



Web 复杂应用的「渐进式」开发

百度 FEX - 张博

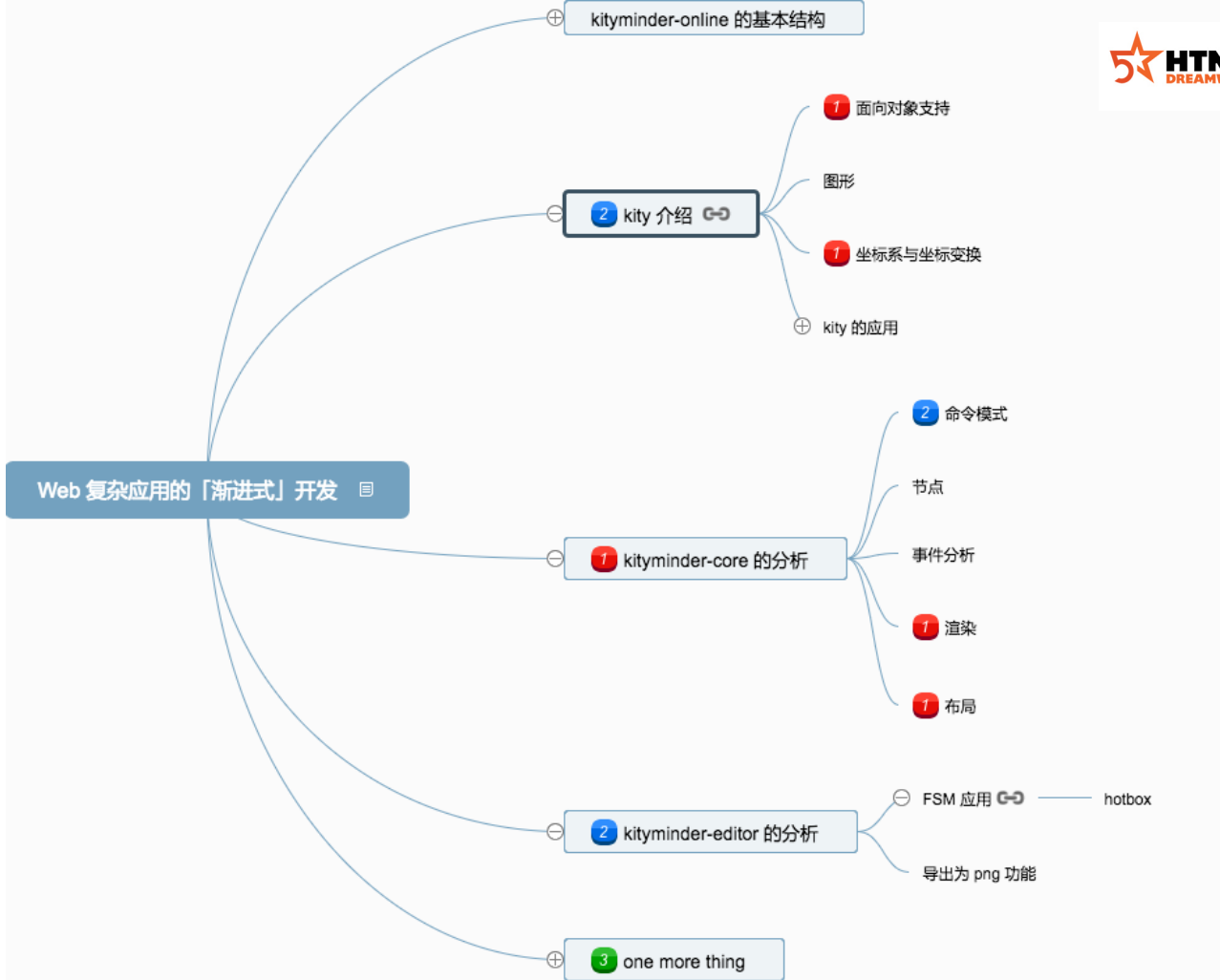
2017.04

| FEX 团队和个人介绍

- FEX 团队
 - UEditor：富文本编辑器
 - WebUploader：文件上传组件
 - FIS：前端集成解决方案
 - 百度脑图：在线思维导图工具
 - 百度 H5：在线 H5 制作工具
- 个人介绍
 - 百度脑图负责人
 - 百度 H5 开发者

| 内容概要

- 百度脑图简介
- 百度脑图结构
- Kity 介绍
- Kityminder-core 分析
- Kityminder-editor 分析
- One more thing



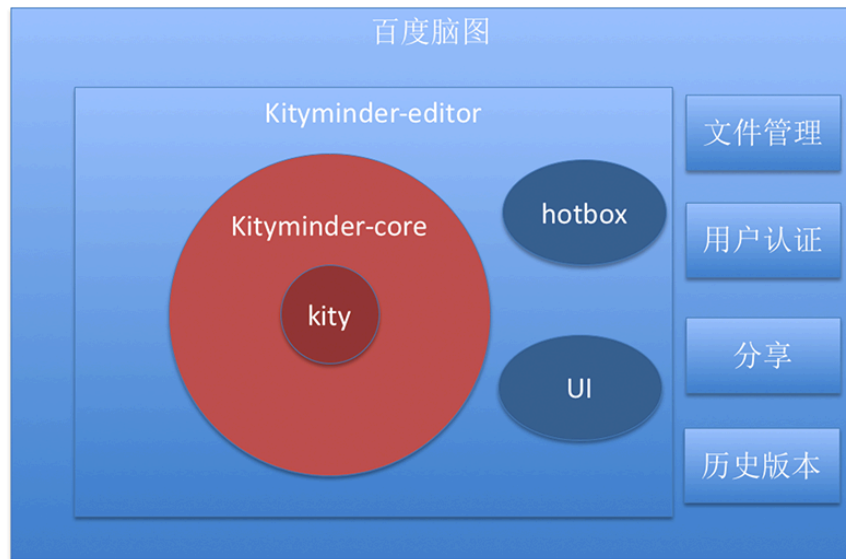
| 百度脑图是什么？

- 强大的在线思维导图编辑工具
- 用途：
 - 工作规划
 - 头脑风暴
 - 会议纪要
 - 读书笔记
 - 考试大纲



百度脑图结构

百度脑图结构



定制程度增加、通用性降低



思路 外观 视图

← 插入下级主题 插入上级主题 ↑ 上移 编辑 链接 图片 备注 × 1 2 3 4 5 6 7 8 9 × 添加

→ 插入同级主题 ↓ 下移 删除 5 6 7 8 9



Vertical toolbar with icons for zooming (plus, minus, double arrows), a central crosshair, a circular arrow, and a view icon. A preview window shows a zoomed-in view of the central node.



思路 外观 视图



插入下级主题 插入上级主题 插入同级主题

上移 下移

编辑 删除

链接

图片

备注

1 2 3 4 5 6 7 8 9



添加



Kity

| Kity 矢量图形库

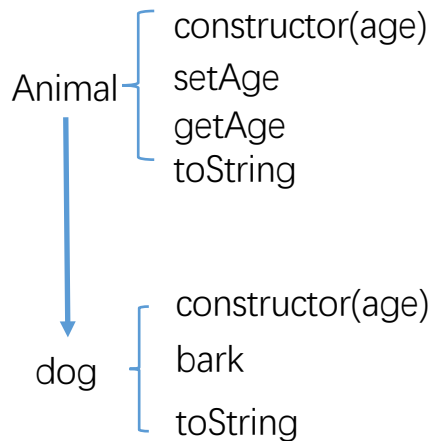
 <https://github.com/fex-team/kity>

- 面向对象支持
- 图形
- 坐标系
- 动画和时间线
- 事件
- 颜色和调色器

| Kity 面向对象支持

- 定义类
 - `Kity.createClass(className, classDefine);`
- 拓展类
 - `Kity.extendClass(class, extension);`
- 支持继承 (base)、混合(mixins)

