

# Alice v3

为构建更好的支付宝而作

- 颗粒化
- 注重协作
- 轻量级
- 标准
- 兼容
- 快速构建



# Alice v3

为构建更好的支付宝而作

- 颗粒化
- 注重协作
- 轻量级
- 标准
- 兼容
- 快速构建



小鱼，支付宝前端开发攻城师

微博 @sofish | [HTTP://sofish.de](http://sofish.de)



SUMMARY

# 内容概要

---

## 1 从一个CSS文件开始

为什么会有现在这种模式？

## 2 构建 Alice 样式库的基础

如何去构建这样一个库，如何做才是最适合我们的？

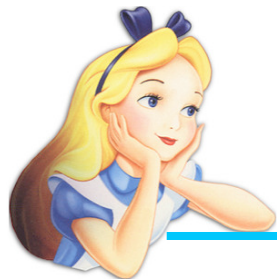
## 3 未来与设想

Alice 小组的计划进一步工作



[illegible]

# pa.css



从一个 CSS 文件开始

# 优化一下这个CSS文件吧

1. 压缩版本：94k
2. 1200+ 行(6000+)
3. 全局设置没有限制，文档不齐全
4. 所有系统样式都放在同一个文件中



pa.css





从一个 CSS 文件开始

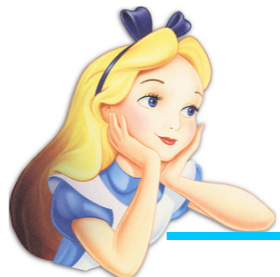
# 优化一下这个CSS文件吧

1. 压缩版本: 94k
2. 1200+ 行(6000+)
3. 全局设置没有限制, 文档不齐全
4. 所有系统样式都放在同一个文件中

无从下手



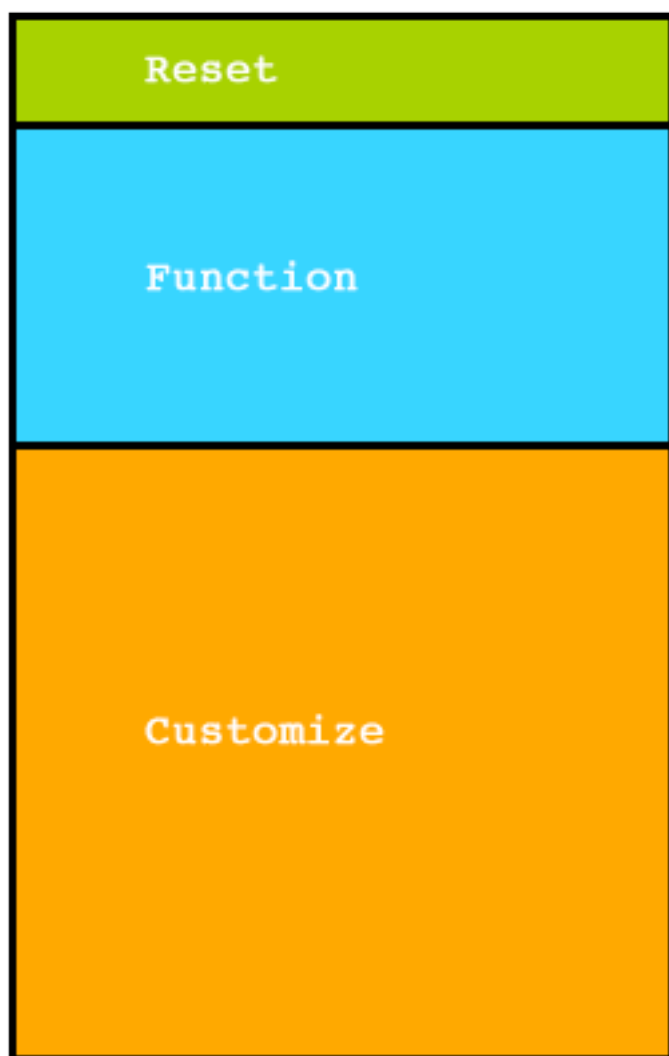
pa.css



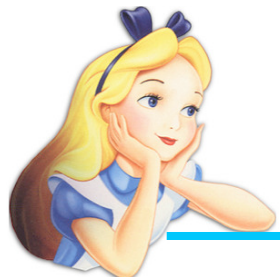
从一个 CSS 文件开始

# 这个无从下手的东东

old PA.CSS



