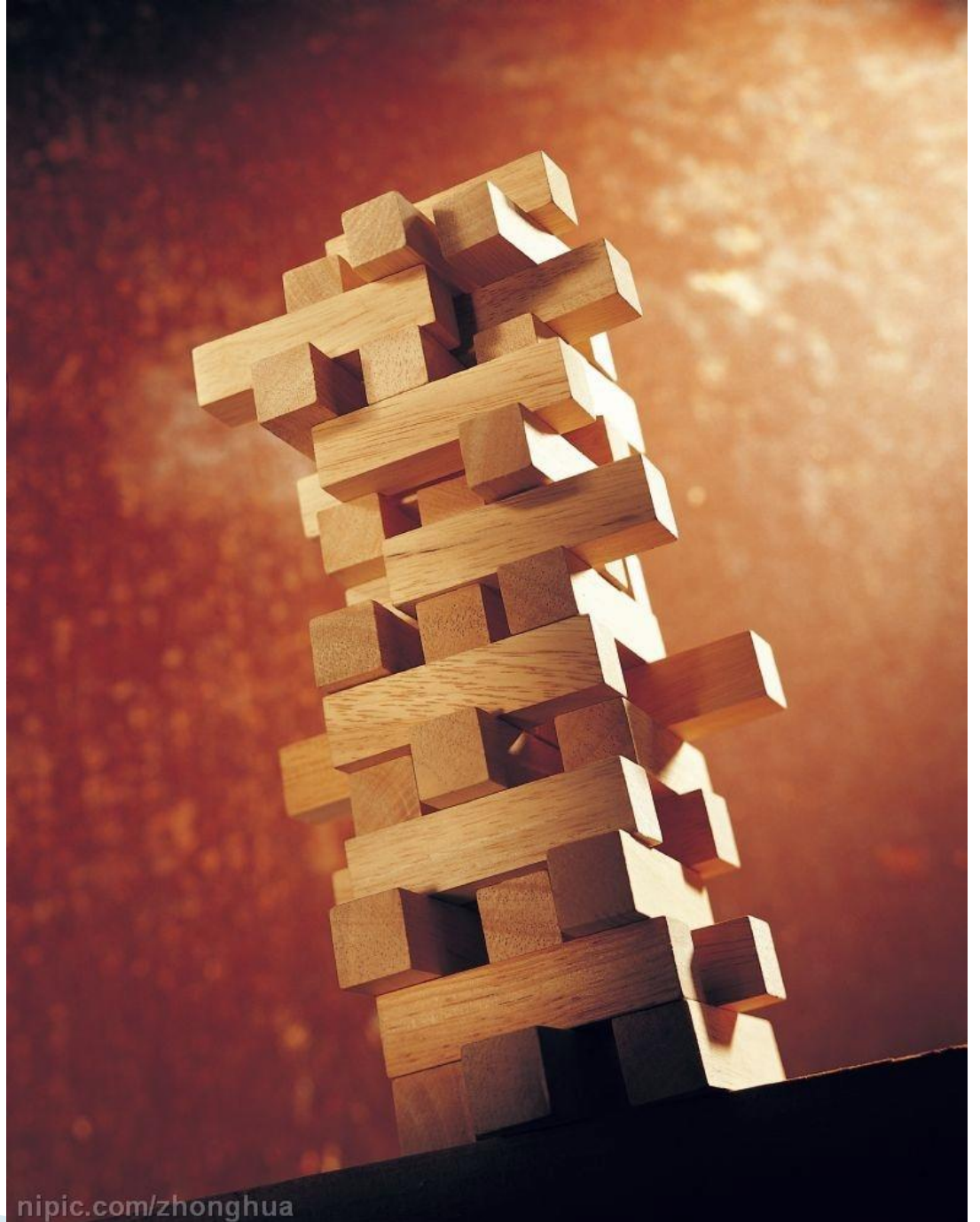
add : 模块定义

use : 动态加载模块与使用

# Kissy 模块化机制

- ▶ Taobao 首页示例
  - ▶ 模块化思想
  - ▶ 前端代码的演变
  - ▶ Kissy 模块化
- 

# 模块



# Taobao 首页模块划分



# 首页 js 文件

- ▶ 每个模块控制自身区域的交互与业务逻辑

<http://a.tbcdn.cn/??>

[s/kissy/1.1.6/switchable/switchable-pkg-min.js,](#)

[s/kissy/1.1.6/suggest/suggest-pkg-min.js,](#)

[s/kissy/1.1.6/datalazyload/datalazyload-pkg-min.js,](#)

[s/kissy/1.1.3/flash/flash-pkg-min.js,](#)

[p/fp/2011a/expressway/profile-min.js,](#)

[p/fp/2011a/header/header-min.js,](#)

[p/fp/2011a/attraction/attraction-min.js,](#)

[p/fp/2011a/expressway/expressway-min.js,](#)

[p/fp/2011a/category/category-min.js,](#)

[p/fp/2011a/category/sub-promotion-min.js,](#)

[p/fp/2011a/guide/alimama-ecpm-min.js,](#)

