



# Python 让前端更轻松

基于 Python 的前端方案实践

潘俊勇 [weibo.com/panjunyong](http://weibo.com/panjunyong)

[easydo.cn](http://easydo.cn) 易度云办公

# 前端开发的困惑

知乎

搜索话题、问题或人...



提问

首页

话题

发现

消息



潘俊勇

前端开发

HTML5

模块化

修改

还要多少年, 前端开发才能像客户端开发那样轻松? 修改

标题党了,, 具体说意思吧, 看到 Web Components 规范进展很多, Google 的 Polymer, Mozilla 的 X-Tags, Facebook 的 React, 以及一些 MV\* 框架, 比如 Angular, Ractive, 都在使用自定义标签功能加强抽象. HTML 原先抽象起来太艰难了, 现在终于好转.. 我想知道, 要多少年才能到大家都能顺手呢? 修改

20 条评论 分享 邀请回答

举报

32 个回答

按票数排序

349

徐飞, RIA

linzhe cai、唐伟诚、shell von 等人赞同

我们来分析一下究竟哪些因素让前端开发这么困扰。

取消关注



653 人关注该问题



产品设计师  
技能书



知乎 马力 自选集 | 「鹽」系列

相关问题

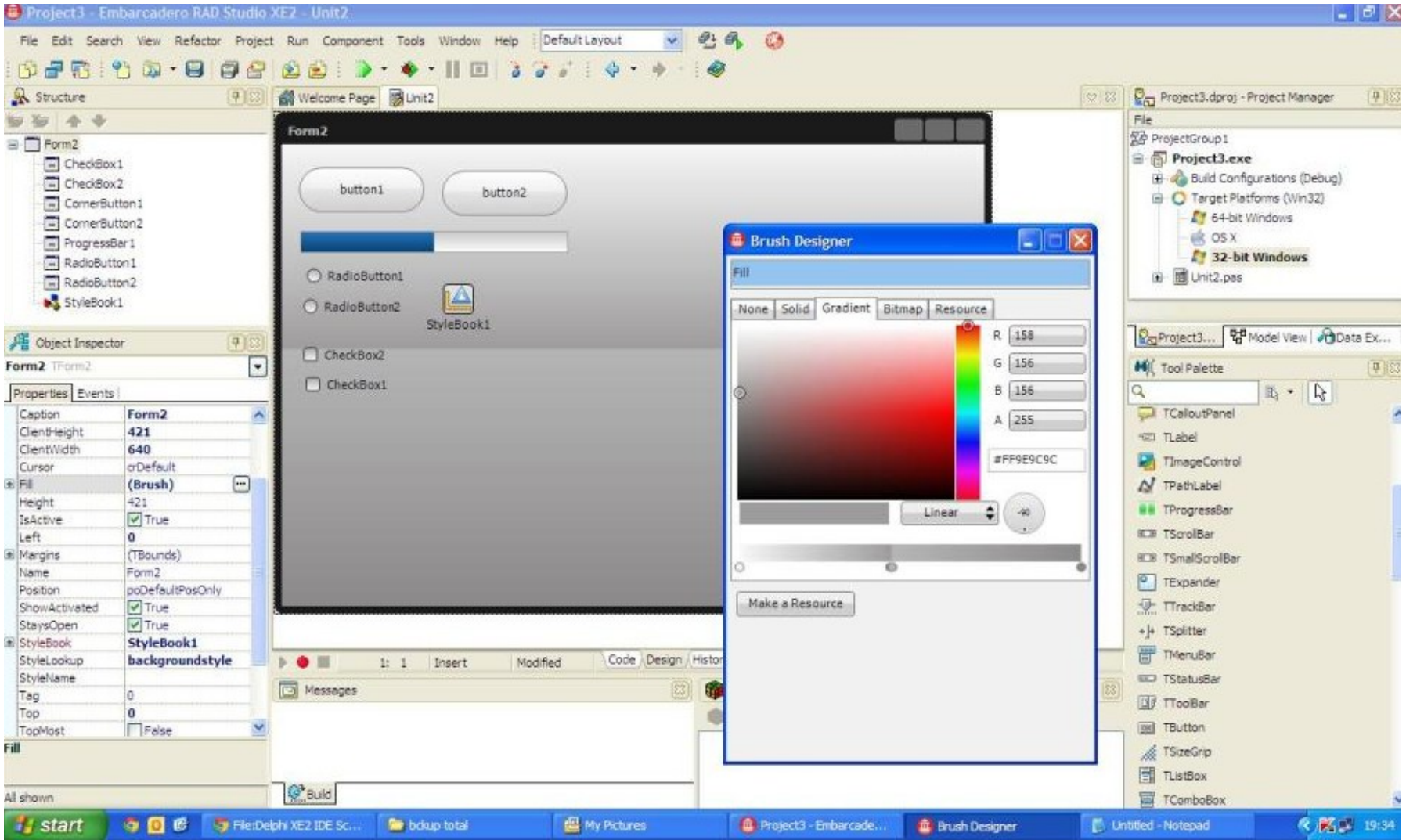
成熟工具链支撑简单快速开发

# 桌面 C/S 时代：曾经的美好

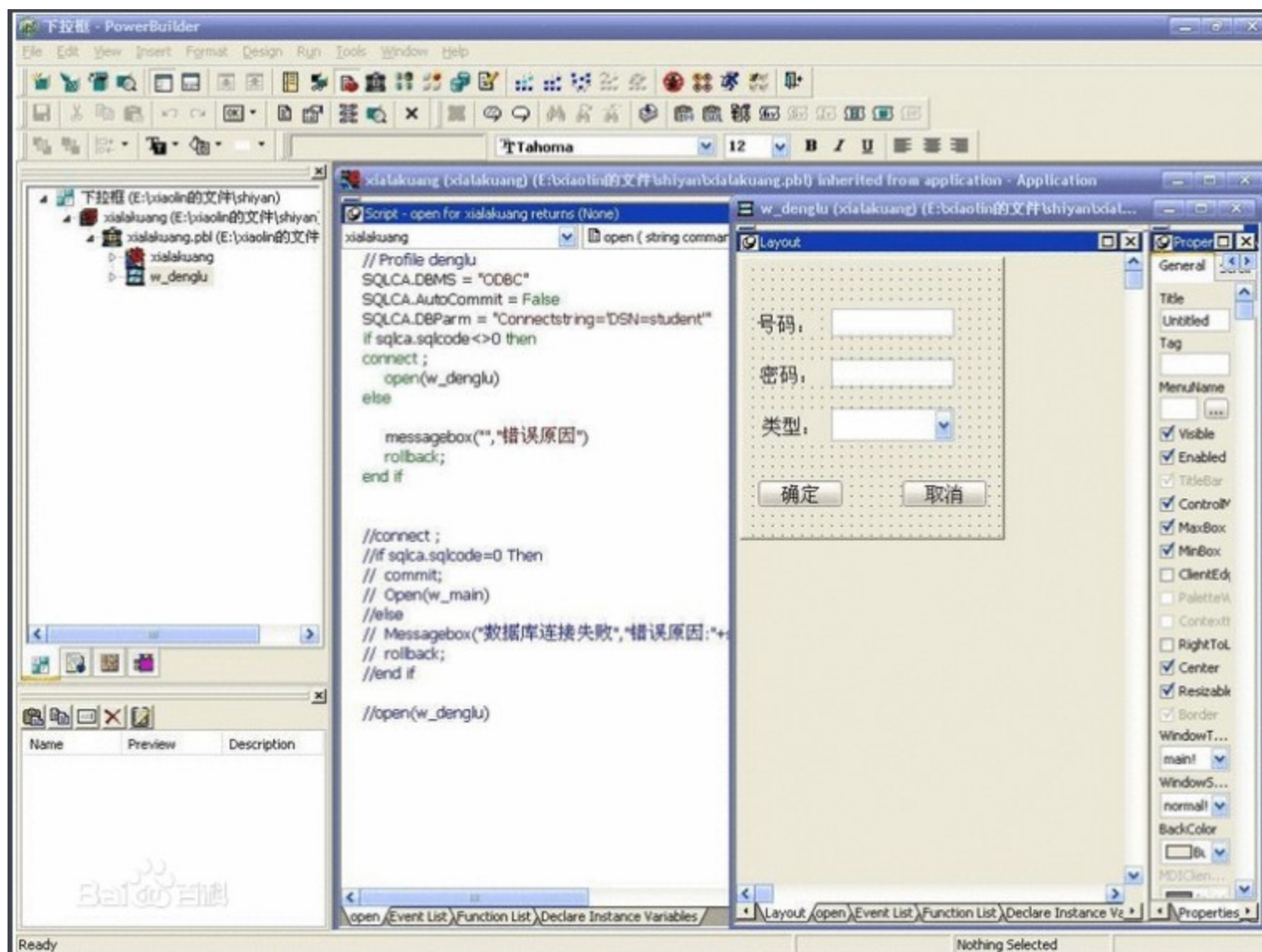


**PyCon 2014** 北京 上海 珠海 苏州 杭州 主办: PyChina.org, GDG

# 组件编程，图形化 IDE



# PowerBuilder : 图形化, 脚本化



数据库应用开发神器

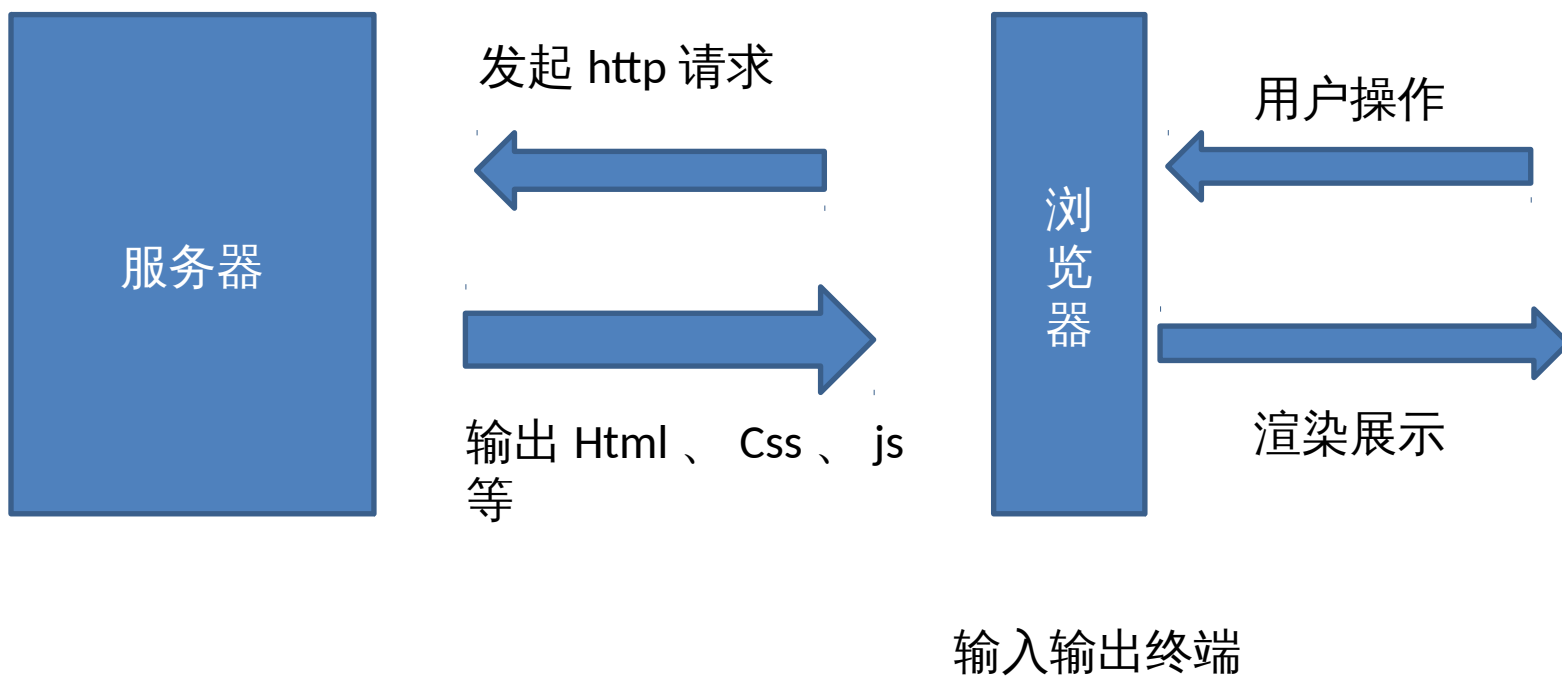
# 效果

- 更低的成本
- 对开发人员要求低
- 不分前后端
- 更高的开发效率

桌面安装复杂，浏览器打开即用，世界进入 B/S 时代