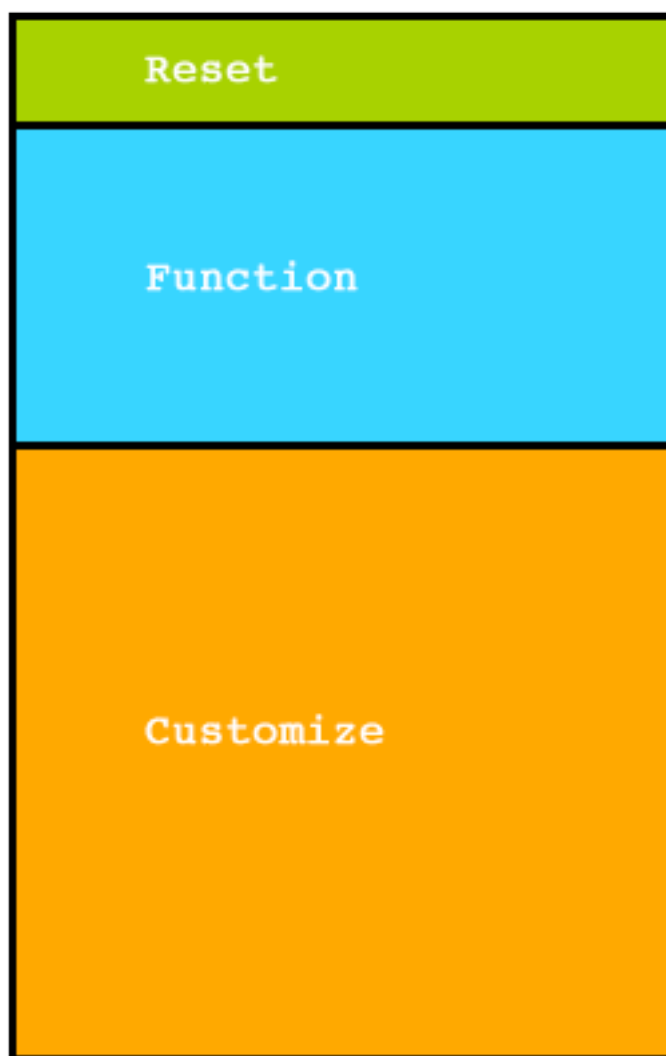




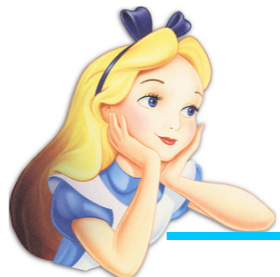
从一个 CSS 文件开始

# 这个无从下手的东东

old PA.CSS



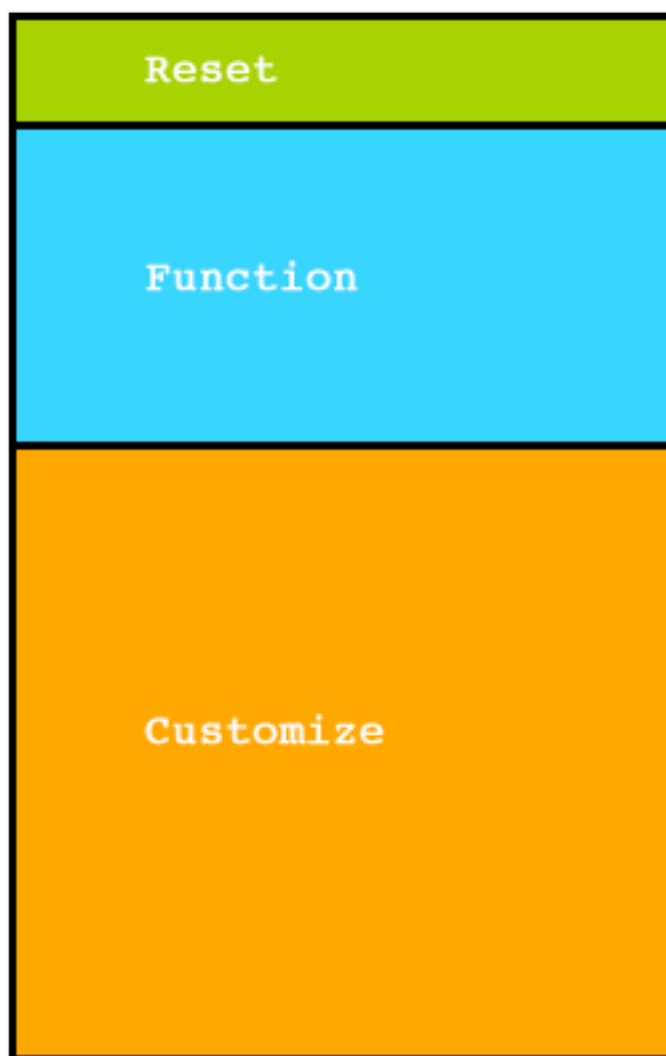
- 文件太大，我只有几个页面，不需要 100k / 6000+ 行的代码



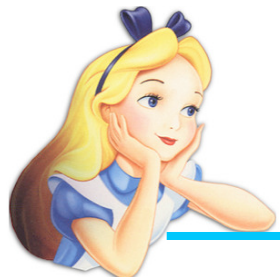
从一个 CSS 文件开始

# 这个无从下手的东东

old PA.CSS



- 文件太大，我只有几个页面，不需要 100k / 6000+ 行的代码
- 改版了，我要用什么？组件大多是个人版的，还要不要用

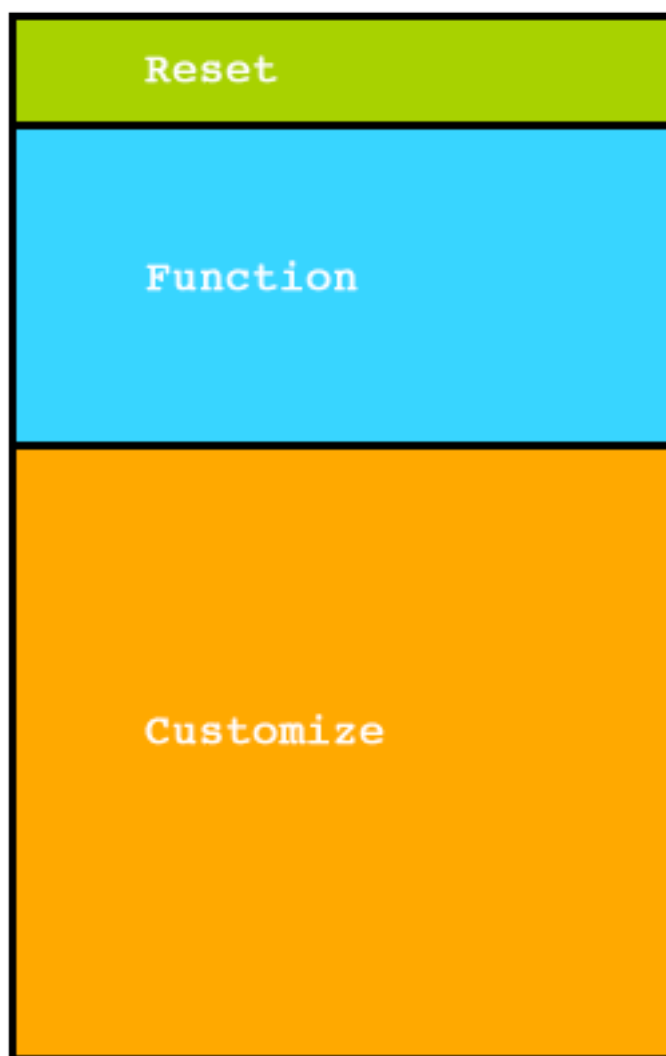


从一个 CSS 文件开始

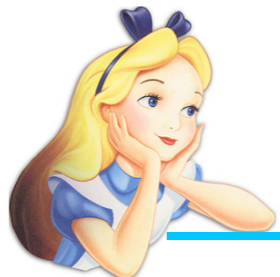
# 这个无从下手的东东

old

PA.CSS



- 文件太大，我只有几个页面，不需要 100k / 6000+ 行的代码
- 改版了，我要用什么？组件大多是个人版的，还要不要用
- 太多全局设置了，改了怕影响其他设系，不敢改啊

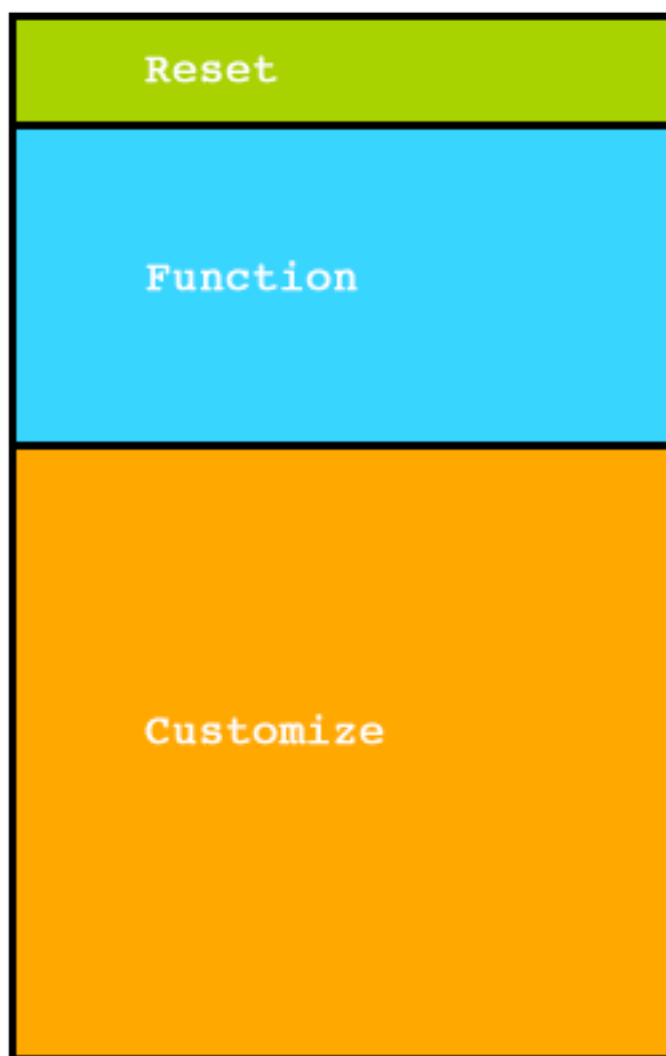


从一个 CSS 文件开始

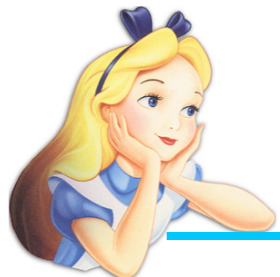
# 这个无从下手的东东

old

PA.CSS



- 文件太大，我只有几个页面，不需要 100k / 6000+ 行的代码
- 改版了，我要用什么？组件大多是个人版的，还要不要用
- 太多全局设置了，改了怕影响其他设系，不敢改啊
- 新手，不知道从哪里去入手