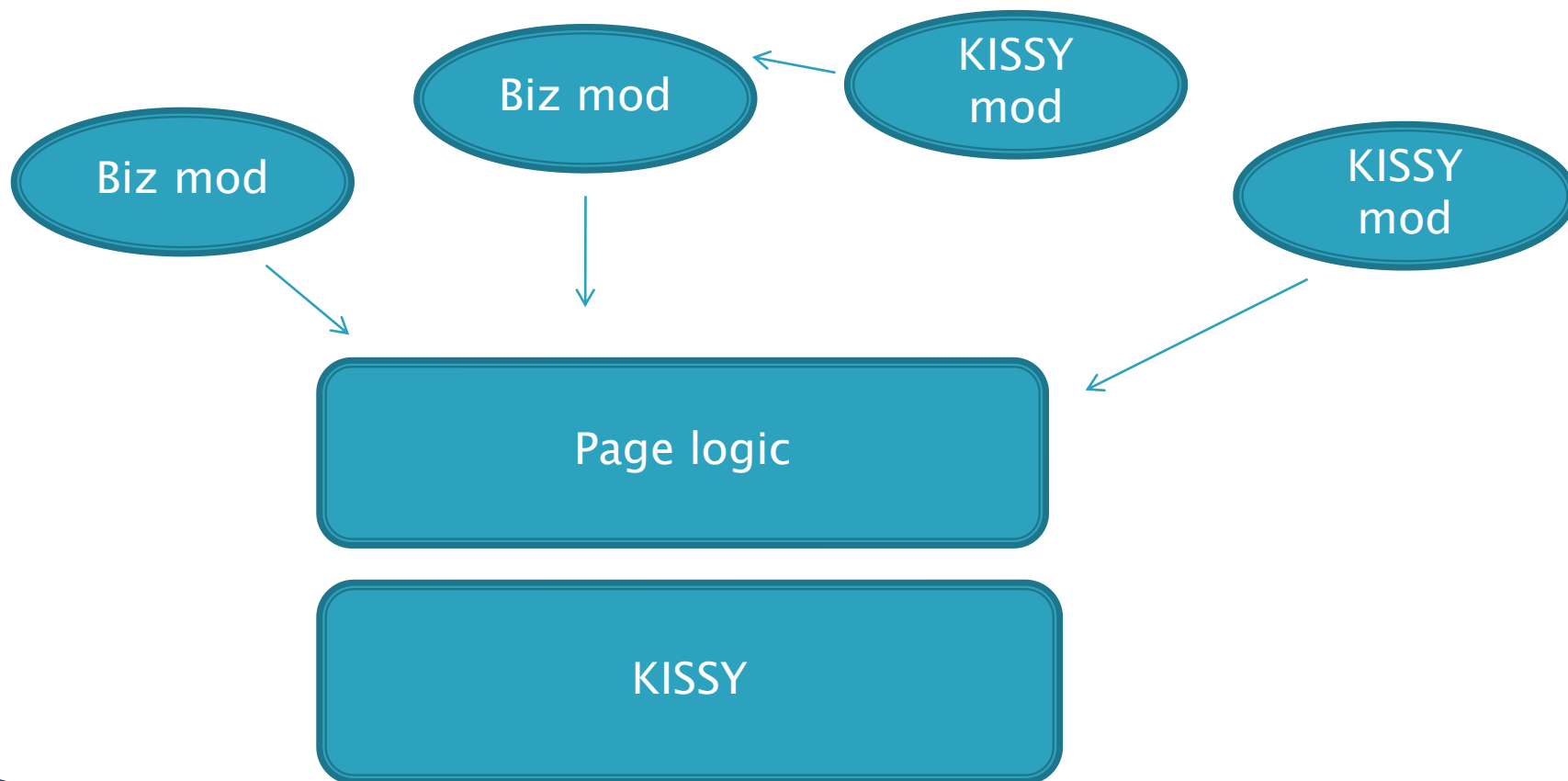
[p/fp/2010c/js/fp-hubble-monitor-min.js,](#)

[p/fp/2011a/hotsale/p4p-min.js,](#)

[p/fp/2011a/footer/footer-min.js?t=20110322.js](#)



# 前端页面构建





# 模块化设计

- ▶ 一种将系统分离成独立功能部分的方法
  - 函数分割成独立功能组
  - 使用面向对象来描述模块接口
  - 通过使用工业标准达到系统的透明可扩展
  - 缺点：由于模块间的消息通信导致性能受损
  - 优点：可维护性，焦点分离

# 语言层面支持

- ▶ Pascal
- ▶ Erlang
- ▶ Perl
- ▶ Python
- ▶ Ruby
- ▶ php
  
- ▶ Java
- ▶ C/C++
  
- ▶ 同步加载

# php

- ▶ A.php 模块定义以及实现

- ▶ -----main.php

```
require_once ("A.php") ;
```

```
//use A
```

- ▶ 使用 require\_once 载入，防止重复
- ▶ 配置 include\_path 定义查找路径
- ▶ 没有离线编译过程，引擎动态优化

# java/python

- ▶ A.java/A.py 模块定义实现

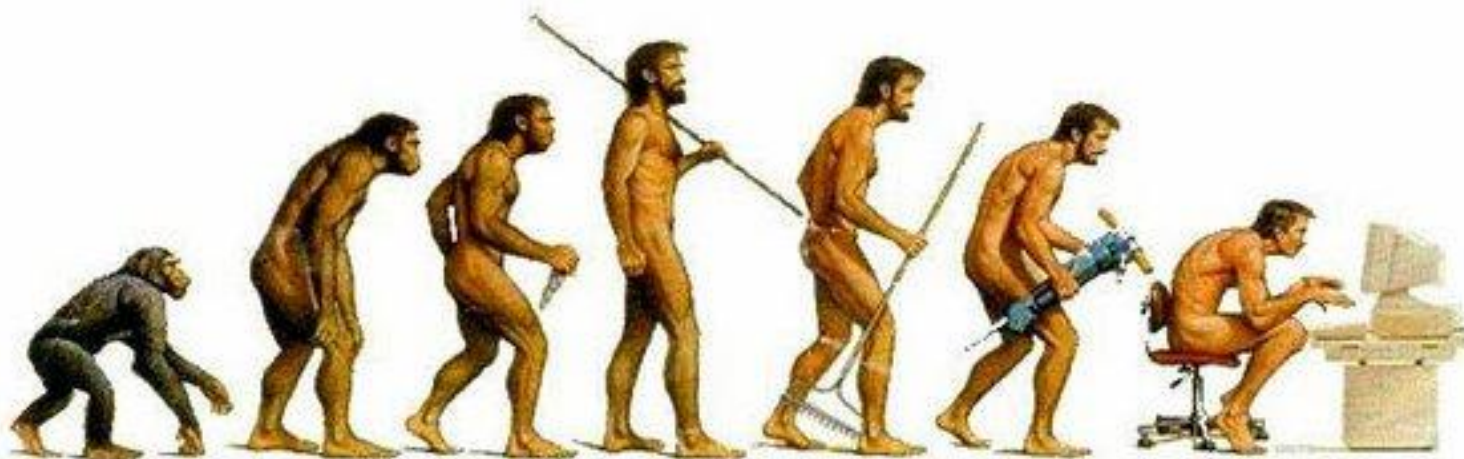
- ▶ -----main.java/main.py

```
import A;
```

```
//use A
```