# WEB 前端 – B/S 时代

# B/S 模型：Thin Client





# 问题：更多语言需要学习

- HTML
- CSS
- Javascript

# 减少语言

- Nodejs : 后端也用 js 来开发
- 曾经的 KSS : 用类似 css 的语法写 JS
- PythonJS :
  - 浏览器上 Python 开发
  - 翻译为 JS
  - 玩具性质

# 问题：事件 变成 http 请求

- route : url 地址和代码的映射
- 安全性
- 无状态 : cookie/session

# 用约定来简化

- Rails
- Django

# 问题：不成熟的浏览器

- Html/js/css 都在不断在发展
- Javascript 自身的缺陷
- 各种浏览器兼容问题
- 第三方组件兼容性问题

# 还好有前端框架

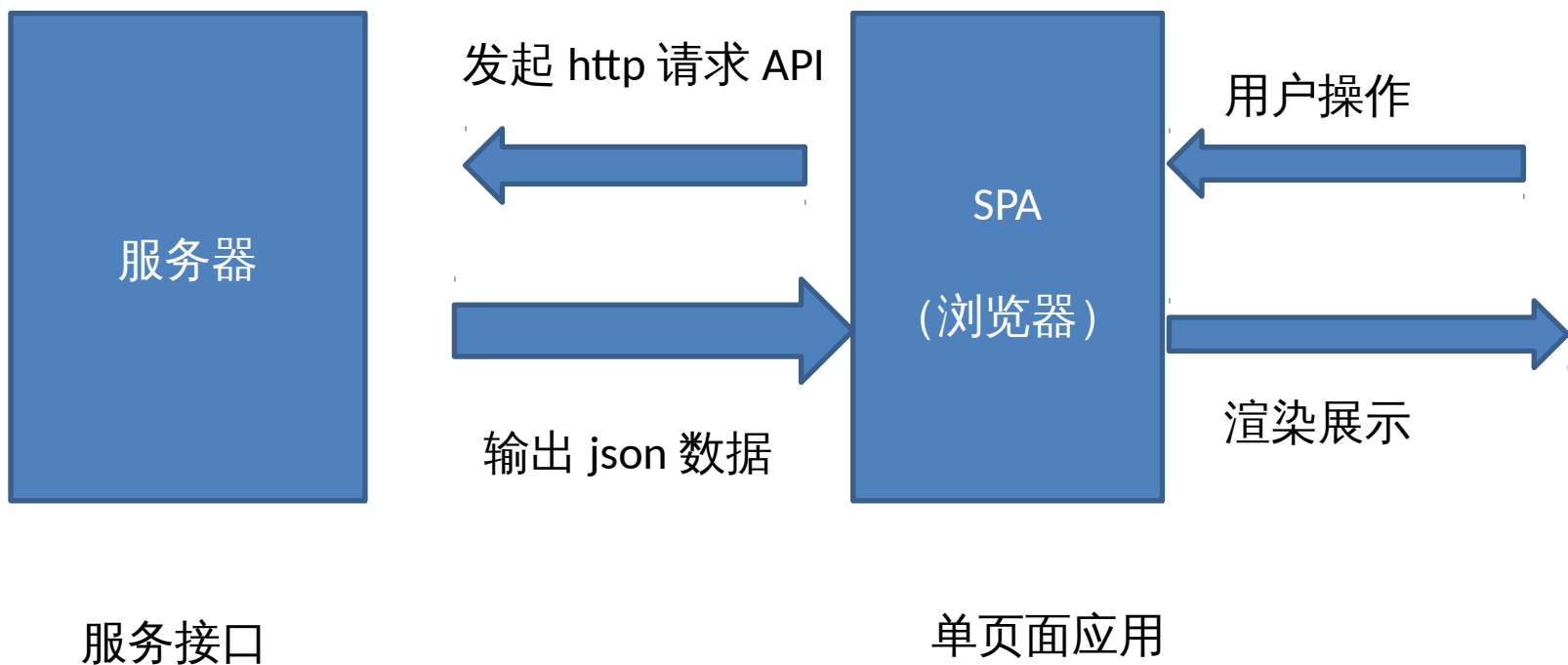
- Twitter Bootstrap
- Semantic UI
- JQuery UI
- 国内的
  - 淘宝 BUI
  - 百度 BlendUI
  - ...