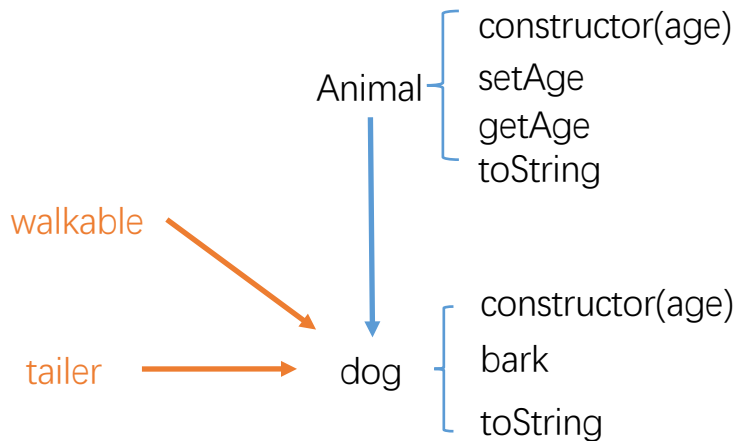
| Kity 继承



```
var Animal = kity.createClass({
  constructor: function (age) {
    this.setAge(age);
  },
  setAge: function (age) {
    this._age = age;
  },
  getAge: function () {
    return this._age;
  },
  toString: function () {
    return 'Animal aged ' + this.getAge();
  }
});
```

```
var Dog = createClass('Dog', {
  base: Animal,
  constructor: function Dog() {
    this.callBase(12);
  },
  bark: function () {
    return 'Dog bark';
  },
  toString: function () {
    return this.callBase() + ' Dog';
  }
});
```

| Kity 混合



```

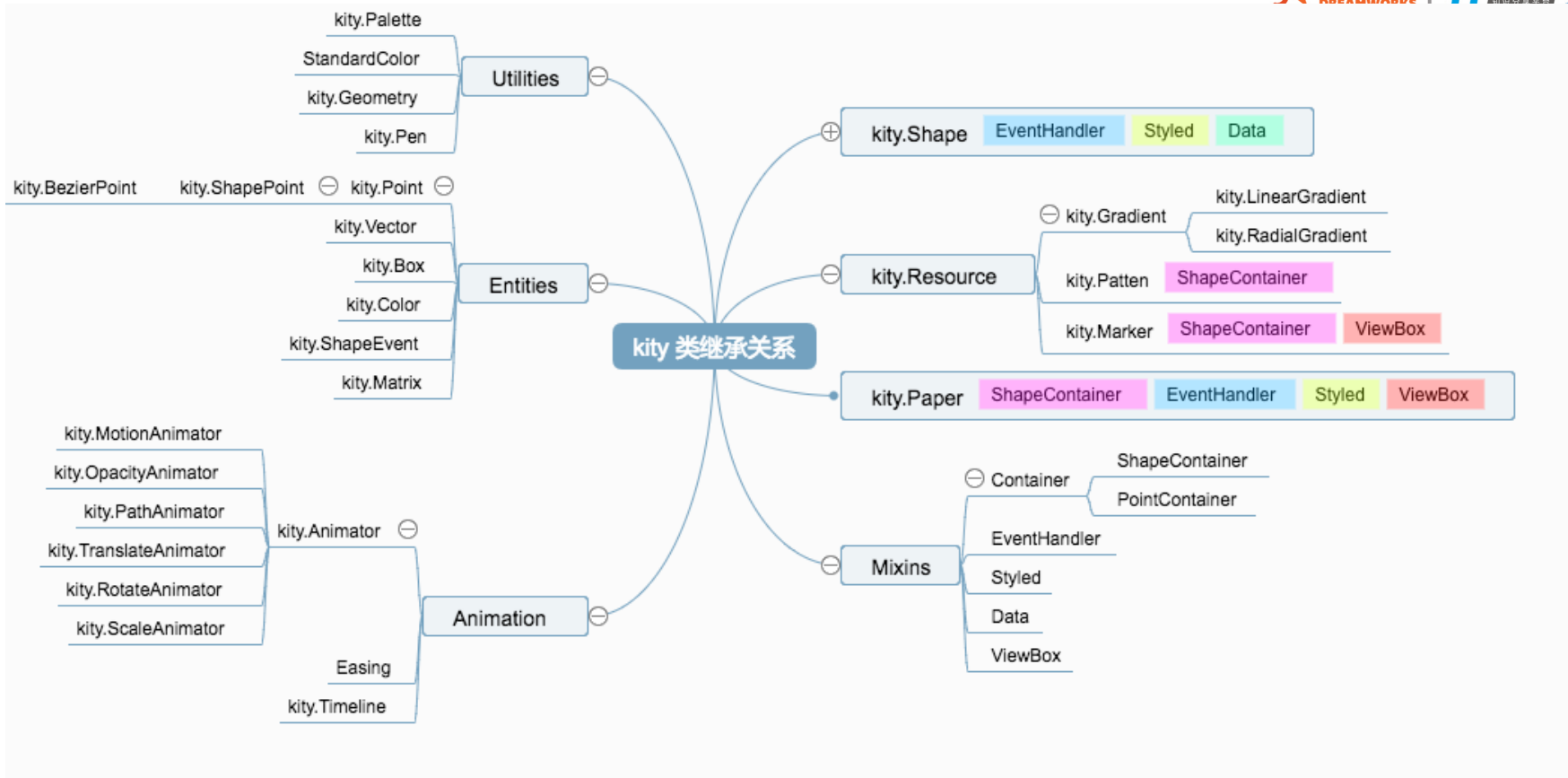
var Walkable = kity.createClass({
  walk: function (name) {
    return 'the ' + name + ' walk';
  }
});
  
```

```

var Tailer = kity.createClass('Tailer', {
  tail: function (length) {
    return 'tail with length ' + length;
  }
});
  
```

```

var Dog = createClass('Dog', {
  base: Animal,
  mixins: [Walkable, Tailer],
  dogWalk: function () {
    return this.mixin('walk', 'dummy') + 'and a dog';
  },
  tail: function () {
    return this.callMixin(12) + ' of a dog';
  }
});
  
```

| Kity 图形

- 渲染类
 - Line, Polygon, Bezier
 - Ellipse, Rect, Circle
- 文本类
 - Text, TextContent
- 容器类
 - Group

```
<div id="kity_paper"></div>

<script type="text/javascript">
  var paper = new kity.Paper('kity_paper');
  var rect = paper.put(new kity.Rect());
  var text = paper.put(new kity.Text());

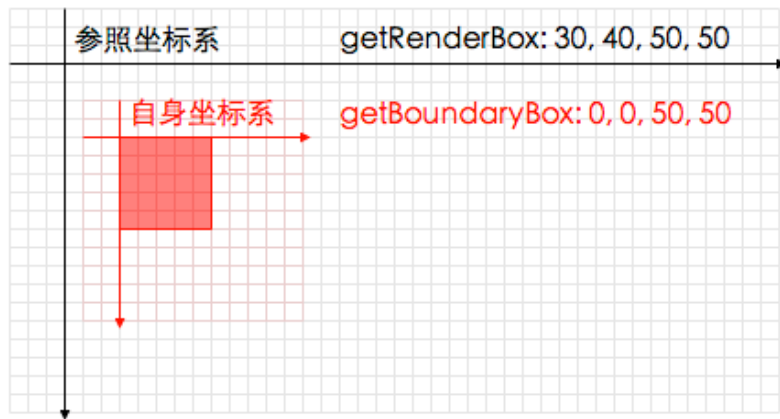
  text.setContent('hello kity!');
  text.fill('white');
  text.setX(100);
  text.setY(200);

  rect.setBox(text.getBoundingBox().expand(-15, -10, 15, 10));
  rect.setRadius(5);
  rect.fill('blue');
</script>
```

A blue rounded rectangle containing the text "hello kity!".

| Kity 坐标系

- 自身坐标系
 - 自带的
 - 只有一个
 - 用于定义图形
- 参考坐标系
 - 观察者选取的
 - 针对不同的参考物，有不同的坐标系
 - 用于观察图形



| Kity 坐标变换

$A(X, Y)$ 是坐标系 O_A 的任意一点, (X', Y') 是 A 在 O_B^A 中的坐标, 线性变换 T_B^A :

$$\begin{aligned} X' &= aX + cY + e \\ Y' &= bX + dY + f \end{aligned}$$

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & c & e \\ b & d & f \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