- ▶ import 引用其他模块
- ▶ classpath/sys.path 定义模块路径，自动补全后缀名查找对应模块文件
- ▶ Java 只用类来实现模块编程，没有合并过程（jar?），只有压缩过程（编译为紧凑的class/pyc）

# 前端代码所经历的....



# 90年代

```
<body onload="init()">  
  <a href="#"  
    onclick=""  
    color="red">  
    Click  
  </a>  
</body>
```

# 2000-2005

- ▶ 脚本，标签，样式分离

```
<body>
```

```
  <a class='go' id='act'>
```

```
    click me
```

```
  </a>
```

```
  <script>
```

```
    window.onload=function(){};
```

```
    var a=document.getElementById('go');
```

```
    a.onclick=function(){};
```

```
  </script>
```

```
</body>
```





# 2005-2009

- ▶ 外部脚本，全局勿扰，闭包

- ▶ -----y.html

```
<body>
```

```
  <a class='go'>click<a>
```

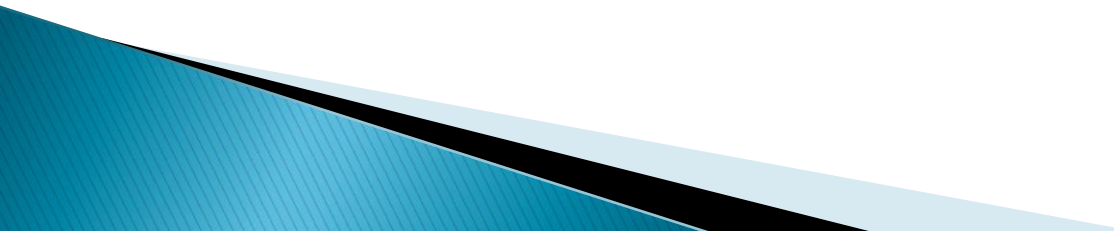
```
  <script src='xx.js'></script>
```

```
</body>
```

- ▶ -----xx.js

```
(function(){})( );
```

# 2010 – 2011

- ▶ 动态并行加载，前端模块依赖，自动压缩打包
  - ▶ Labjs : script loader
  - ▶ Yui3 : module loader
  - ▶ Closure library : module program
- 

# labjs

- ▶ 不需要静态引入外部脚本

**<script>**

```
$LABjs
```

```
.script("a.js")
```

```
.script("b.js")
```

```
.wait()
```

```
.script("c.js");
```

**</script>**

# YUI3

- ▶ 利用 function wrapper 分离 load , execute ,引入模块概念

- ▶ 模块定义:

```
YUI.add("module1",function(S){  
    S.Module1=function(){};  
});
```

- ▶ 模块注册

```
YUI.add({module1:{  
    Fullpath:"xxx.js"  
}});
```

- ▶ 模块使用

```
YUI().use("module1",function(S){  
    //use S.Module1  
});
```

# 未来

▶ Commonjs/module 规范?

▶ `<script  
 src='module.js'  
 data-main='init'>  
</script>`

# Kissy 模块化

## ▶ 原则

- 内聚性
- 松耦合
- 零重复
- 封装
- 可测试性
- 可读性
- 单一职责

## ▶ 模块解耦与焦点分离

# 1.模块也有值

```
KISSY.add("dom", function() {  
    KISSY.DOM = { addClass:function() {} };  
});
```

## ▶ 直接返回值

```
KISSY.add("dom", function(KISSY) {  
    return { addClass:function() {} };  
});
```



## 2.模块名的命名空间

```
KISSY.add("event-target", function() {  
    KISSY.Event.Target = function(){};  
});
```

### ▶ 命名空间精简：

```
KISSY.add("event/target", function() {  
    var EventTarget = function(){}  
    return EventTarget;  
});
```

# 3.去除模块名

- ▶ 根据文件系统确定模块名字

- ▶ event / base.js :

```
KISSY.add(function() {});
```

- ▶ 就表示模块 event/base

## 4.包配置

- ▶ 包与路径约定，批量注册模块集合

```
S.config({  
    packages: [{  
        name: 'm1',  
        path: "http://xx.com/"  
    }]  
});  
S.use("m1/base") -> http://xx.com/m1/base.js
```