移动端和体验要求，系统要求更开放，进入单页面应用时代

# WEB 前端 – 应用时代

# 单页面应用模型：Fat Client

SPA：Single Page Application，单页面 web 应用





# 特点

- 大量逻辑前移到浏览器
- 服务端只提供 API
- 对搜索引擎支持不好
- 前端特别重

# OpenERP : 前端有多重？

## Technological changes

6.0	6.1
Python (+ javascript)	Javascript
Mako	QWeb
CherryPy	Werkzeug
Pages	Single Page Interface
Page requests	JSON-RPC

# OpenERP 6.0 : 一个 Python 应用

cloc: 6.0

Python	16059
Javascript	9782
Templates	5905

# OpenERP 6.1 : 一个 JS 应用 ?

cloc: 6.1

Javascript	16925
Templates	3693
CSS	3064
Python	1874

# 前端 MVC 框架，蜂拥而至

- Backbone
- AngularJS
- Knockout
- ReactJS
- ...

# 前端开发工具链

- Coffeescript
- Less/Sass
- RequireJS/SeaJS
- Bower
- Yeoman
- Grunt

# 前端的迷失

- 前端开发变得异常复杂
- 各种框架轮番上阵
  - 各自存在问题
  - 互不兼容
  - 仍未终结
- 前端程序员很渺小

# 另外的声音：前端去框架才是趋势

## Facebook工程师谈前端：去框架化是趋势

受访者 [Hedger Wang](#) 作者 [杨赛](#) 发布于 2013年12月12日 | [1](#) 讨论

分享到：[微博](#) [微信](#) [Facebook](#) [Twitter](#) [有道云笔记](#) [邮件分享](#)



脚

8.

去框架：让开发者开发所谓的微框架或者小框架，然后呢让它们之间能够彼此相容，然后你竞争去采用比较好的东西



我们怀念桌面时代的美好

# 前端欠发展

- 前端太难
- 优秀的 web 前端很贵… .
- 技术一定往简化发展
- 前端可能是碗青春饭

修修补补，还是根源解决？  
如何省时、省人、省力，技术上可持续发展？

# 怎么破？

修修补补：如何恢复桌面时代的美好

# 微软的尝试：ASP.NET

# 组件：ASP.NET Helpers

---

## WebGrid 帮助器

WebGrid 帮助器简化了显示数据的方法：

- 自动建立显示数据的 HTML 表格
  - 支持不同的格式化选项
  - 支持对数据的分页（第一页、下一页、上一页、最后一页）
  - 支持通过点击列标题进行排序
- 

## Chart 帮助器

"Chart Helper" 能够显示出带有不同格式选项和标记的不同类型的图表图片。

Chart 帮助器能够显示数组、数据库或文件中的数据。

---

## WebMail 帮助器

WebMail 帮助器提供了使用 SMTP (Simple Mail Transfer Protocol) 来发送电邮消息的功能。

---

## WebImage 帮助器

WebImage 帮助器提供了在网页中管理图片的功能。

关键词：剪裁、旋转、调整尺寸、水印。

---

在网页中内嵌帮助器，简化控件的创建

# WebForm 服务器控件：失败

HTML 服务器控件	描述
<a href="#">HtmlAnchor</a>	控制 <a> HTML 元素
<a href="#">HtmlButton</a>	控制 < <b>Web 服务器控件</b>
<a href="#">HtmlForm</a>	控制 < <a href="#">AdRotator</a>
<a href="#">HtmlGeneric</a>	控制其 <a href="#">Button</a>
<a href="#">HtmlImage</a>	控制 < <a href="#">Calendar</a>
<a href="#">HtmlInputButton</a>	控制 < <a href="#">CalendarDay</a>
<a href="#">HtmlInputCheckBox</a>	控制 < <a href="#">CheckBox</a>
<a href="#">HtmlInputFile</a>	控制 < <a href="#">CheckBoxList</a>
<a href="#">HtmlInputHidden</a>	控制 < <a href="#">DataGrid</a>
<a href="#">HtmlInputImage</a>	控制 < <a href="#">DataList</a>
<a href="#">HtmlInputRadioButton</a>	控制 < <a href="#">DropDownList</a>
<a href="#">HtmlInputText</a>	控制 < <a href="#">HyperLink</a>
<a href="#">HtmlSelect</a>	控制 < <a href="#">Image</a>
<a href="#">HtmlTable</a>	控制 < <a href="#">ImageButton</a>
<a href="#">HtmlTableCell</a>	控制 < <a href="#">Label</a>
<a href="#">HtmlTableRow</a>	控制 < <a href="#">LinkButton</a>
	<a href="#">ListBox</a>