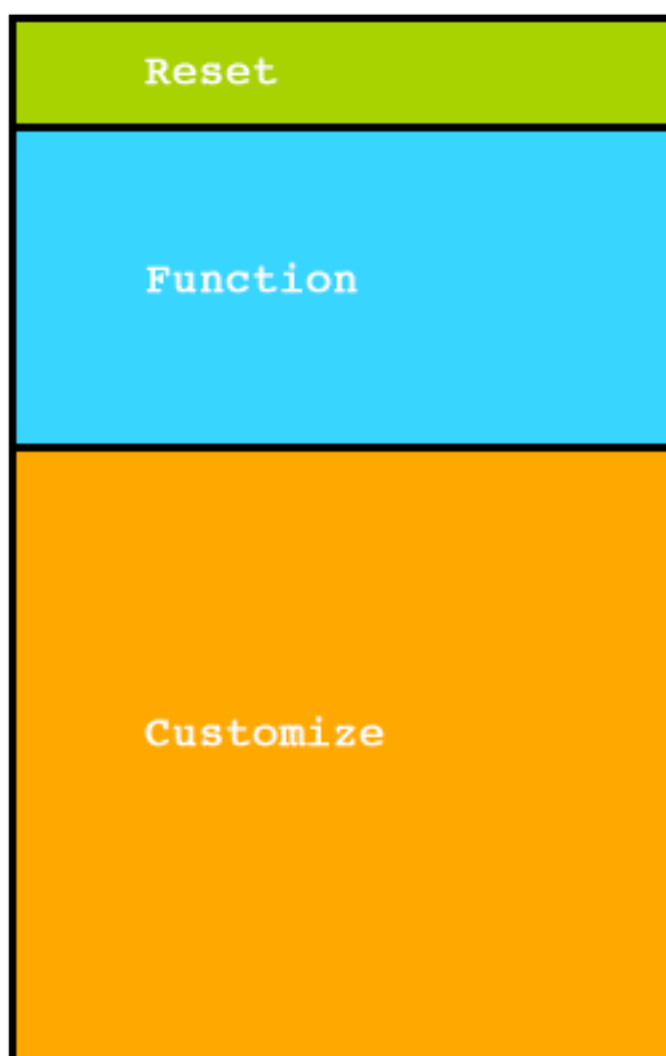


从一个 CSS 文件开始

# 这个无从下手的东东

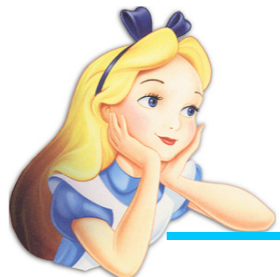
old

PA.CSS



- 文件太大，我只有几个页面，不需要 100k / 6000+ 行的代码
- 改版了，我要用什么？组件大多是个人版的，还要不要用
- 太多全局设置了，改了怕影响其他设系，不敢改啊
- 新手，不知道从哪里去入手

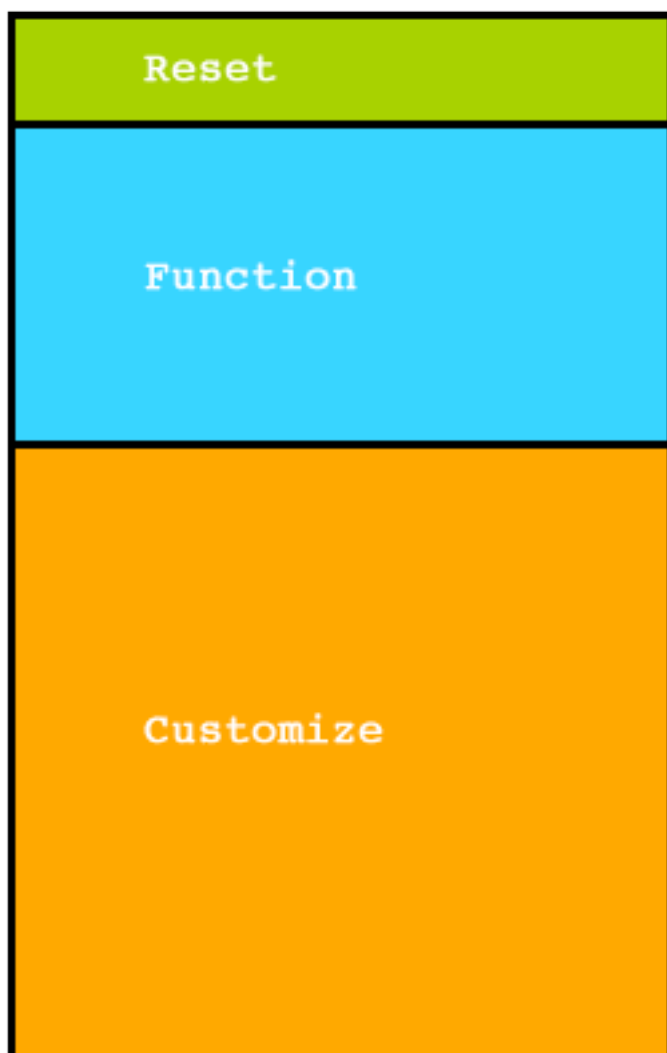
...