称为「二维线性变换矩阵」, 记为 M , 默认值为 $(1, 0, 0, 0, 1, 0)$ 。

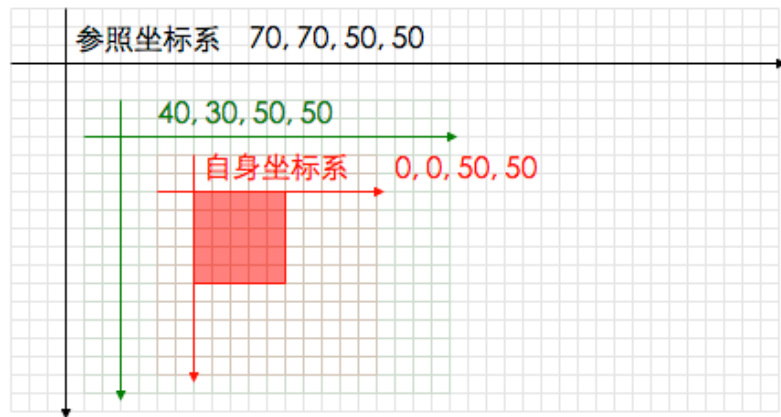
平移(translate) : $(1, 0, e, 1, 0, f)$

旋转(rotate) : $(\cos \theta, \sin \theta, 0, -\sin \theta, \cos \theta, 0)$




缩放(scale) : $(\alpha, 0, 0, 0, \beta, 0)$

| Kity 坐标变换函数

- 相对于指定坐标系的线性变换
 - `Shape.getTransform(refer) : matrix`
- 相对于图形自身坐标系的区域
 - `Shape.getBoundingBox() : box`
- 相对于指定坐标系的区域
 - `Shape.getRenderBox(refer) : box`
- 事件发生时相对指定坐标系的位置
 - `ShapeEvent.getPosition(refer): Point`



| 基于 Kity 的复杂应用

kityminder	kityformula	shushuo
 百度脑图	 百度公式	 数说
强大的在线脑图编辑器	前端数理化公式解决方案	专业的数据可视化服务
naotu.baidu.com	gongshi.baidu.com	shushuo.baidu.com

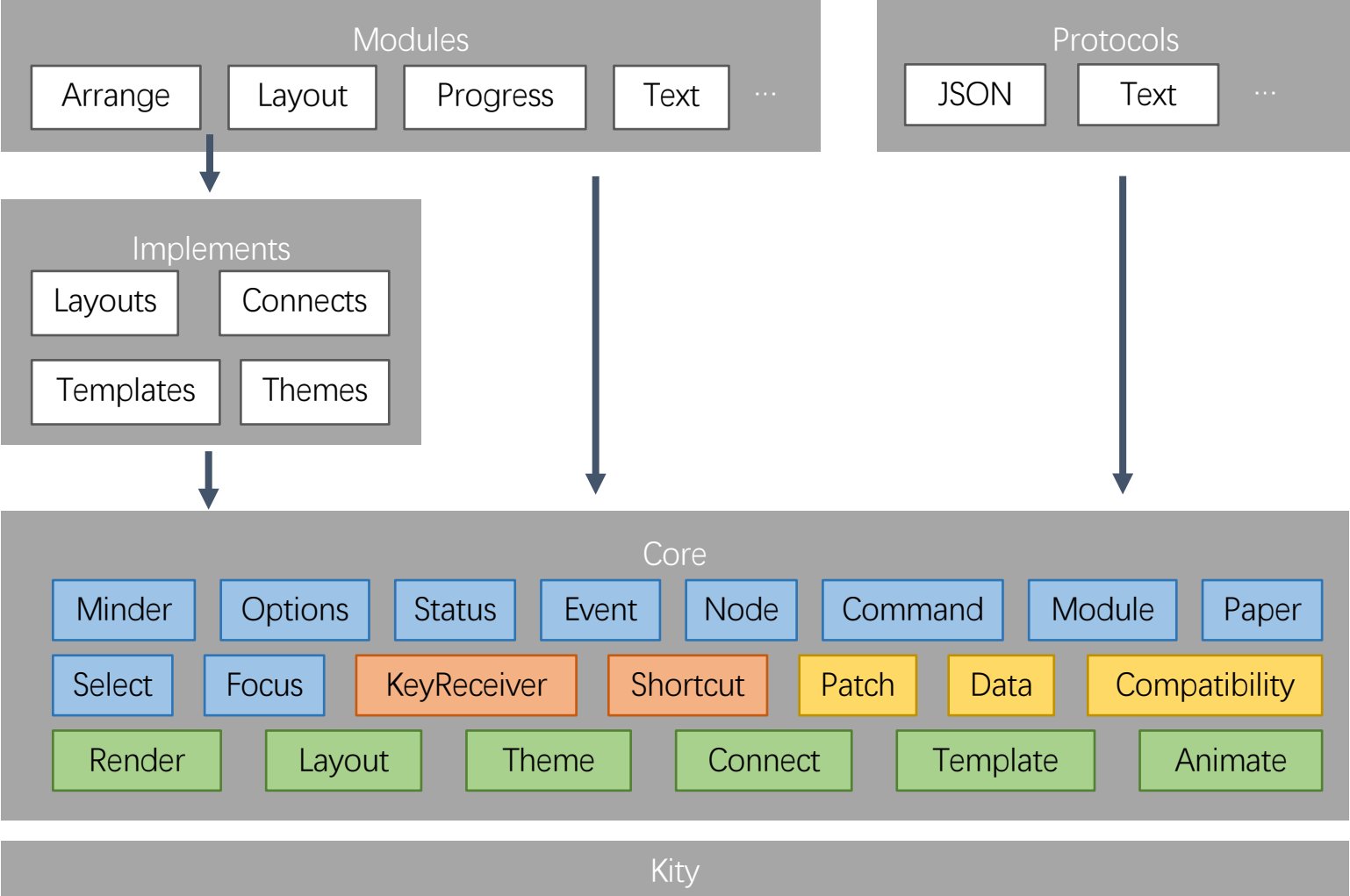
更多内容请参考 <http://kity.baidu.com>

Kityminder-core

| Kityminder-core

 <https://github.com/fex-team/kityminder-core>

- 总体结构
- 命令
- 节点
- 事件
- 渲染
- 布局



| Kityminder-core 命令

- `execCommand(command, args)`
 - 执行命令
- `queryCommandState(command)`
 - 查询命令状态：可执行(0)、不可执行(-1)、已执行(1)
- `queryCommandValue(command)`
 - 查询命令结果

例如：`minder.execCommand('expand')` – 展开当前节点

| Kityminder-core 常见命令

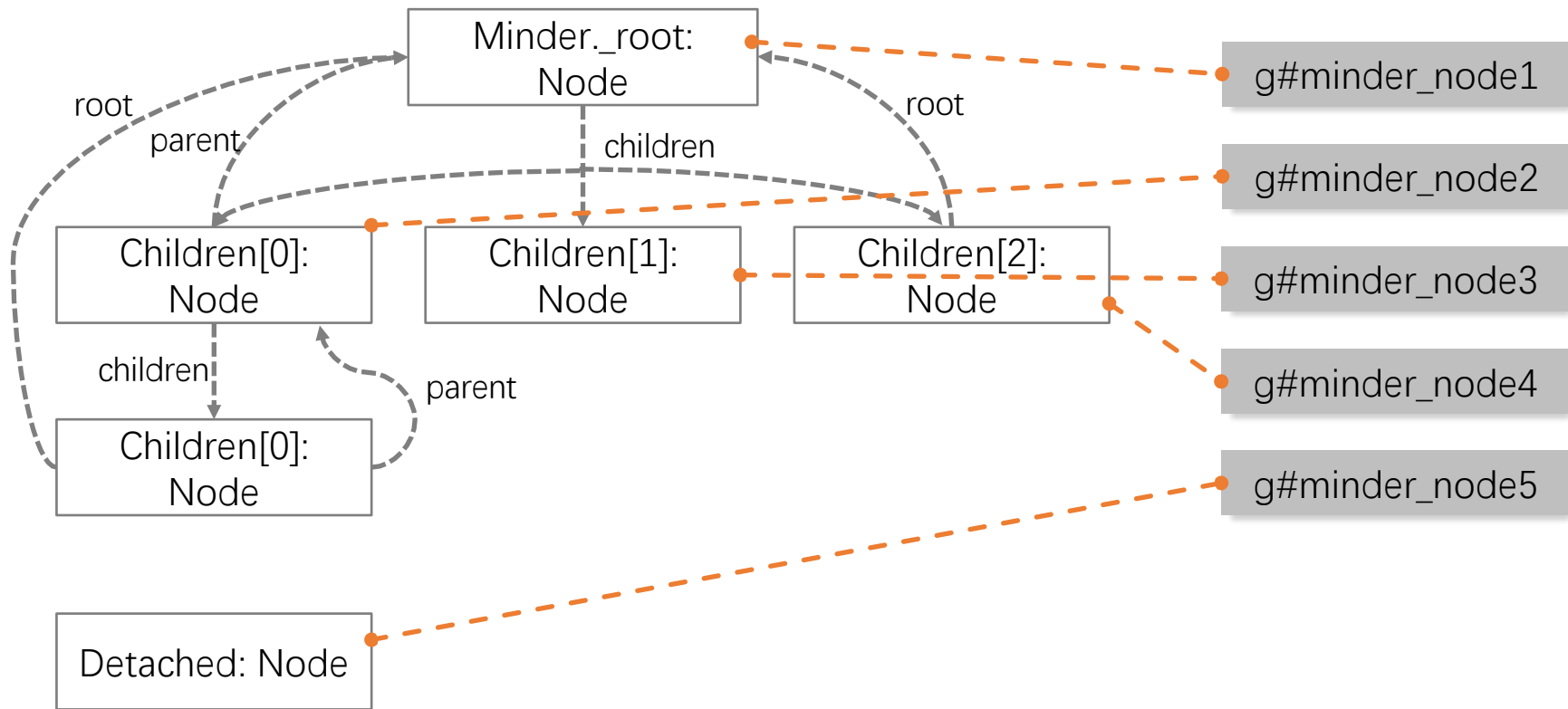
- AppendChildNode : 添加子节点到选中的节点中
- Arrange : 调整选中节点的位置
- Camera : 设置当前视野的中心位置到某个节点上
- CopyStyle : 拷贝选中节点的当前样式
- Image : 为选中的节点添加图片
- Progress : 设置节点的进度信息