内嵌在 html 中，弱化 http 协议，转为事件驱动

GWT、dartlang、Polymer

# GOOGLE 的方案

# GWT

- 用 java 开发 web 前端：编译为 js
- RIA：全端
- 回到桌面时代
- Google Adwards, Google Wallet
- 问题：
  - 搜索引擎不友好
  - 位置逐渐被 Dart 取代



GWT



# Dart: 替代 JS

- 语言更干净
- 自带开发工具
- 支持 polymer 组件化开发
- 一种语言，支持前后端
- 可编译为高效的 js ，支持各种浏览器
- 快速进入语言排行前 20



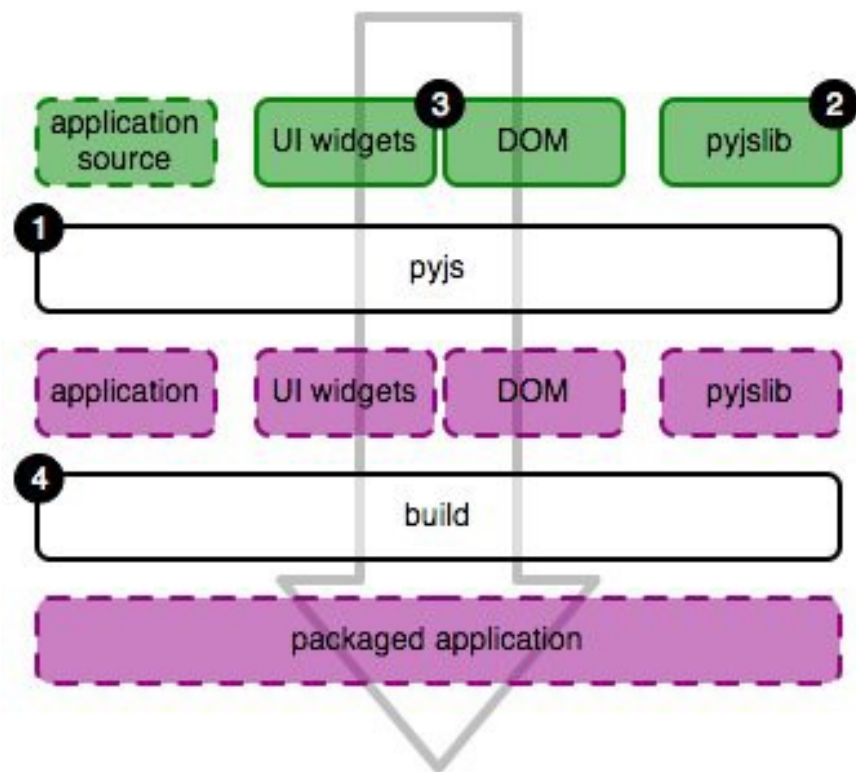
# Polymer: Web Component 实现

- W3C web 组件技术规范
  - Custom Element
  - Template
  - Shadow DOM
  - HTML import
- 实现： Polymer
- 代表 web 未来
- 是终点吗？



# 号外：Pyjamas

- Python 版本的 GWT
- 自带 UI 陈旧
- 用户不多



一个小 (DiaoSi) 团队如何不被技术浪潮摧毁

# 易度前端的演进

# 作为一个小公司

- 技术主管不大懂 javascript
- 招不起昂贵专业的前端
- 害怕技术选型错上贼船
- 历史负担：自制前端框架
- 技术更新，成本非常高
- 如何平滑支持移动端

# 作为一个开发平台

- 解放开发者
  - 专注业务需求
  - 少折腾技术
  - 技术延续性
- 目标：
  - 力求最简
  - 三脚猫功夫，也能开发
  - 快速开发
  - 屏蔽实现细节