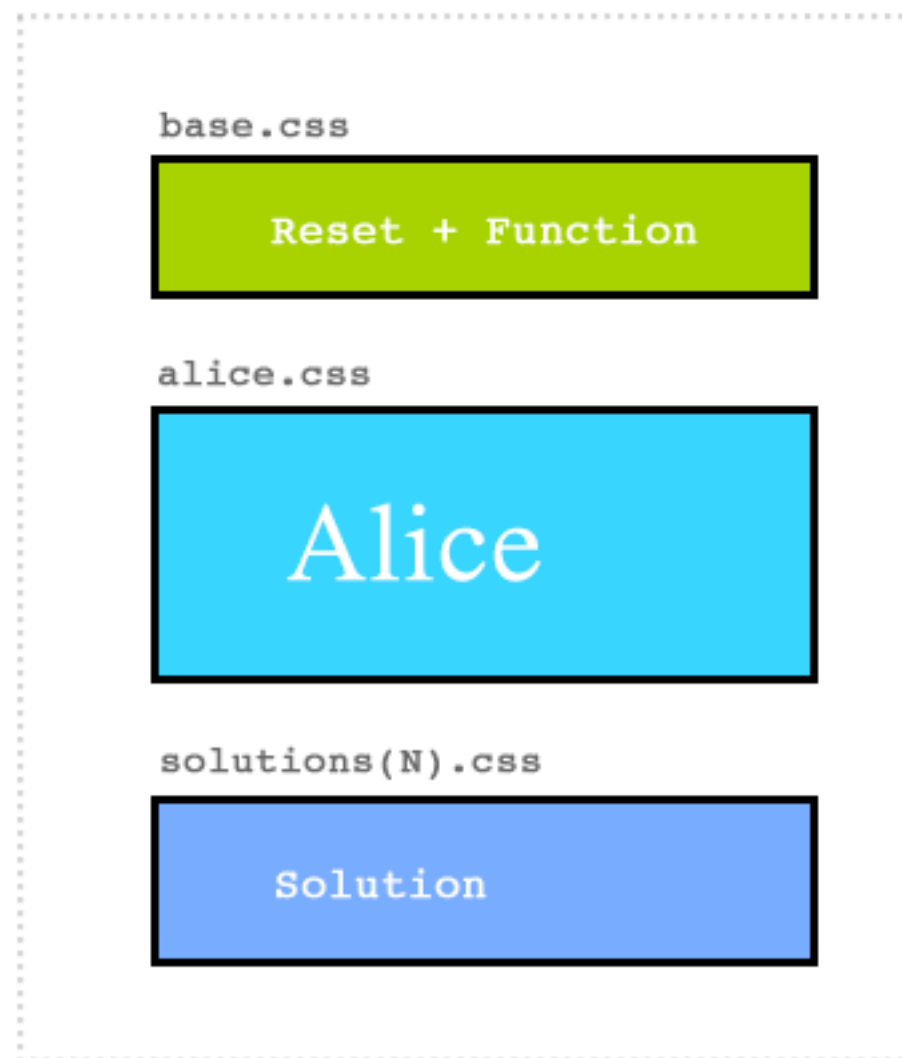
从一个 CSS 文件开始

# 设计模式

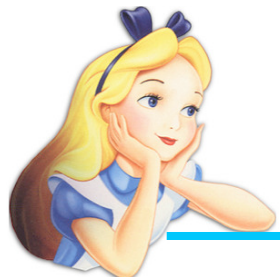
old PA.CSS



new ALICE V3





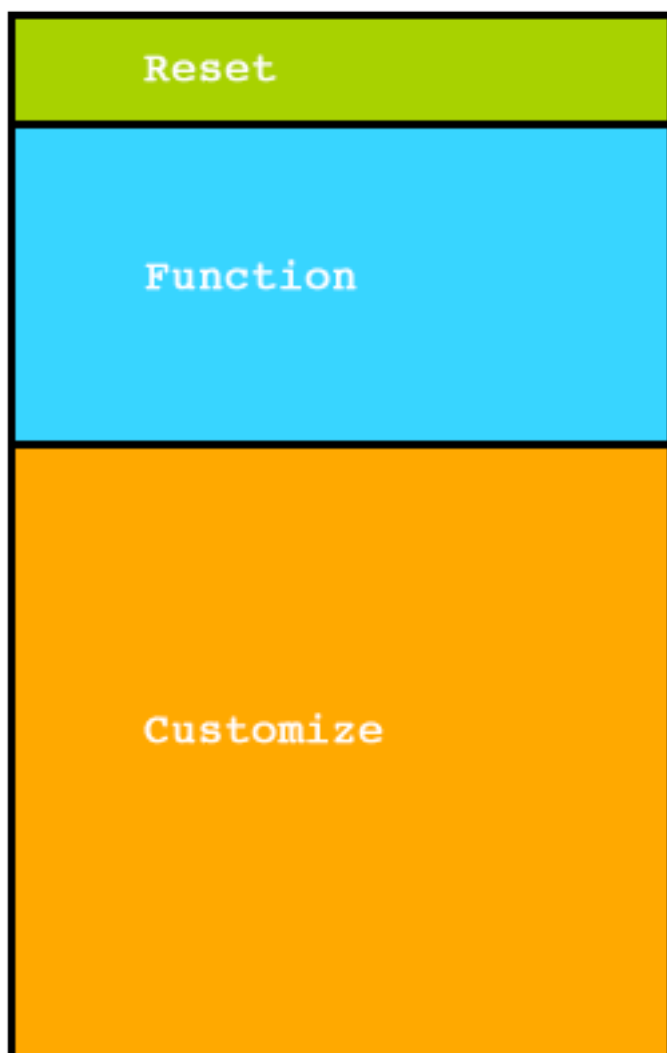


从一个 CSS 文件开始

# 设计模式

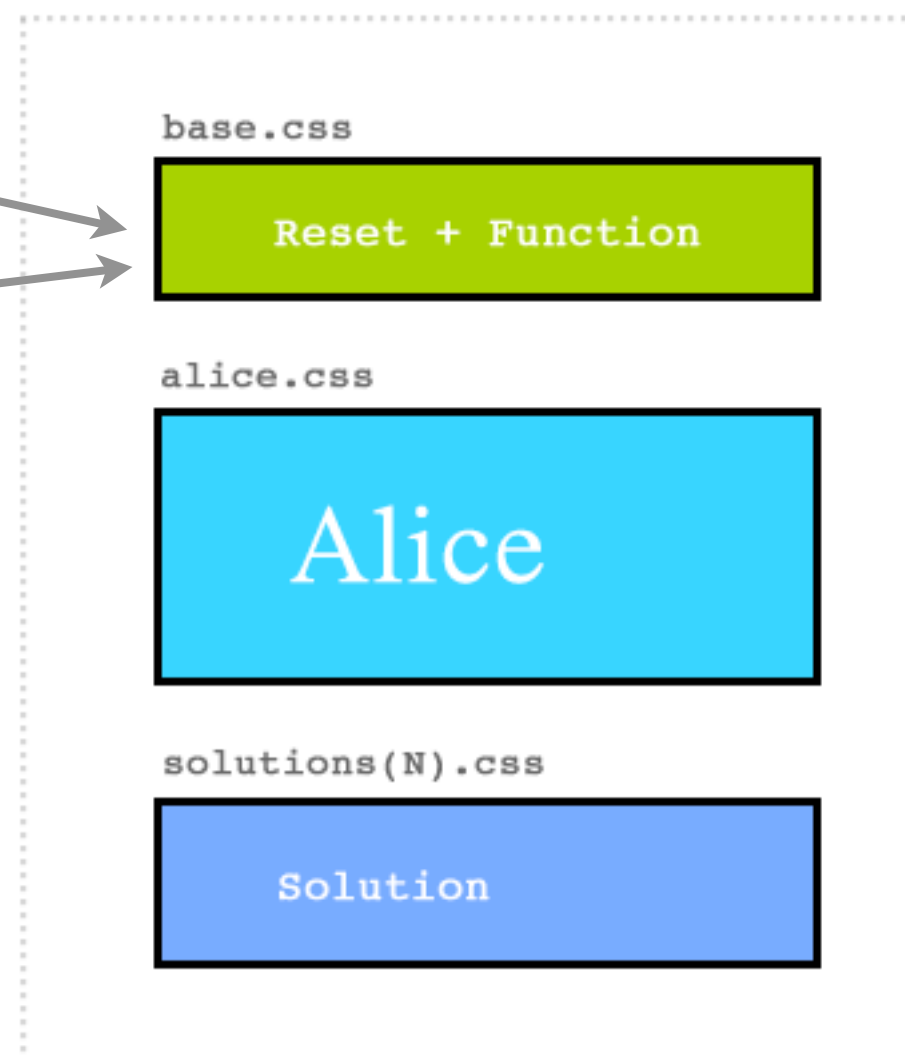
old

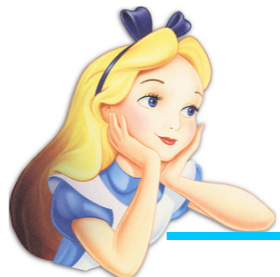
PA.CSS



new

ALICE V3



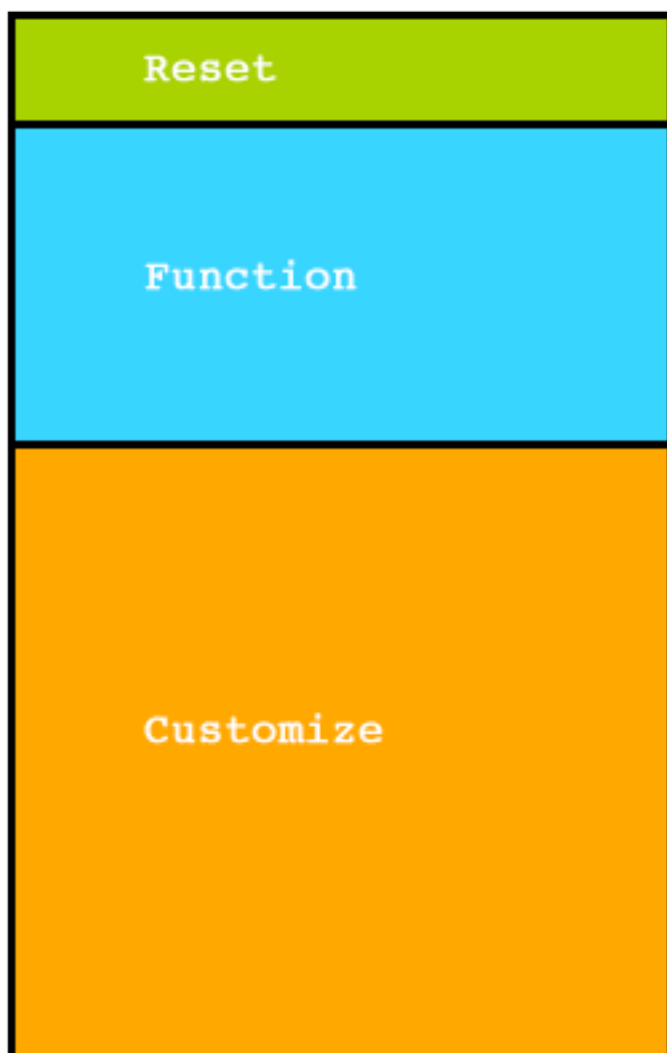


从一个 CSS 文件开始

# 设计模式

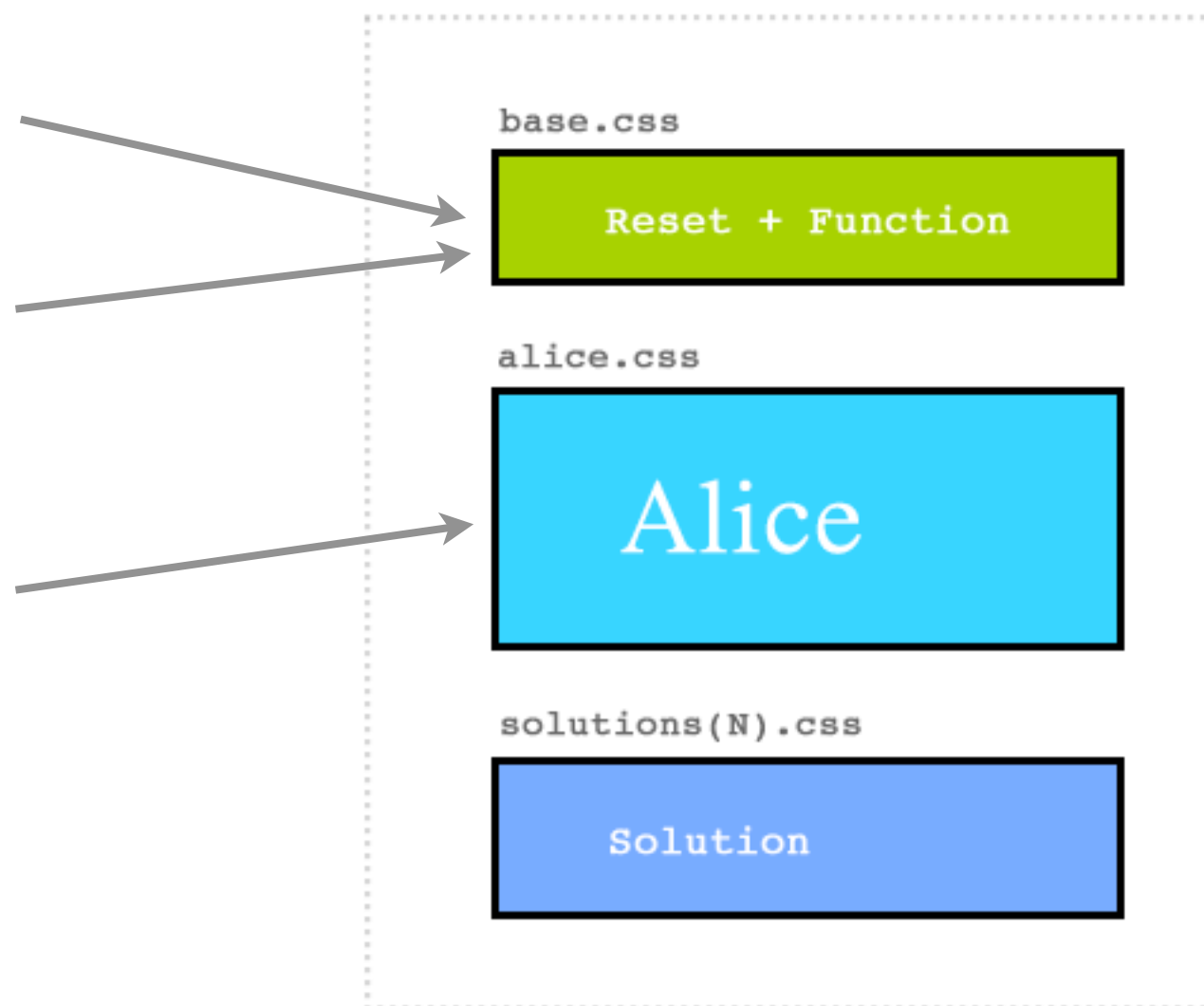
old

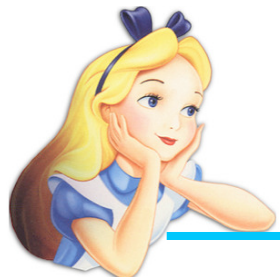
PA.CSS



new

ALICE V3

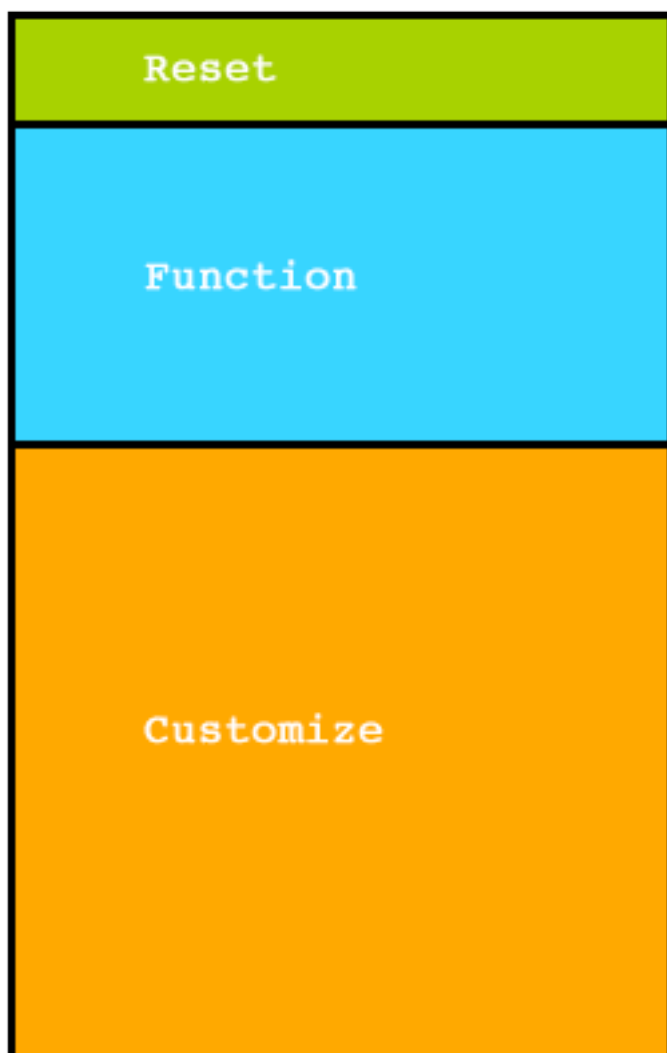