更多命令请参考 <https://github.com/fex-team/kityminder-core/wiki/command>

| Kityminder-core 节点数据结构

- parent : 父节点
- children : 子节点数组
- root : 根节点
- data : 节点数据, 如: 文字、备注、优先级
- rc : renderContainer, 节点渲染出的容器
- attached : 是否独立节点
- type: root, main, sub

Kityminder-core 节点图示



| Kityminder-core 节点方法

- 树相关
 - isRoot, isLeaf, getCommonAncestor, getLevel, tranverse...
- 数据相关
 - getData, setData, getText...
- 运行时支持
 - createNode, appendNode, detachNode...

| Kityminder-core 事件

- 作用
 - 代理 kity 事件
 - 内外通信, 模块解耦
- 使用
 - 监听和取消: on/off 函数
 - 触发: fire
- 阶段
 - fire 内部分为四个阶段: before, pre, on, after
 - 在 before 阶段可以阻止事件

| Kityminder-core重要事件

- on('contentchange')
 - 内容改变时触发
- on('interactchange')
 - 产生交互时触发（可理解为风吹草动）
- on('beforeexeccommand')
- on('preexeccommand')
- on('execcommand')
- on('afterexeccommand')
 - 命令被执行时依次被触发

| Kityminder-core 渲染

- 渲染：决定节点外观，并在渲染后提供 contentBox
- 目前共有 10 个渲染器 (renderer)
 - Text
 - Progress
 - Priority
 - Hyperlink
 - Note
 - Image
 - Resource
 - Outline
 - Expander
 - shadow



| Kityminder-core 渲染

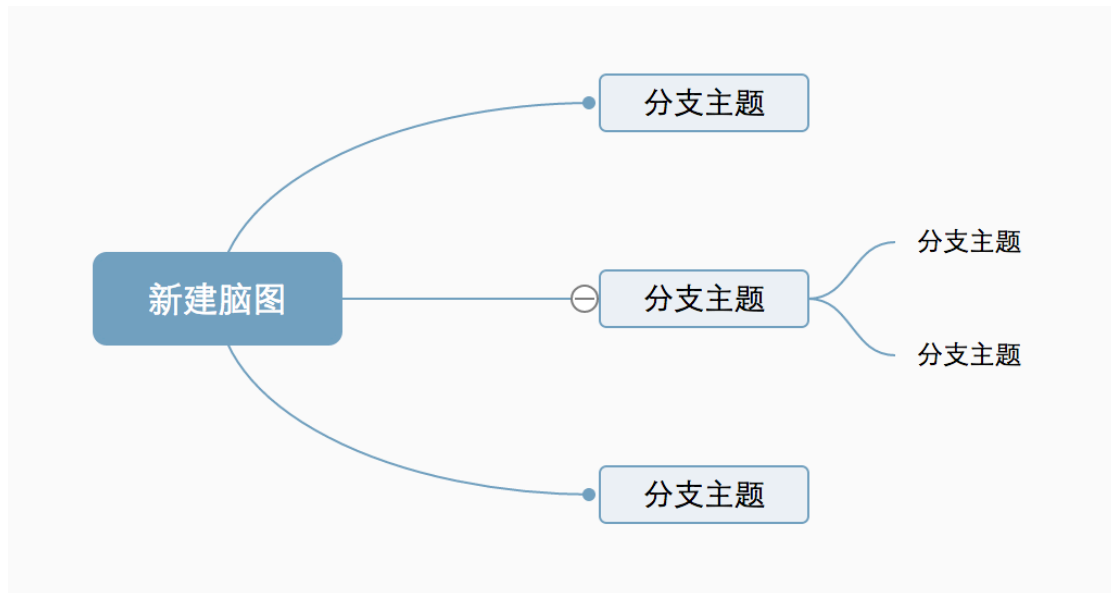
- 渲染器基类 core/render 提供运行时支持
- 位置：center, top, bottom, left, right, outline, outside
- 结果：getContentBox



| Kityminder-core 布局

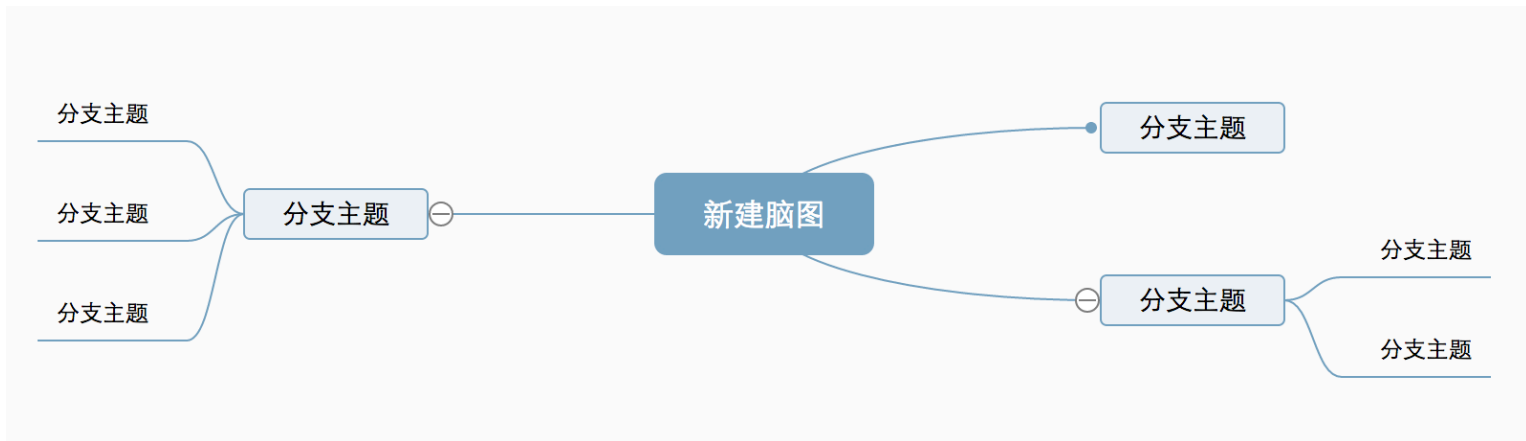
- 布局：决定一个节点相对父节点的位置，并提供一个 layoutBox
- 目前提供了 5 种主要布局
 - btree: left, right, top, bottom
 - mind
 - fileTree: filetree-up, file-tree-down
 - tianpan
 - Fish-bone: fishbone-master, fishbone-slave
- 模板是子布局和连线的组合