# 尝试过 backbone

- 当年 Zope 时代的感觉
- 泥潭里挣扎
- 3 个月，完成前端框架
- 革命成本太高
- 放弃

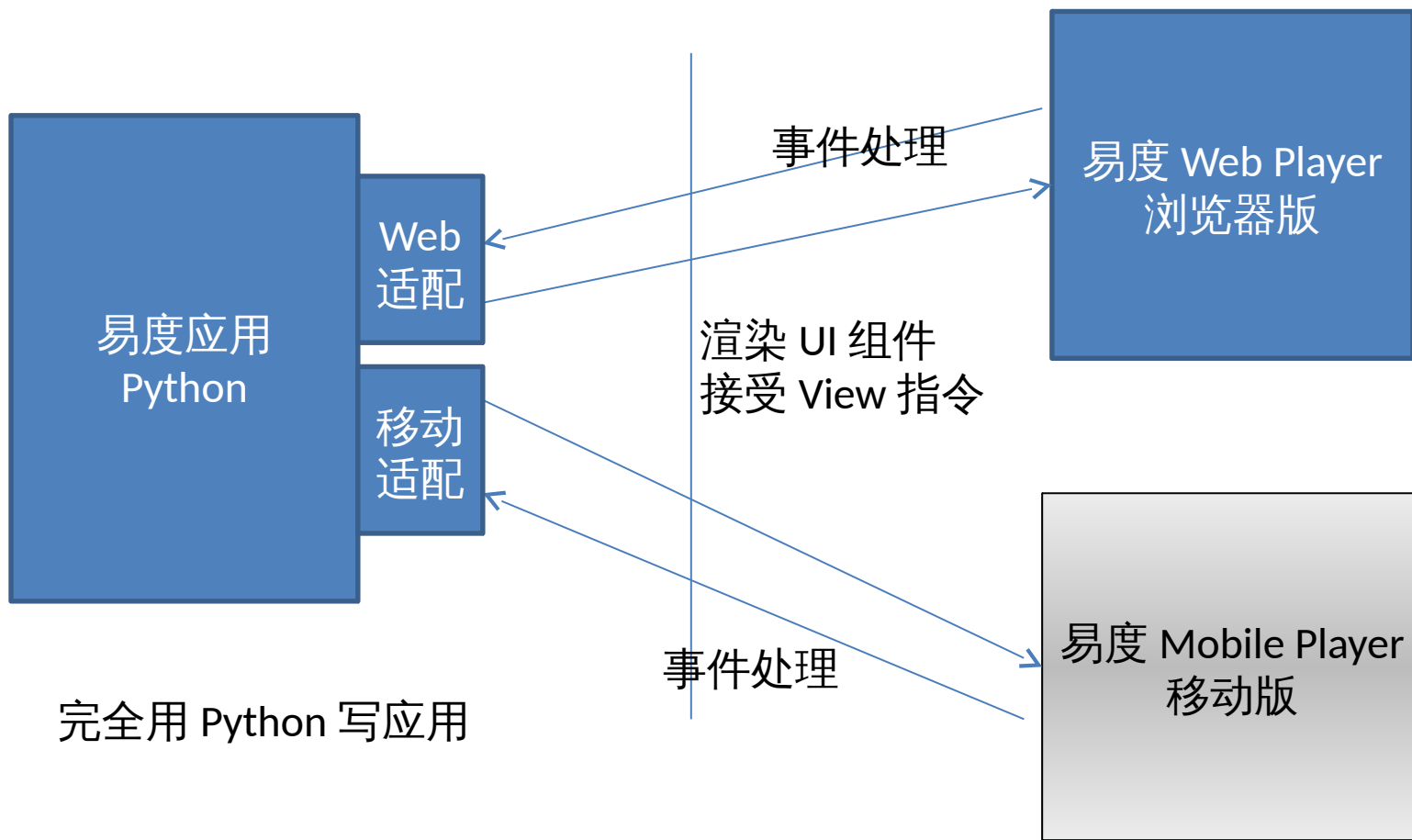
# 只能演进，而非革命

- 尽量少的人
- 最少的技能
- 快速开发
- 实用、少折腾



LIFE IS SHORT  
USE PYTHON

# 易度前端 : Fit Client



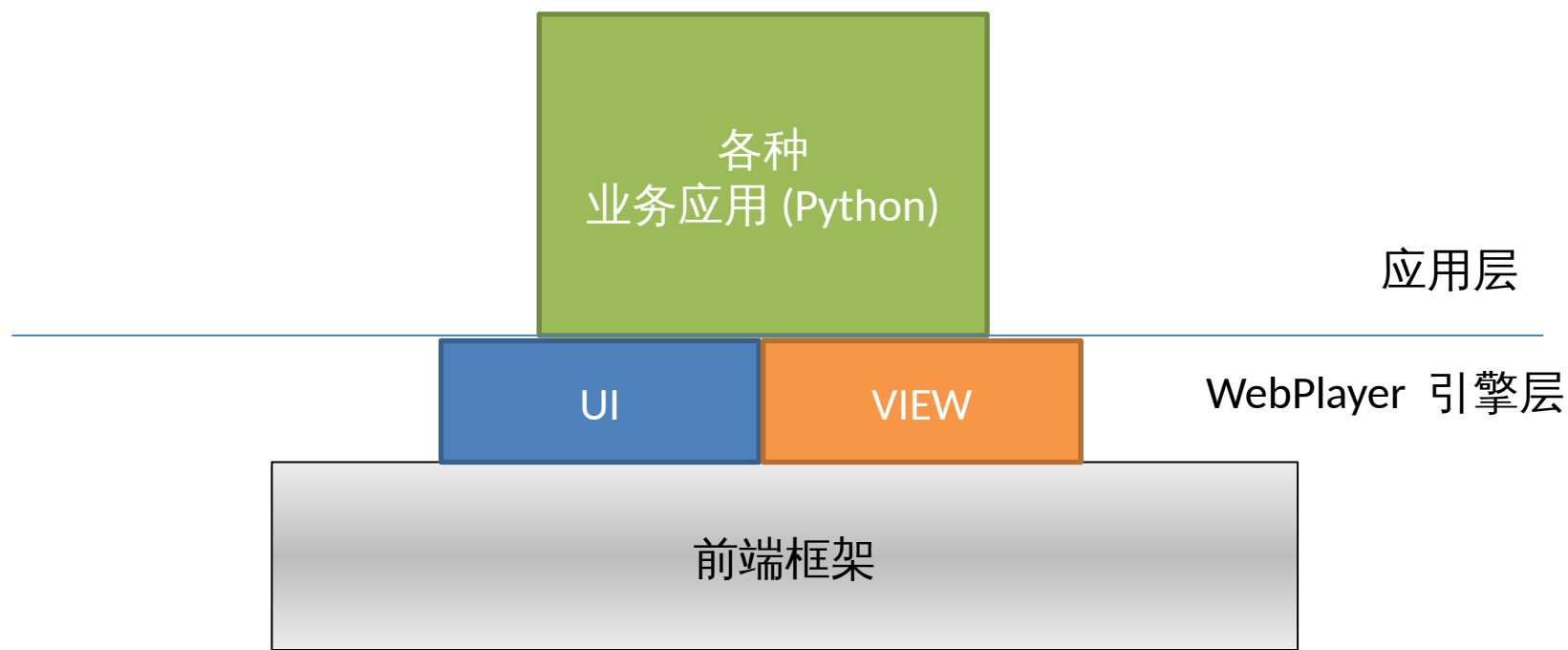
易度 Player : 不胖不瘦, 刚好

# 分层：前端引擎和应用

易度 Player（前端引擎）：20%    易度应用：80%

- javascript
- 基础
  - 整体外观风格
  - 栏目导航
  - 加载进度
  - 出错处理
- 动态
  - 组件渲染展示
  - 交互事件触发
  - 界面行为控制
- python
- 输出 UI 界面
- 交互事件处理
- 控制界面行为

# 前端引擎和应用



招一个 NB 的前端维护引擎即可  
前端技术革新，只需升级引擎层

# 使用 UI 组件

```
ui.panel()\n    .add(ui.button('Click me')\n        .add(ui.text('Hello, world')))\n+ ui.button(' 新增' )\n    .icon('add')\n    .loading(' 正在提交... ')\n    .on('click', '@zopen.test:test')
```

# UI 组件示例

## 按钮条

按钮1 按钮2

## 标签页

tab1

tab2

tab3

content1

## 导航

title1

title2

## 导航树

level1

level1

level2

level2 1

level2

level3

## 编辑器

```
1 print "H
```

## 表格

hello

World!

done

## 面板

title

content

footer

## 折叠面板

▼ title

content

## 列表组

国庆大放假，10天！潘俊勇 2004-12-1

关于报销的最新通知张三

上班不打卡，扣全月工资！牛鼻公司的NB老板

# UI 组件库

使用 Python 表单类似 Bootstrap 的组件

系统所有功能组件化

## [2 ui 元素](#)

[2.1 链接ui.link](#)

[2.2 徽章ui.badge](#)

[2.3 输入控件ui.field](#)

[2.4 一段文字ui.text](#)

[2.5 段落方式ui.paragraph](#)

[2.6 按钮ui.button](#)

[2.7 html代码ui.html](#)

## [3 UI集合](#)

[3.1 列表组ui.list\\_group](#)

[3.2 表格ui.grid](#)

[3.3 下拉菜单ui.menu](#)

[3.4 表单ui.form](#)

[3.5 按钮组ui.button\\_group](#)