# 5. 依赖注入

## ▶ 定义模块

```
KISSY.add("xx", function(S, DOM) {  
    , {  
        requires: ["dom"]  
    });
```

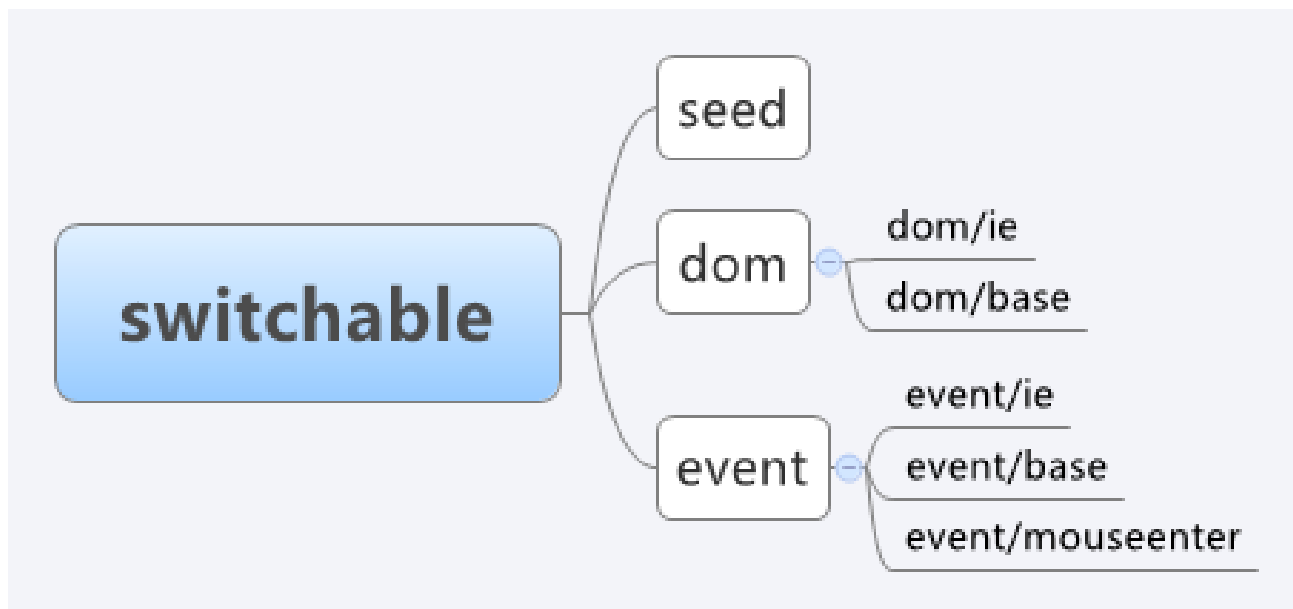
# 5. 依赖注入

## ▶ 使用模块:

```
S.use("dom", function (S, DOM) {  
    //use DOM  
});
```

## 6. 模块依赖打包工具

- ▶ Module compiler : 根据应用打包最紧凑的框架库



# Kissy 组件模型



- ▶ 组件 == 模块集合
- ▶ 淘宝组件开发所经历的阶段介绍
- ▶ Suggest->Switchable->overlay->menu

# 阶段1

## ► 代表: Suggest





# 代码采样

▶ //默认配置

```
var defaultConfig={};
```

```
function Suggest(config){
```

```
  S.mix(config,defaultConfig,false);
```

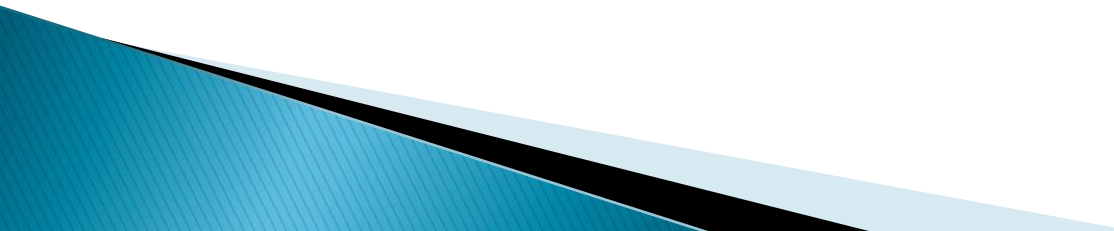
```
  //初始化
```

```
}
```

```
//功能
```

```
S.augment(Suggest,EventTarget,{});
```