| Kityminder-core 布局



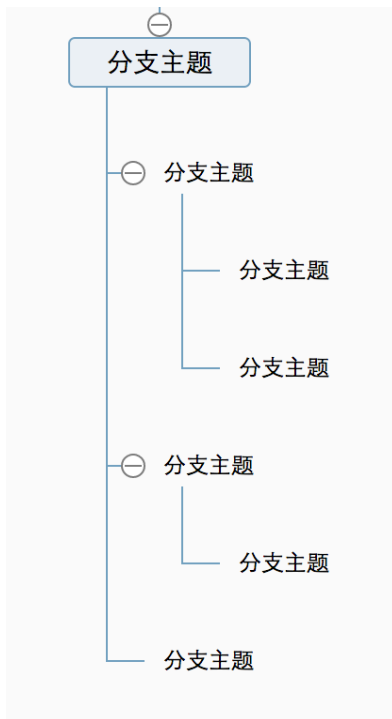
Btree-right

| Kityminder-core 布局



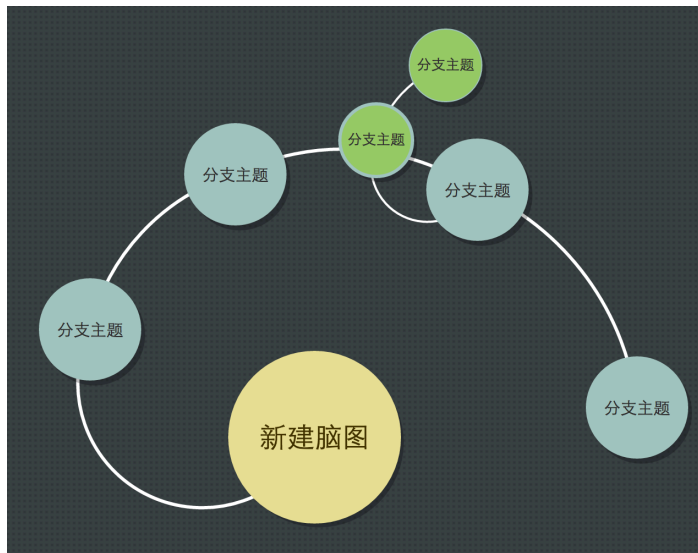
Mind

| Kityminder-core 布局



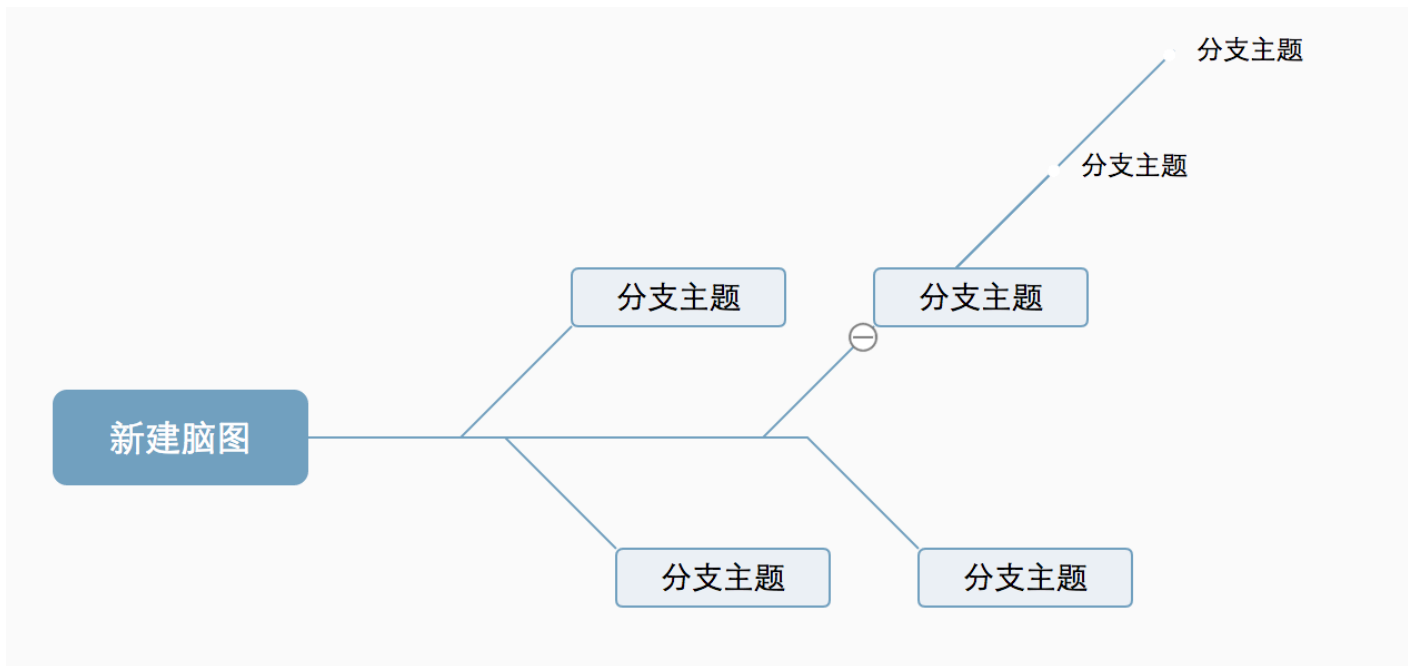
Filetree-down

| Kityminder-core 布局



tianpan

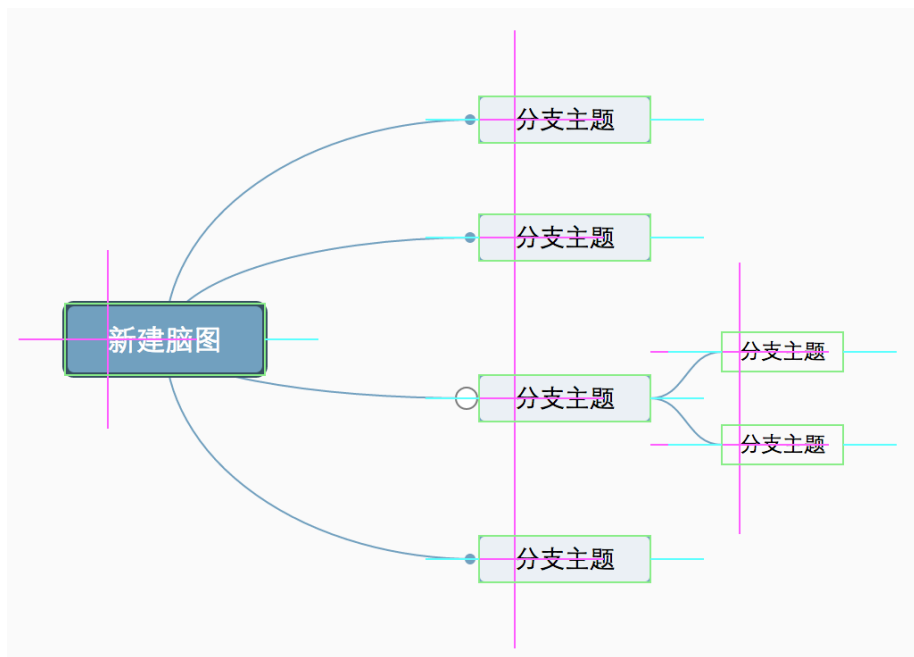
| Kityminder-core 布局



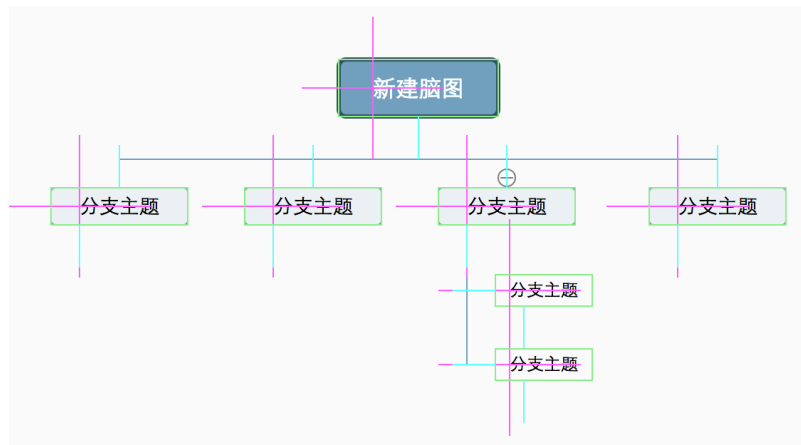
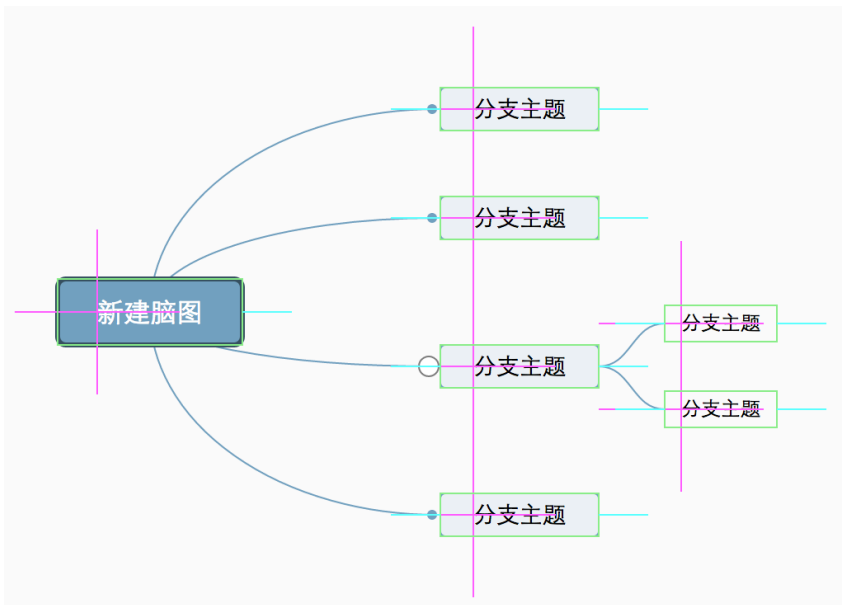
fishbon
e