[3.6 面板ui.panel](#)

[3.7 导航ui.nav](#)

[3.8 标签页ui.tabs](#)

[3.9 路径ui.breadcrumb](#)

[3.10 树ui.tree](#)

[3.11 分页条ui.pagination](#)

## [4 系统功能组件 ui.portlets](#)

[4.1 文件查看器](#)

[4.2 我的工作](#)

[4.3 流程历史](#)

[4.4 容器下的对象树](#)

[4.5 文件列表](#)

[4.6 可选视图](#)

[4.7 文件相册](#)

[4.8 表单列表](#)

[4.9 附件的展示](#)

[4.10 关注按钮](#)

[4.11 授权按钮](#)

[4.12 收藏按钮](#)

[4.13 新建流程](#)

[4.14 评注区域](#)

[4.15 标签组面板](#)

[4.16 流程历史](#)

# View 指令集

- `view.message(' 出错了', type='error', )`
- `view.modal(form, width=600)`
- `view.layout.main().set_content(form)`
- `view.layout.right().set_content(form)`





# 可视化制作表单

已有字段：

添加一个新的字段：

序号	字段名	标题	类型	描述
1	title	标题	单行文本	
2	description	描述	文本框	
3	responsibles	负责人	人员选择	该任务的负责人
4	comment	处理说明	文本框	
5	reviewer	检查人	人员选择	该任务的检查人
6	reviewer_comment	检查意见	文本框	
7	start	开始时间	日期	
8	end	期限	日期	
9	attach	附件	文件选择	

选择字段类型:

- 选择字段类型:
- 单行文本
- 多行文字
- 文本框
- 密码
- 整数
- 小数
- 逻辑字段(真/假)
- 日期
- Selection
- 标签
- 人员选择
- 流程选择
- 流程单选择
- 文件选择
- 文件夹选择
- Location Selection
- 公式字段
- Panel
- 动态表格

```
form = root.packages.get_form_obj('zopen.test:form')
ui.form(title, description, action)\
    .fields(form)\
        .button('submit', '提交')\
            .on('submit', '@zopen.test:submit_form')
```

# 组件选择 (DOM) 和操作

# 向 Web Component 致敬

```
view.closest('tabs')\  
  .active_panel()\  
  .set_content(ui.h1('hello, world'))
```

```
view.find('tree#leftnav')\  
  .expand()
```