# 阶段1 特点

- ▶ 单类，单文件，单模块
  - ▶ 逻辑（数据，控制，渲染）集中
  - ▶ 维护性不佳
- 

# 阶段2

## ▶ 代表：Switchable



给力秒杀 家装大牌联合巨献



公告

论坛

规则

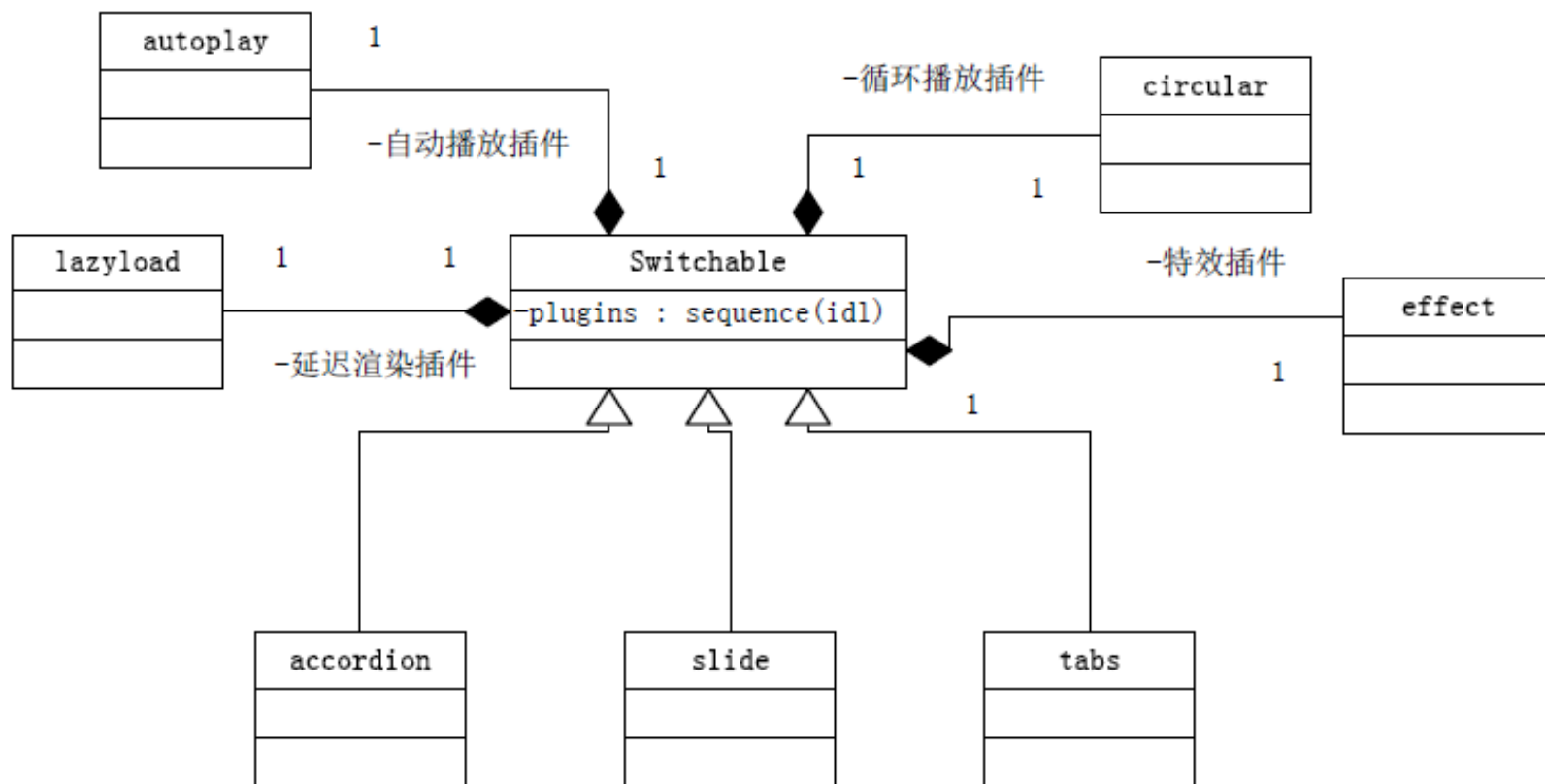
安全保障

公益

- 政协委员建言扶植电商
- 高位瘫痪 开网店创业
- 卖家主动提醒买家维权
- 淘宝购物你有信心吗?



# 阶段2结构图

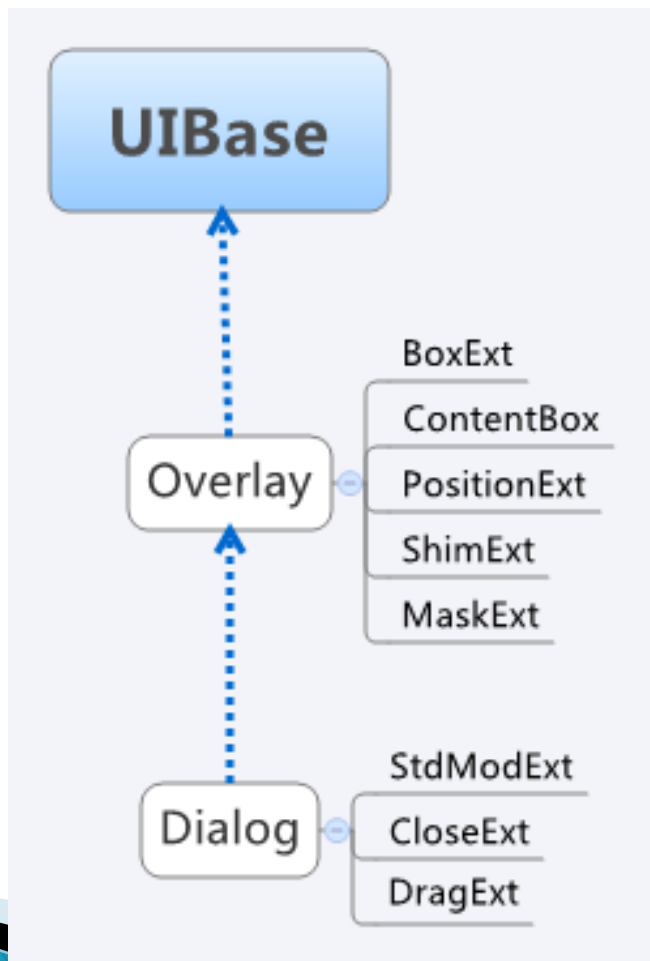


# 阶段2特点

- ▶ 总体逻辑分离为几个模块文件.
- ▶ 单个模块逻辑功能（数据，控制，渲染）集中.
- ▶ 分离机制不能被其他组件公用，仅适合自身.