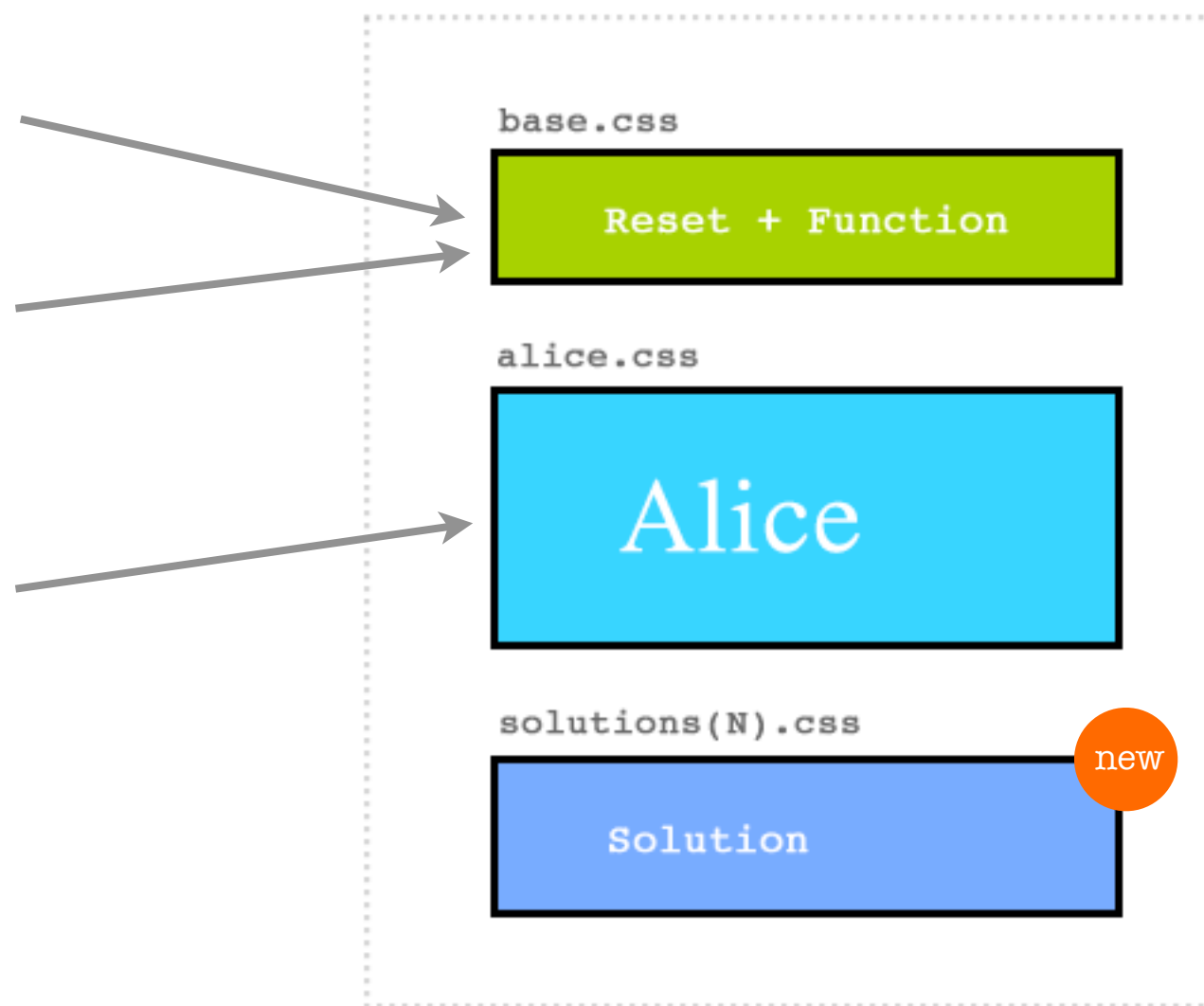
从一个 CSS 文件开始

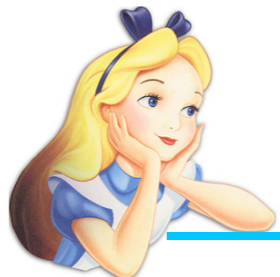
# 设计模式

old PA.CSS



new ALICE V3

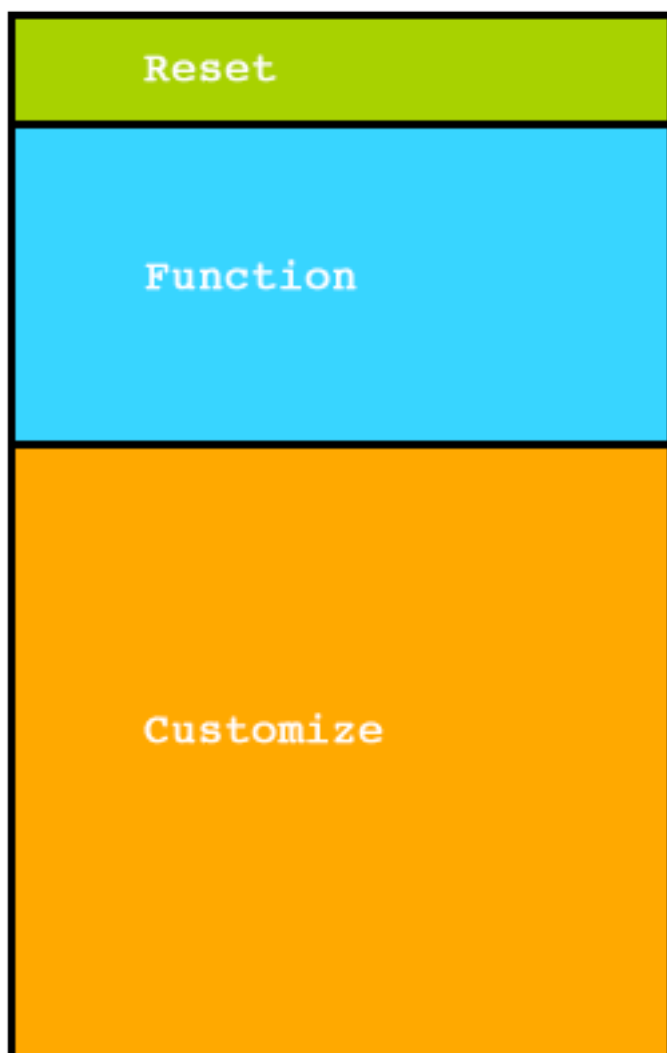




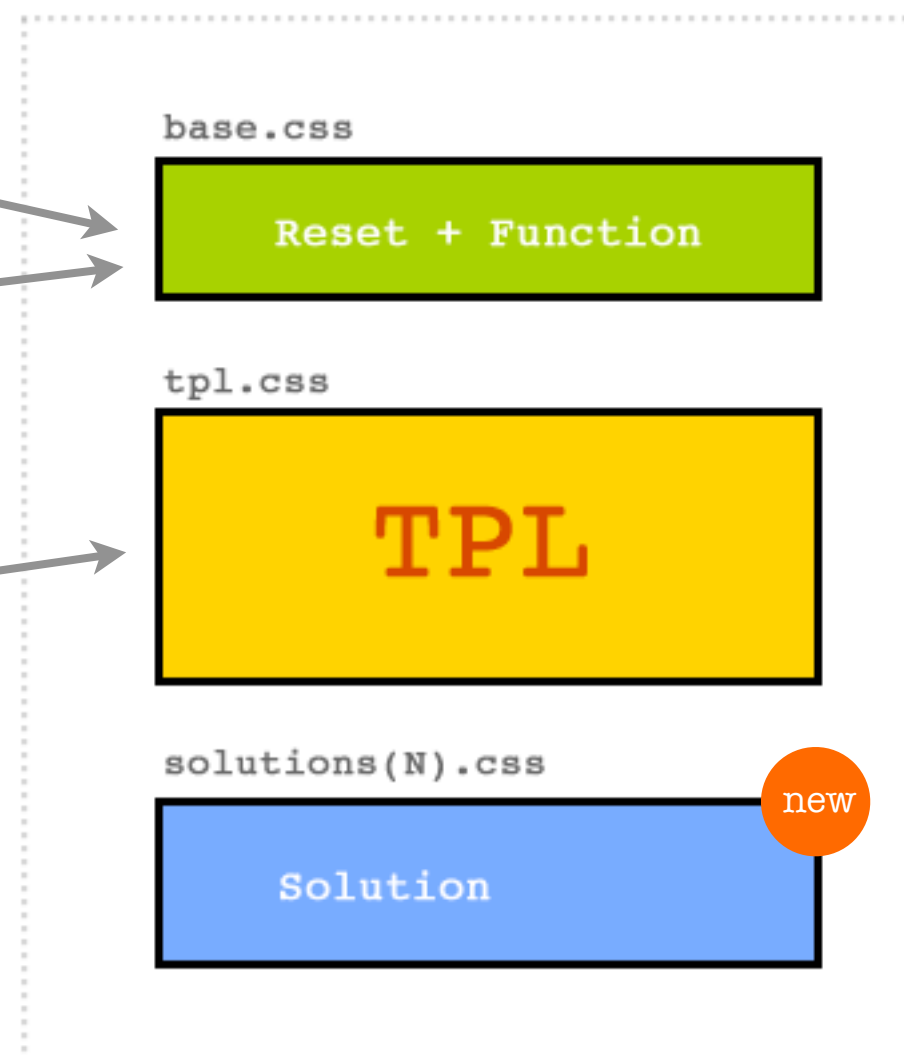
从一个 CSS 文件开始

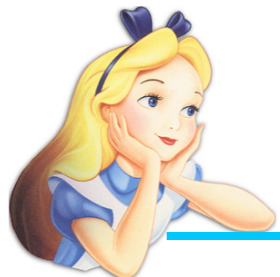
# 设计模式

old PA.CSS



new ALICE V3





从一个 CSS 文件开始

# base.css

---

## ● CSS reset

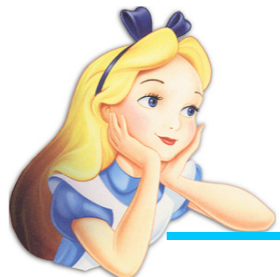
包括 HTML5 标签的 reset

## ● 小功能:

.fn-clear 清除浮动

.fn-hide/.fn-show 相当于 `display:block;/display:none;`

.fn-left/.fn-right 相当于 `float:left;/float:right;`



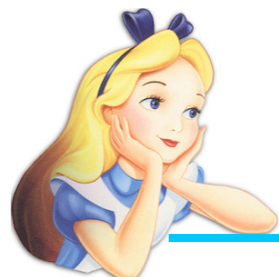
从一个 CSS 文件开始

# TPL 结构模板

```
<h3 id="link-0">一般列表(.ui-list)</h3>
<ul class="ui-list">
  <li class="ui-list-item"><a href="#">新闻第一条</a></li>
  <li class="ui-list-item"><a href="#">新闻第一条b</a></li>
  <li class="ui-list-item"><a href="#">新闻第一条c</a></li>
</ul>
```

```
/*
 * @name: 导航
 * @overview: 一般导航
 * @require: null
 */
.ui-nav{}
.ui-nav-item{}
.ui-nav-item-current{}
.ui-nav-item-child{}
.ui-nav-item-active-child{}
```