| Kityminder-core 布局

- 本质：子节点相对父节点的位置
- 过程（以 btree 为例）：
 1. 子树 align 和 stack
 2. getBranchBox 根据主题设定，再次调整
- 结果：getLayoutBox



| Kityminder-core 连线



- 基础：vertexIn/Out, vectorIn/Out, 决定了连线的出入点/方向

| Kityminder-core 模板



思维导图

Mind + btree:left/right

Arc + under



组织结构图

Btree:bottom

Poly



鱼骨图

Fishbone:master/slave

Fishbone + L



文件结构图

Btree:bottom + filetree:down

Poly + L



逻辑结构图

Btree:right

Arc + bezier



天盘图

tianpan

tianpan

Kityminder-editor



| Kityminder-editor

 <https://github.com/fex-team/kityminder-editor>

- 基本原理
- hotbox 与有穷状态机
- 探究：导出为 png 功能

| Kityminder-editor 基本原理

- 基于 Angular 1.x
- 依赖于 execCommand 系列函数
- 添加了 hotbox 和 UI (button, dialog 等)

思路

外观 视图



插入下级主题

插入上级主题

上移

编辑



1

2

3

4



添加

插入同级主题

下移

删除

链接

图片

备注

5

6

7

8

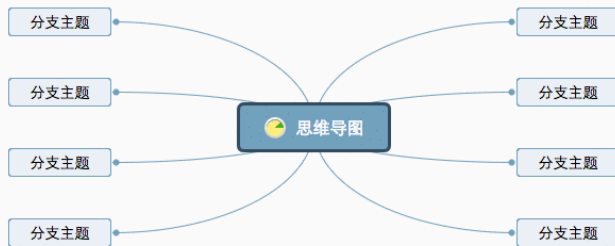
9



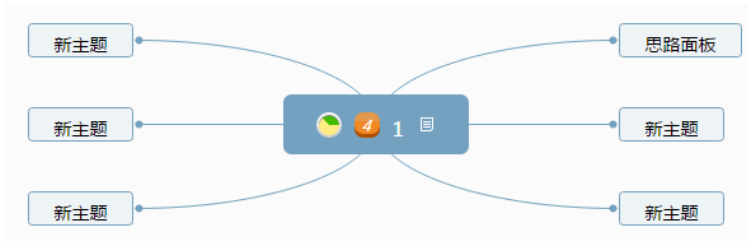