# 阶段3

- 代表：Overlay using UIBase



# 代码采样

```
var Component=UIBase.create(Parent,['MixComponent'],{
  initializer:function(){},
  _uiSetX:function(){},
  renderUI:function(){},
  bindUI:function(),
  destructor:function(){},
},{
  ATTRS:{
    x:{},
  },
  HTML_PARSER:{
    yEl:".cl"
  }
});
```

属性x变化处理逻辑  
渲染周期，生成html  
绑定事件，控制逻辑  
销毁逻辑

声明属性特征

从html中获取组件属性

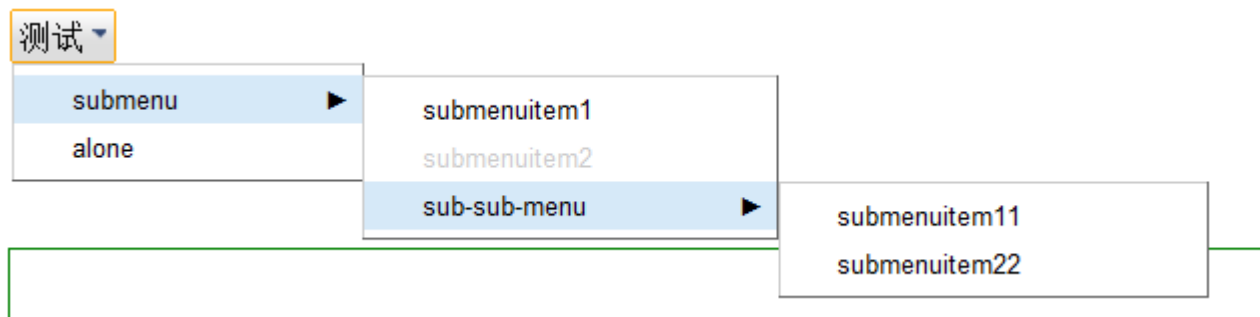
# 阶段3特点

- ▶ 完善的属性管理机制
  - ▶ 生命周期管理
  - ▶ mixin 支持的基础框架
- 
- ▶ Refer: <http://goo.gl/sgvXP>

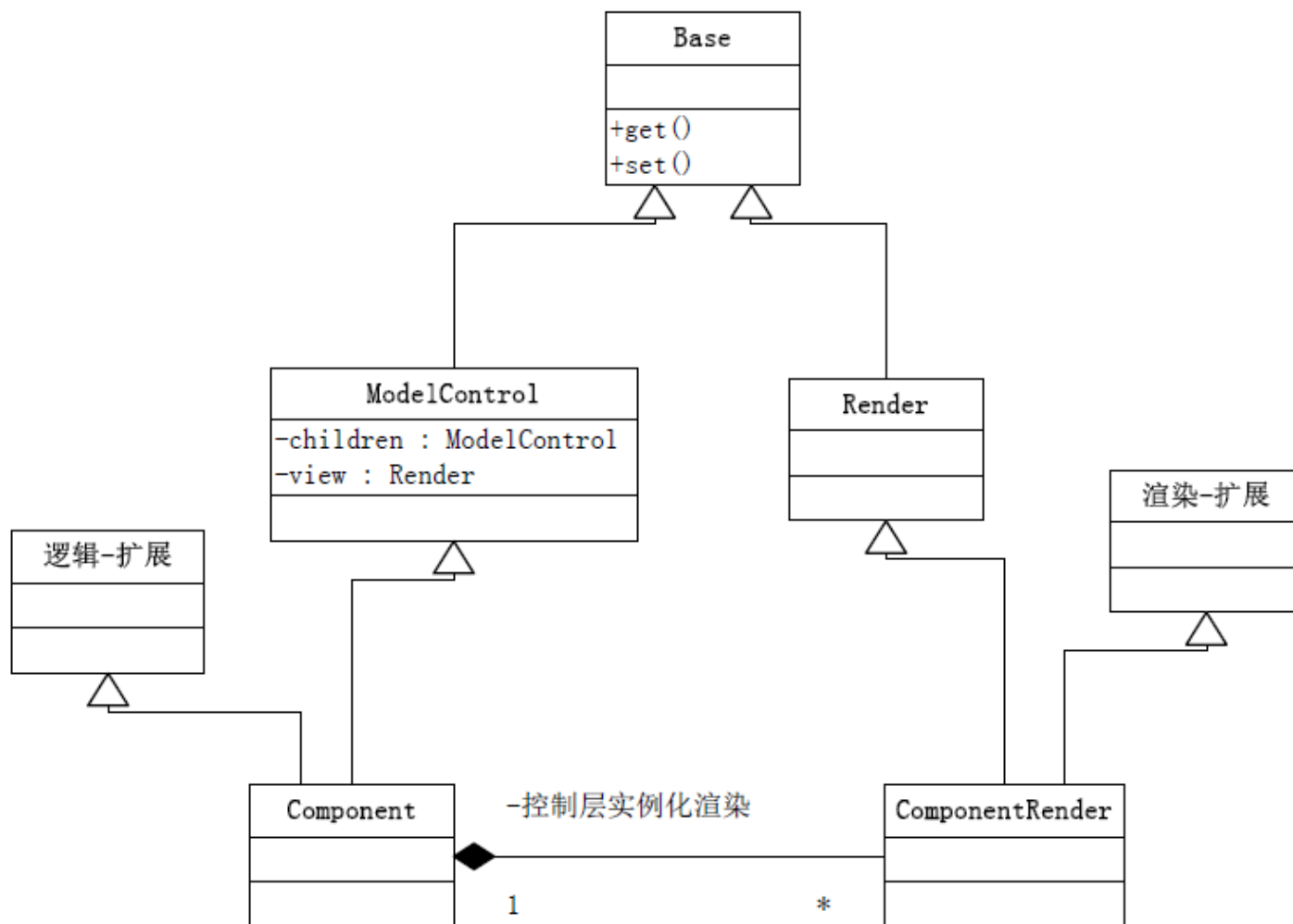


# 阶段4

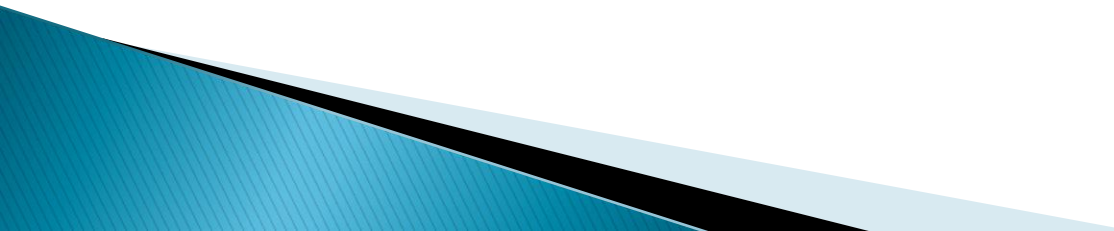
- ▶ 代表: menu,menubutton
- ▶ Component , MVC 分离