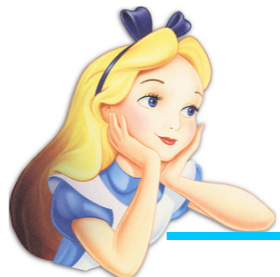




从一个 CSS 文件开始

# Solutions

 <p>radius:1px</p> <p>1像素圆角</p>	<p>上传头像:</p> <p>点击选择图片</p> <p>图片文件最大5MB。</p> <p>文件上传</p>	<p>暂无图图</p> <p>no thumb</p> <p>自适应父元素高度</p>	<p>暂无图图</p> <p>no thumb</p> <p>Copy</p>	<p>下拉选择</p> <p>下拉选择</p> <p>IE6啊真讨厌，DIV被select刺穿</p> <p>Select-Free</p>	<p>旋转木马</p> <p>旋转木马</p> <p>Rotation</p>
 <p>Layout</p>	<p>大小: 正常   中   大</p> <p>m Ipsum is simply dumn er took a galley of type ntially unchanged. It wa</p> <p>custom font-size</p>	<p>@FONT-FACE</p> <p>From the Internet Ex</p> <p>@font-face</p>	<p>DEMO: 禁用输入法</p> <p>ime-disabled</p> <p>ime-disabled</p>	<p>PNG</p> <p>24</p> <p>IE6 png 24</p>	<p>fixed</p> <p>fixed</p> <p>IE6 position-fixed</p>
<p>z-index fixed</p> <p>IE6 z-index</p>	<p>Min-Width</p> <p>Max-Height</p> <p>IE min-max</p>	<p>outline</p> <p>IE outline</p>	<p>box-shadow</p> <p>IE box-shadow</p>	 <p>IE liner-gradient</p>	<p>文字</p> <p>? px</p> <p>? px</p> <p>? px</p> <p>web font</p>
<p>中</p> <p>vertical-horizoncal</p>	<p>暂无图图</p> <p>no thumb</p> <p>nowrapURL</p>	<p>暂无图图</p> <p>no thumb</p> <p>localStorage</p>	<p>暂无图图</p> <p>no thumb</p> <p>kill iframe</p>	<p>暂无图图</p> <p>no thumb</p> <p>vertical-form-elements</p>	<p>Float:center</p> <p>Item1 Item2 Item</p> <p>float:center</p>

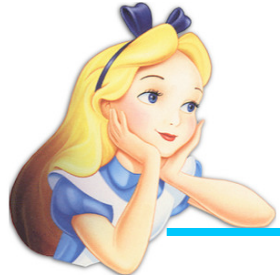


从一个 CSS 文件开始

# 解决问题

---

1. 减少依赖，仅存 `base.css`
2. 减肥：把 `solution` 分出来，使用 **TPL**
3. 颗粒化，只依赖 `base.css`，并分离各系统的依赖



从一个 CSS 文件开始

# 解决问题

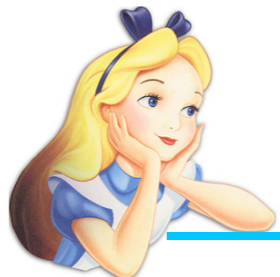
1. 减少依赖，仅存 `base.css`
2. 减肥：把 `solution` 分出来，使用 **TPL**
3. 颗粒化，只依赖 `base.css`，并分离各系统的依赖

`tpl.css`

**TPL**

`solutions(N).css`

Solution

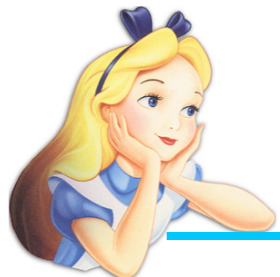


从一个 CSS 文件开始

# 解决问题

1. 减少依赖，仅存 `base.css`
2. 减肥：把 `solution` 分出来，使用 **TPL**
3. 颗粒化，只依赖 `base.css`，并分离各系统的依赖

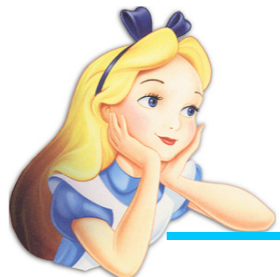
```
/*  
 * @name: ui-组件名  
 * @overview: 组件描述  
 * @required: 与其他组件的依赖关系，一般为null  
 */
```



从一个 CSS 文件开始

# 解决问题

---



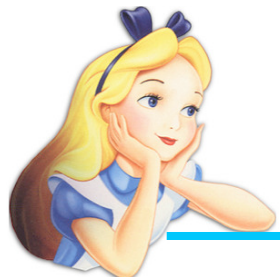
从一个 CSS 文件开始

# 解决问题

---

只有 `base.css`，  
只有 `solution` 和 `TPL`，  
没有实际的东西





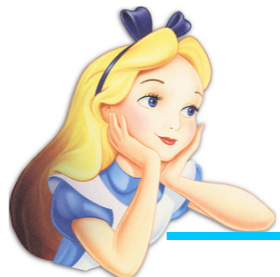
从一个 CSS 文件开始

# 解决问题

---

只有 base.css,  
只有 solution 和 TPL,  
没有实际的东西

坑爹



从一个 CSS 文件开始

# 解决问题

相同的结构 (HTML)  
构建不同的视觉效果

HTML

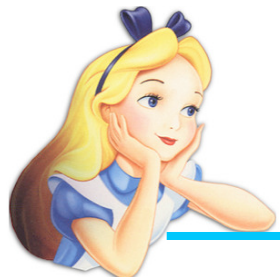


Pesonal

Merchant

Life

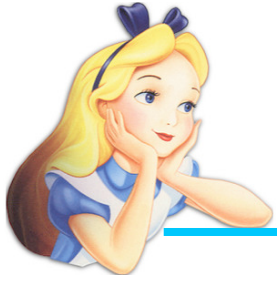
Home



从一个 CSS 文件开始

# 解决问题

---



从一个 CSS 文件开始

# 解决问题

`base.css`

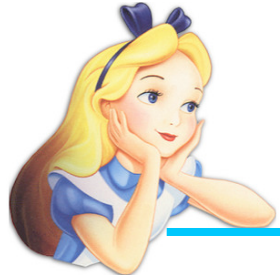
Reset + Function

`tpl.css`

TPL

`solutions(N).css`

Solution



从一个 CSS 文件开始

# 解决问题

`base.css`

Reset + Function

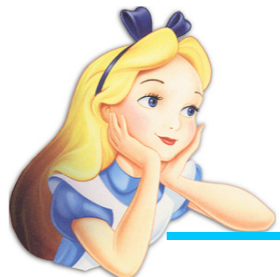
`tpl.css`

TPL

`solutions(N).css`

Solution

只有 `base.css`,  
只有 `solution` 和 TPL,  
大家的编码风格  
还是一样。



从一个 CSS 文件开始

# 解决问题

`base.css`

Reset + Function

`tpl.css`

TPL

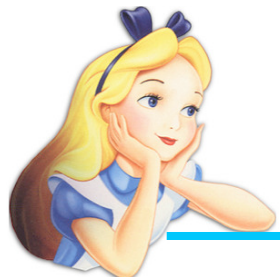
`solutions(N).css`

Solution

坑爹

只有 `base.css`,  
只有 `solution` 和 `TPL`,  
大家的编码风格  
还是一样。





从一个 CSS 文件开始

# Alice v3 与样式库

new ALICE v3

base.css

Reset + Function

tpl.css

TPL

solutions(N).css

Solution

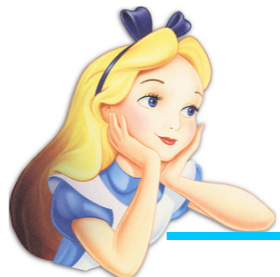
new STYLE LIB

Pesonal

Merchant

Life

Home



从一个 CSS 文件开始

# Alice v3 与样式库

new

ALICE V3

base.css

Reset + Function

tpl.css

TPL

solutions(N).css

Solution



基于规范

new

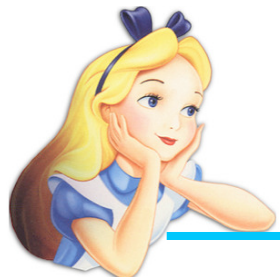
STYLE LIB

Pesonal

Merchant

Life

Home



从一个 CSS 文件开始

# Alice v3 与样式库

## Alice TPL / 模板

Alice 基于 tpl 模板，适应多变的视觉设计。

tpl.css	tpl.html
---------	----------

## Alice Gallery / 系统UI组件

用户之家 Home <sup>new</sup>	积分 Ppoint	企业版 Ea	老个人版 pa.css
个人版 Personal	生活助手 life.css		
解决方案 Solution			