| Kityminder-editor hotbox

 <https://github.com/fex-team/hotbox>

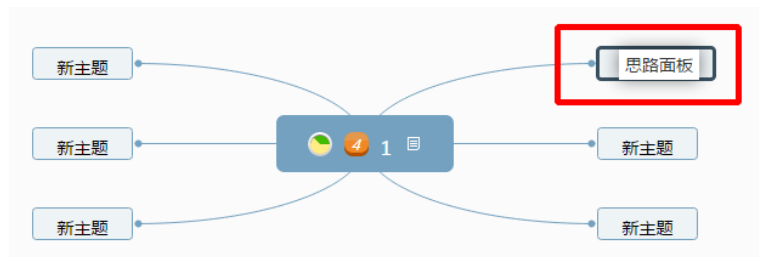
- 热盒 UI 是一种高效的上下文交互方式



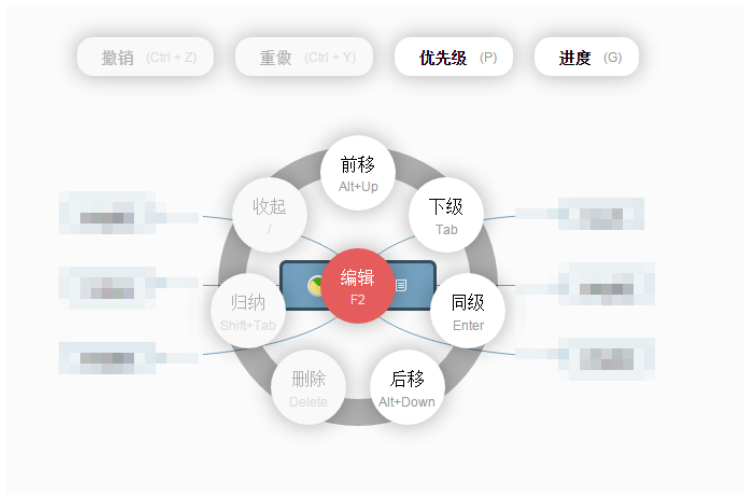
| Kityminder-editor 有穷状态机



Normal

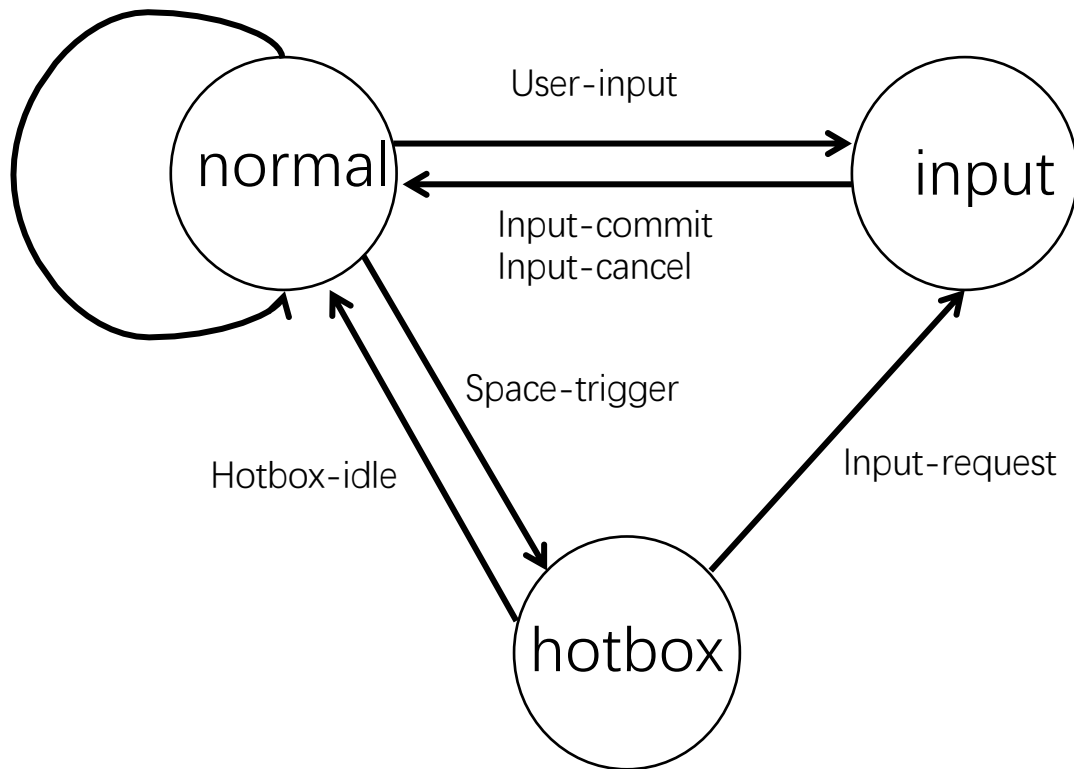


input

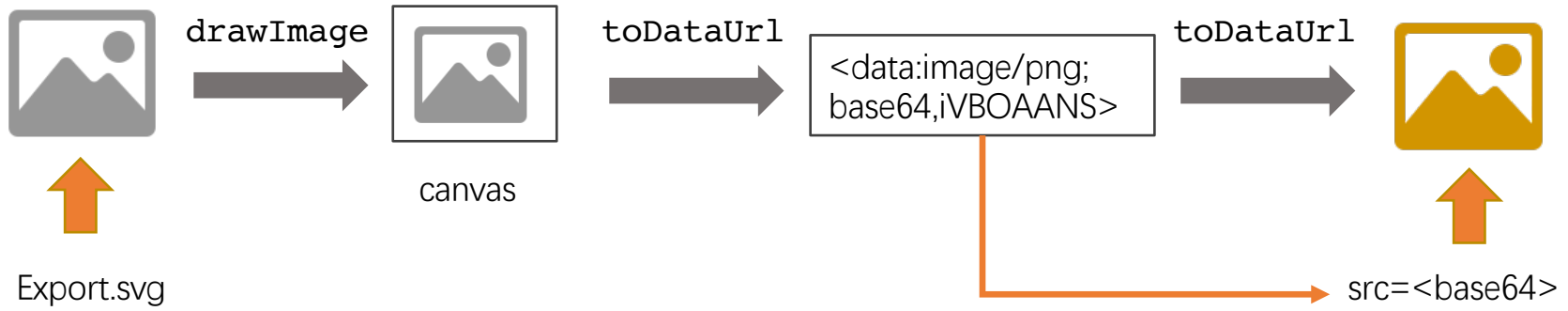


hotbox

| Kityminder-editor 有穷状态机



| SVG 导出为 PNG



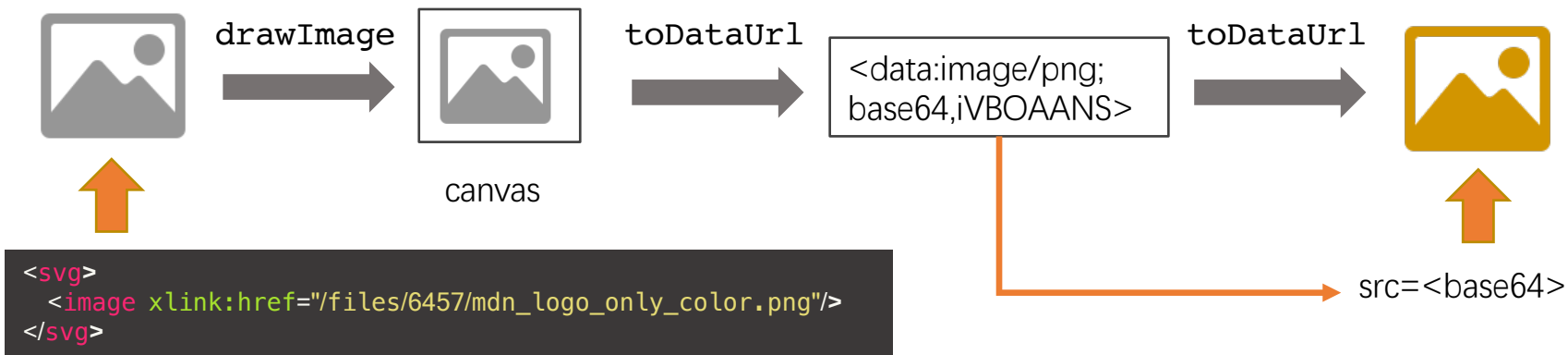
| SVG 导出为 PNG

- 问题一：canvas 尺寸限制（单位：像素）
 - Chrome: 16,384 * 16384
 - Firefox: 32, 767 * 16384
 - IE 10: 8,192 * 8192

解决办法：切成 Tile，分片导出

| SVG 导出为 PNG

- 问题二：svg 包含图片，导不出来



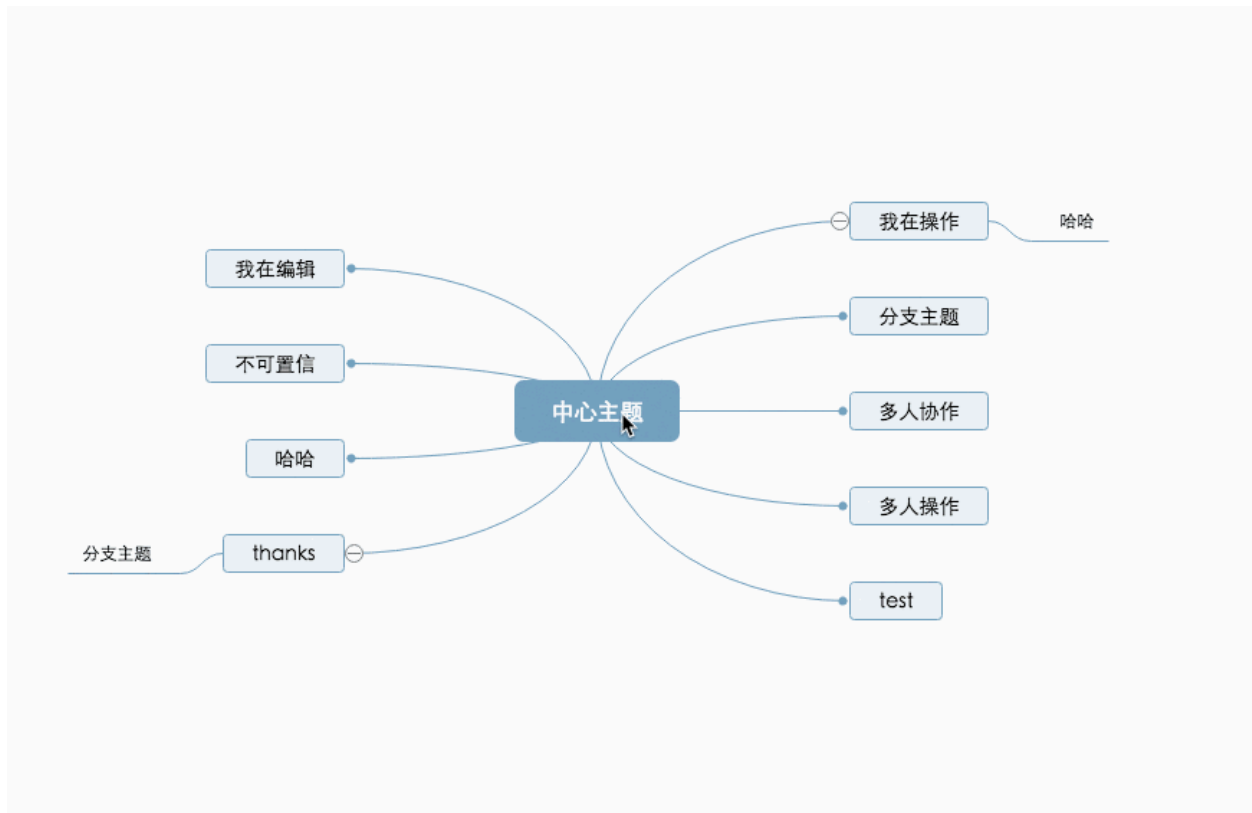
解决办法：svg 安全策略限制，手动将图片贴到 Canvas 上

更多内容：<http://fex.baidu.com/blog/2015/11/convert-svg-to-png-at-frontend/>

One more thing



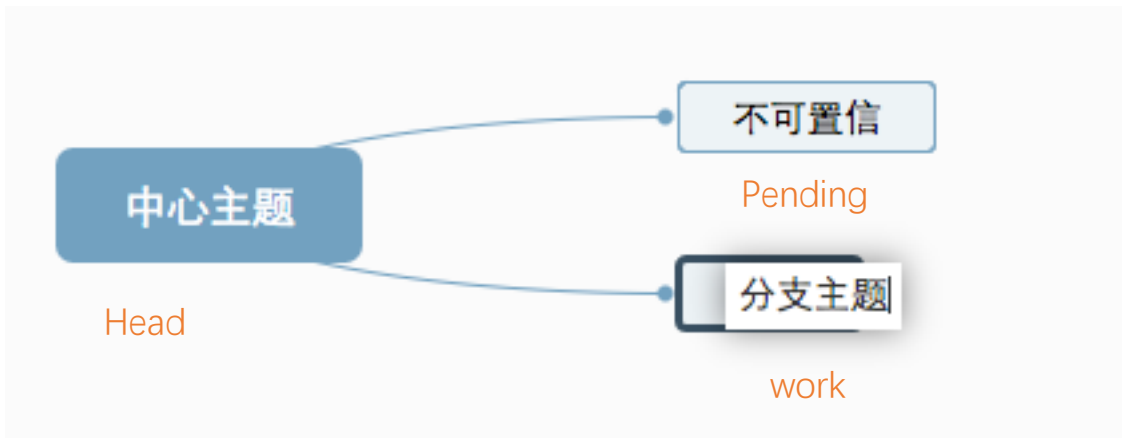
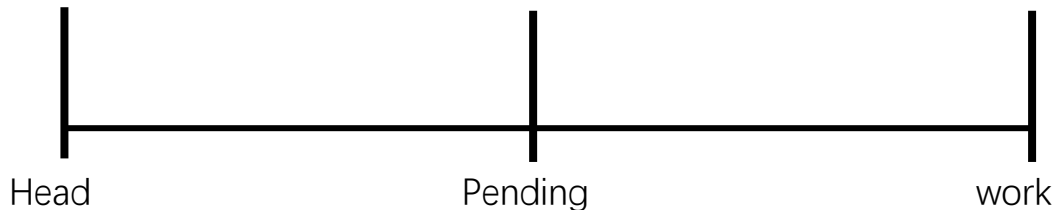
多人实时协同编辑