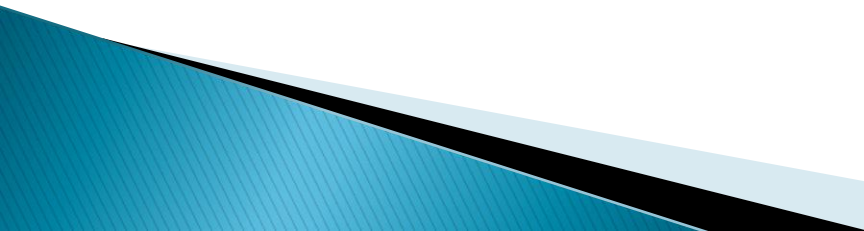
# 层次结构



# 阶段4特点

- ▶ 基于阶段3发展而来
  - ▶ 根据 MVC 逻辑模块分离
  - ▶ Render : 渲染逻辑
  - ▶ ModelControl: 数据与控制逻辑
- 

# KISSY 相关工具与第三方框架

- ▶ 单元测试框架
  - ▶ Build 构建工具
  - ▶ 文档生成工具
- 

# 单元测试框架

- ▶ Behavior-driven development framework



- ▶ 

```
describe("Jasmine", function() {  
  it("makes testing JavaScript awesome!",  
    function() {  
      expect(yourCode).toBeLotsBetter();  
    });  
});
```

# Kissy on jasmine

## KISSY 集成化测试

#test2

#test6

### KISSY.Anim Test Cases

[Jasmine](#) 1.0.1 revision 1286311016

Show ☐ passed ☐ skipped

Running...

run all

[anim](#)

run

### KISSY Seed Test Cases

[Jasmine](#) 1.0.1 revision 1286311016

Show ☒ passed ☐ skipped

43 specs, 0 failures in 0.487s Finished at Thu Mar 24 2011 13:23:20 GMT+0800 (China Standard Time)

run all

[kissy.js](#)

run

[S.mix](#)

run

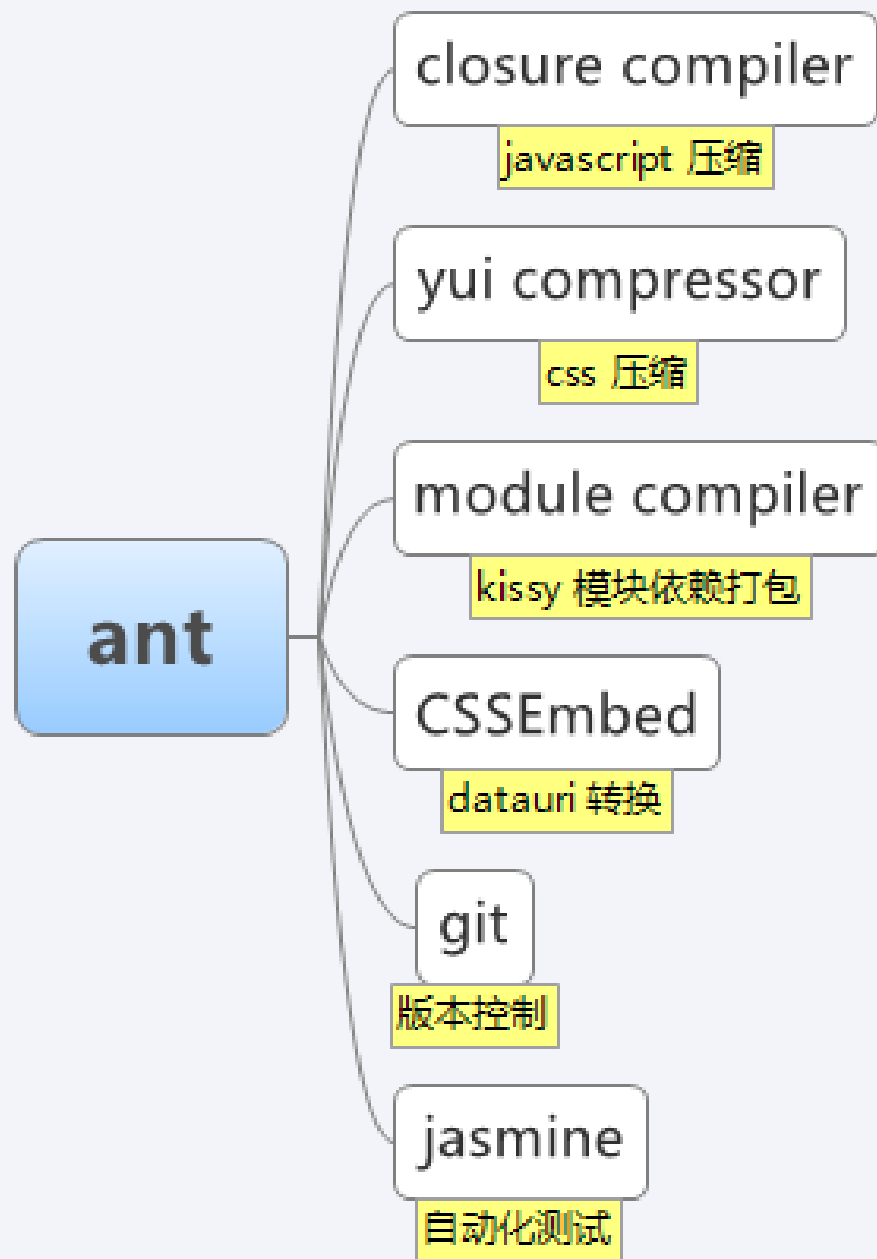
[S.merge](#)

run

[S.augment](#)

run

# 构建工具



# 文档工具

- ▶ Sphinx : 半自动化文档工具



- ▶ Cheers for a great tool that actually makes programmers want to write documentation!