# 用户触发事件处理

- `ui.link('click me')\n .on('click', '@zopen.test:test')`
- 所有脚本提交到服务端脚本
- 用 Python 处理

# 自定义事件：界面联动

- 触发删除事件：  
`view.trigger('dataitem-removed', \n\n {'uid':12312, 'title':'123123'})`
- 多处处理事件：  
`ui.script().on('dataitem-removed', "@zopen.test:refresh")`

# 示例：珠三角沙龙活动页面

广州十月云专题 权限 设置 未关注 + 收藏

**成员** 编辑

**活动PM**  
潘俊勇

**文案**  
还没有任何成员

**报名管理**  
还没有任何成员

**外联**  
还没有任何成员

**主持**  
还没有任何成员

群组聊天

标题:	广州十月云专题
描述:	云给开发者带来了便利,也许,还带来了许多坑,谁知道呢。孙子有云:知己知彼者百战不殆!这个金秋十月,让我们齐聚珠江边,细数各路优秀的云产品,吐吐各种坑爹的槽,更有机会与云服务创业团队架构师们 face to face 摩擦出火花!
开始日期:	2014-10-25
结束日期:	2014-10-25
地址:	

评注(0)

管理员说...

```
form_def = root.packages.get_metadata_obj('zopen.techparty:saloon')

return {'main': ui.portlets.actions_bar(context, request)\
        + ui.form(title='沙龙信息').fields(form_def, data=context.md, edit_fields=[])\
        + ui.portlets.comment(context, request),

        'right': ui.portlets.subscribe(context, request) \
        + ui.portlets.favorite(context, request, show_text=True)\
        + ui.portlets.team(context, request)

}
```

# 示例：投票应用

周五晚上羽毛球报名

+ 投票

共有10人投票

选项	百分比	投票人数	人员
去	70%	7	王扬明,莫芙蓉,王新宇,管理员,陈仕添,陈星,钟燕娟
不去	40%	4	王扬明,邱文武,黄家安,潘俊勇

评注(0)

🗨 管理员说...

```
table = ui.grid(len(context.md['options']) + 1, 4)
table.set(0, 0, '选项')\
      .set(0, 1, '百分比')\
      .set(0, 2, '投票人数')\
      .set(0, 3, '人员')

return ui.panel(ui.h1(context.md['title']),
               ui.portlets.new_dataitem(context, request, '投票')) + \
      ui.paragraph(context.md['description']) + \
      ui.paragraph('共有%d人投票' % len(context)) + \
      table + \
      ui.portlets.comment(context, request)
```

# Python 就是一种 DSL

- Python 的表达能力，不比 html 差
- Bootstrap 版本升级，重写 html 工程浩大
- Python 写界面，与具体实现隔离

```
<ul class="nav nav-tabs" role="tablist">
  ...
  <li role="presentation" class="dropdown">
    <a class="dropdown-toggle" data-toggle="dropdown" href="#">
      Dropdown <span class="caret"></span>
    </a>
    <ul class="dropdown-menu" role="menu">
      ...
    </ul>
  </li>
  ...
</ul>
```

# 回归桌面开发模式

- 一切皆 python
  - 学习更简单
  - 几乎不用懂 web
  - 全端开发
  - 更容易 code review
- 组件化开发：为 web 未来同步
- 事件驱动：屏蔽网络
- 部分的图形化开发：可以更多



# Web Player 可定制皮肤

The image displays three examples of custom web player skins for different websites, each with a unique color scheme and navigation structure.

- 易度云办公 (Everydo Office):** Features a dark red header with navigation links: 易度云办公, 我的工作台, 客户管理, 文件中心, 办公流程. It includes a search bar with "全站搜索", a notification badge with "1", and a user profile "潘俊勇". The main content area shows a breadcrumb "文件中心 > 产品文档 > 数据字典" and action buttons for "分享", "更多", "编辑", and "新建".
- 易度开发者 (Everydo Developer):** Features a teal header with navigation links: 易度开发者, 开发教程, 开发参考, API, 博客, 论坛. It includes a search bar with "全站搜索", a notification badge with "1", and links for "其他站点", "设置", and "管理员". The main content area shows a breadcrumb "开发参考" and action buttons for "分享", "更多", "编辑", and "新建".
- 珠三角技术沙龙 (Zhujiang Delta Technology Salon):** Features a dark blue header with navigation links: 珠三角技术沙龙, 我的工作台, 活动, 文档库, 会员, 投票. It includes a search bar with "全站搜索", a notification badge with "0", and links for "其他站点", "设置", and "管理员". The main content area shows a breadcrumb "我的工作台" and sub-navigation links: "待办事项", "工作历史", "我创建的", "我收藏的".



easydo. cn

产品

应用市场

购买

产品博客

# 引擎分离的额外收获

- 前端需求变得简单
- WebPlayer 引擎实现并不复杂
- 不再需要 MVC 框架
- 对前端新技术，不再焦虑
- 现有前端积累得以延续
- 未来升级更加平滑

# 开源了吗？

- 未开源
- 一种思想，很容易实现
- 目前只实现的 web 版本
  - UI  $\square$  html
  - VIEW  $\square$  转换 JS
- 想过基于 QML 的 QT 版本

DEMOS

# 谢谢大家

- 欢迎提问
- 在线互动  
[weibo.com/panjunyong](http://weibo.com/panjunyong)