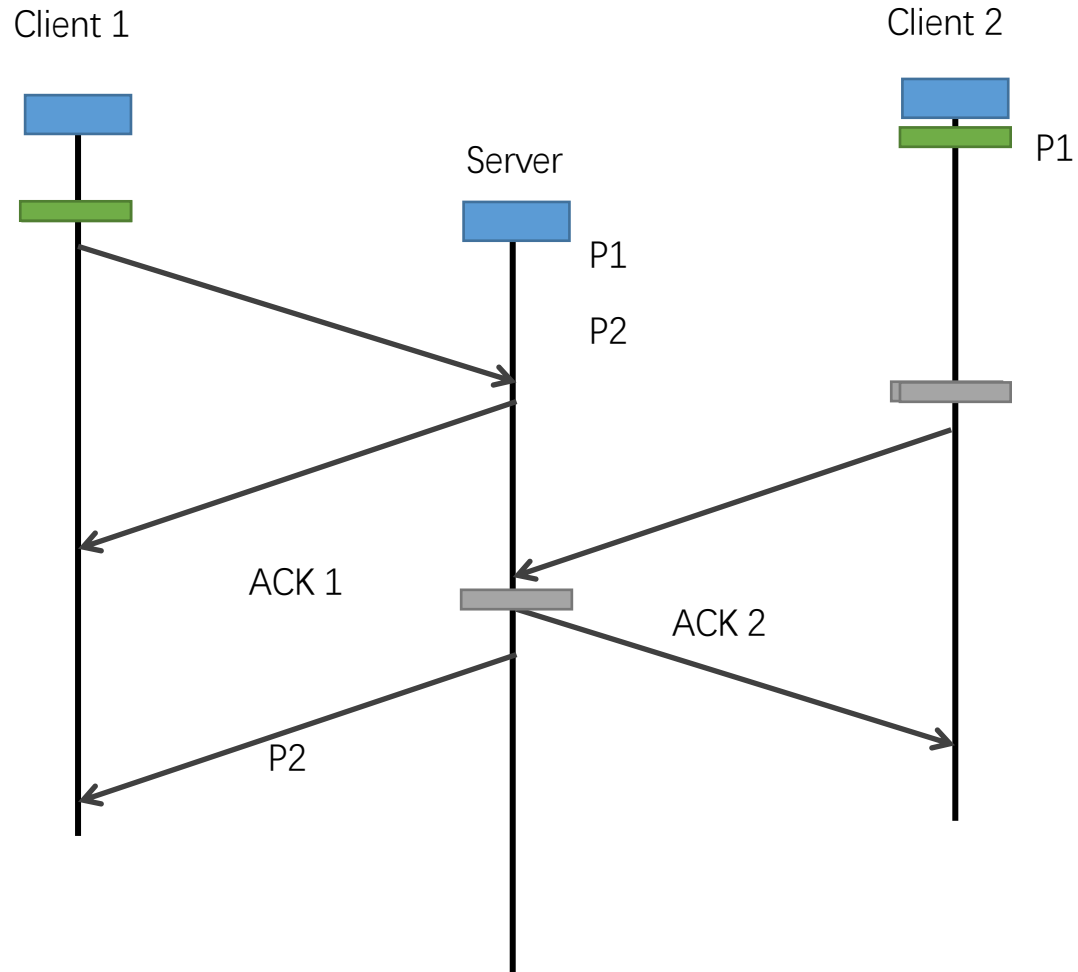


| 多人实时协同编辑

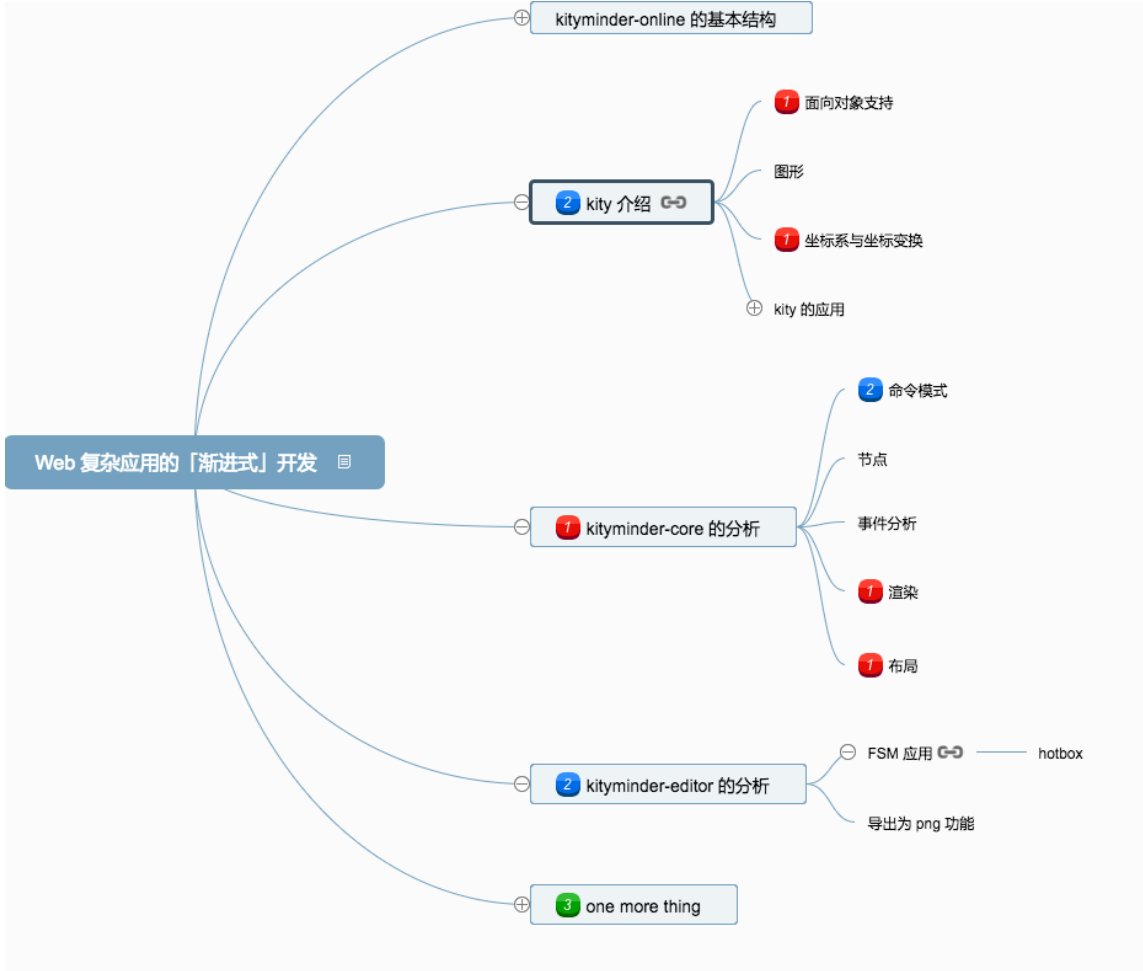
- 基于 websocket 技术：Socket.io
- 数据包：json-diff 形成的 patch

| 多人实时协同编辑





| 总结



- 逻辑分层平衡了项目开源和自身应用，不同的人群各取所需。

Thanks !



Q&A