# 最终文档



[Home](#) / [Blog](#) / [About](#)

## KISSY Loader 文档

by 承玉 <yiminghe@gmail.com>

### 内容

- [API](#)
  - [引入](#)
  - [添加模块](#)
    - [函数接口](#)
    - [KISSY 1.2 新增接口](#)
  - [使用模块](#)
    - [函数接口](#)
    - [kissy 1.2 新增接口](#)
- [使用示例](#)
  - [KISSY 1.2 前](#)
    - [注册模块](#)
    - [定义模块](#)
    - [使用模块](#)
    - [demo](#)
  - [KISSY 1.2](#)
    - [包配置](#)

# Kissy 展望



Kissy 核心

Components  
电子商务组件  
库

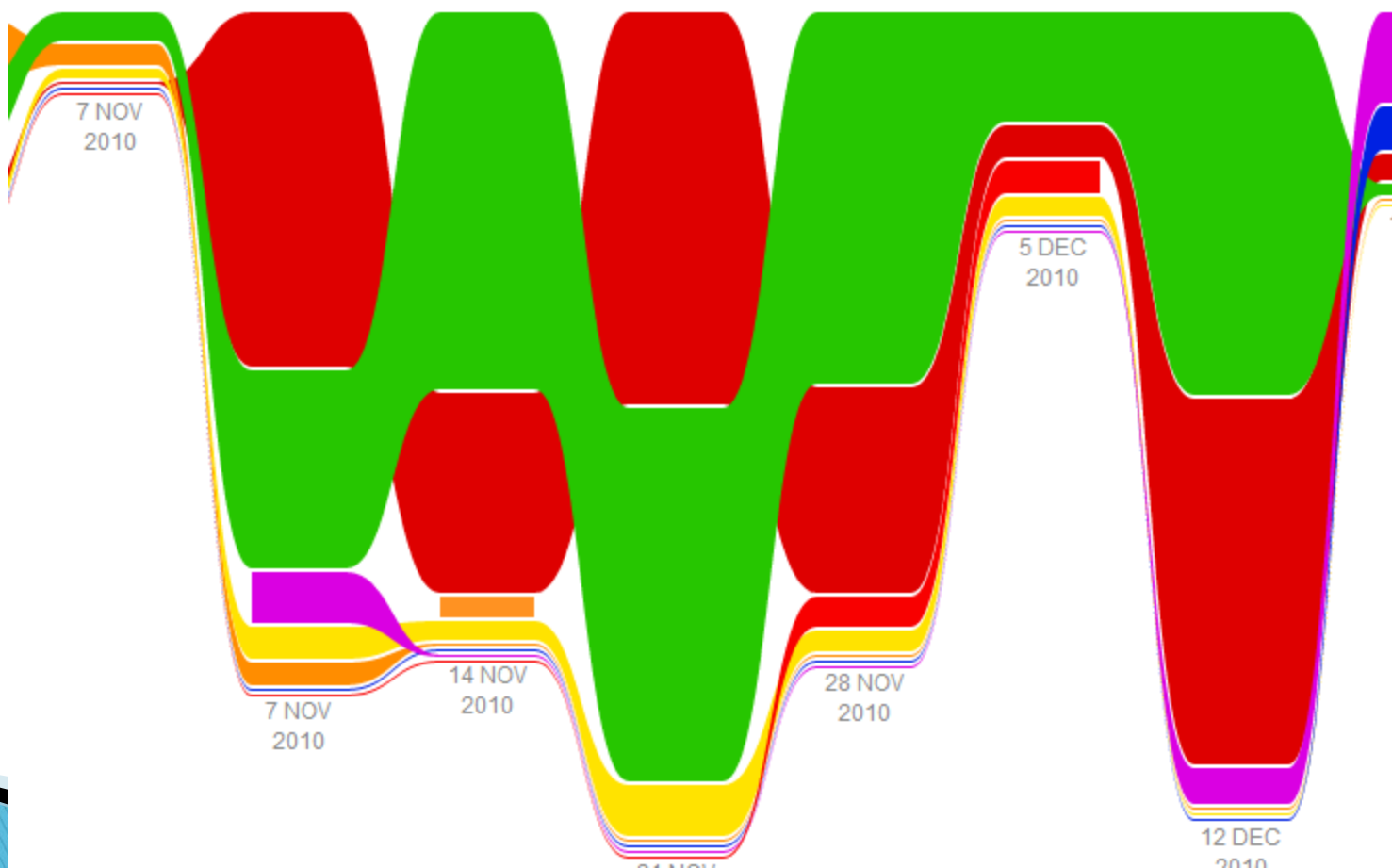
基于kissy独  
立开源项目

# kissyteam



# 期待你的参与

- ▶ [Kissyteam.github.com](https://Kissyteam.github.com)



# FAQ



- ▶ Gtalk/email: [yiminghe@gmail.com](mailto:yiminghe@gmail.com)
- ▶ Blog: <http://yiminghe.javaeye.com>