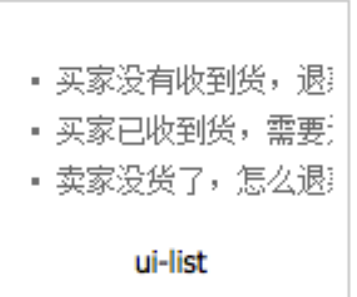
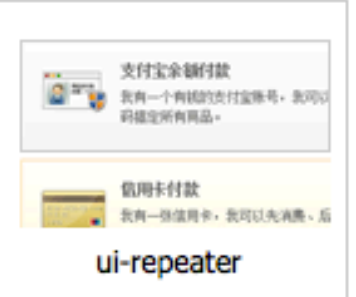


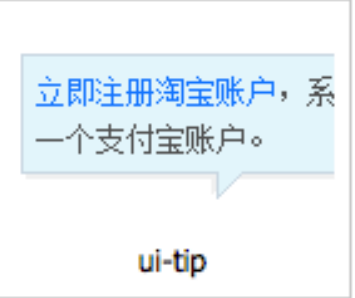





## Alice Schema / Alice 架构图

Alice Schema
--------------

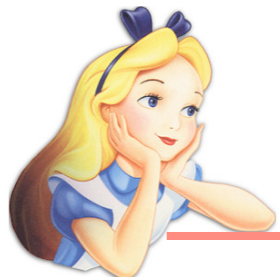


从一个 CSS 文件开始

# Alice v3 与样式库

 <p>ui-nav</p>	 <p>ui-title</p>	 <p>ui-step</p>	 <p>ui-list</p>	 <p>ui-repeater</p>	 <p>ui-dropdown</p>
 <p>ui-page</p>	 <p>ui-tip</p>	 <p>ui-message</p>	 <p>ui-form</p>	 <p>ui-input</p>	 <p>ui-button</p>
 <p>ui-tab</p>	 <p>ui-table</p>				

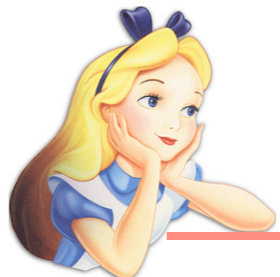




构建 Alice 样式库的基础

# Alice v3 规范

---

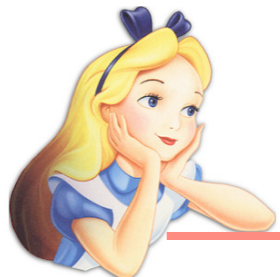


构建 Alice 样式库的基础

# Alice v3 规范

---

1. 设计模式
2. 文档结构/命名/注释
3. 组件库构建规范
4. Maven 自动部署与展示平台
5. 其他



构建 Alice 样式库的基础

# Alice v3 规范

---

1. 设计模式
2. 文档结构/命名/注释
3. 组件库构建规范
4. **Maven** 自动部署与展示平台
5. 其他

申请访问权限：

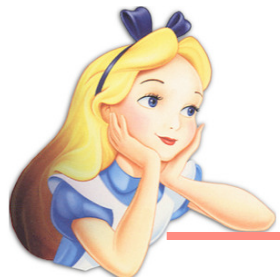
[https://docs.google.com/Doc?docid=0AWiI12yCmwaoZGNiemJqOGpfMThjanM5NWJjaA&hl=en\\_US](https://docs.google.com/Doc?docid=0AWiI12yCmwaoZGNiemJqOGpfMThjanM5NWJjaA&hl=en_US)





```
// 组件名
.ui-tab
```

```
// 组件模块
.ui-tab-container
```



构建 Alice 样式库的基础

# 组件编写：命名/模块化规范

1

// 组件名  
.ui-tab

// 组件模块  
.ui-tab-triggers

.ui-tab-trigger-cell

current

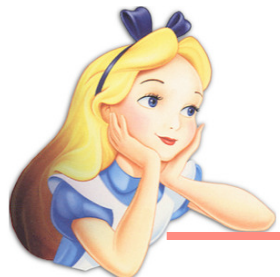
display:none;

// 组件模块内的模块  
.ui-tab-container-cell  
  
// 模块状态  
.ui-tab-container-cell-current

.ui-tab-container-cell

.ui-tab-con

// 组件模块  
.ui-tab-container



构建 Alice 样式库的基础

# 组件编写：命名/模块化规范

1

// 组件名  
.ui-tab

2

// 组件模块  
.ui-tab-triggers

.ui-tab-trigger-cell

current

display:none;

// 组件模块内的模块

.ui-tab-container-cell

// 模块状态

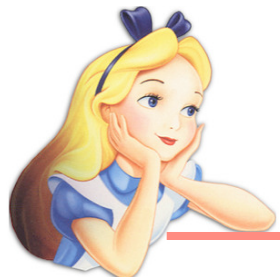
.ui-tab-container-cell-current

.ui-tab-container-cell

.ui-tab-con

2

// 组件模块  
.ui-tab-container



构建 Alice 样式库的基础

# 组件编写：命名/模块化规范

1

// 组件名  
.ui-tab

2

// 组件模块  
.ui-tab-triggers

.ui-tab-trigger-cell

current

3

3

// 组件模块内的模块

.ui-tab-container-cell

// 模块状态

.ui-tab-container-cell-current

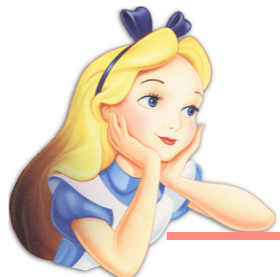
display:none;

.ui-tab-container-cell

.ui-tab-con

2

// 组件模块  
.ui-tab-container



构建 Alice 样式库的基础

# 组件编写：命名/模块化规范

首页 上一页 **1** 2 3 ... 40 41 下一页 尾页

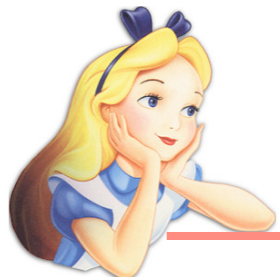
区分其他模块  
用“模块名+唯一名称”来区分

① 组件名

② 组件模块

```
<div class="ui-page">
  <span class="ui-page-item ui-page-item-info">共41条记录, 共5页</span>
  <a href="#" class="ui-page-item ui-page-item-first">首页</a>
  <a href="#" class="ui-page-item ui-page-item-prev">上一页</a>
  <a href="#" class="ui-page-item ui-page-item-current">1</a>
  <a href="#" class="ui-page-item">2</a>
  <a href="#" class="ui-page-item">3</a>
  <span class="ui-page-item">...</span>
  <a href="#" class="ui-page-item">40</a>
  <a href="#" class="ui-page-item">41</a>
  <a href="#" class="ui-page-item ui-page-next">下一页</a>
  <a href="#" class="ui-page-item ui-page-last">尾页</a>
</div> <!-- .ui-page -->
```

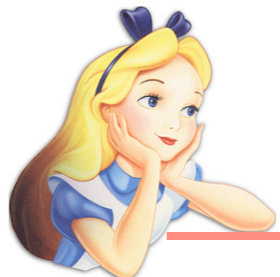
③ 模块状态



构建 Alice 样式库的基础

# 组件编写：命名/模块化规范

---



构建 Alice 样式库的基础

# 组件编写：命名/模块化规范

Normal

hover

disabled

登录

登录

登录

```

      ① 组件名
      ↑
// 正常状态
<div class="ui-button">
  <div class="ui-button-text">登陆</div>
</div>
      ② 组件模块
      ↑

      ① 组件状态
      ↑
// hover 状态
<div class="ui-button ui-button-hover">
  <div class="ui-button-text">登陆</div>
</div>

// disabled 状态
<div class="ui-button ui-button-disabled">
  <div class="ui-button-text">登陆</div>
</div>
```



Alice 小组的计划进一步工作

# 未来的工作

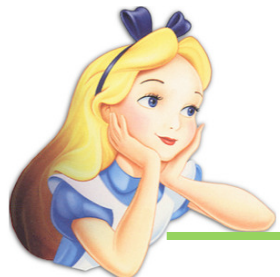
- CSS3 方面的应用和影响交互/视觉的实践研究
- 规范自动化检测工具
- 让更多人能用  
开源与自动化的结合，让运营能自己添加页面
- 增加库的应用，如强化搜索和代码实时修改

...

未来







Alice 小组的计划进一步工作

# 未来的工作

- CSS3 方面的应用和影响交互/视觉的实践研究

- 规范自动化检测工具

- 让更多人能用

开源与自动化的结合，让运营能自己添加页面

- 增加库的应用，如强化搜索和代码实时修改

...

- HTML4和5的应用

未来





*Thank You !!*

*& FA2*



keynote by sofish  
<http://sofish.de>



thx