## [腾讯前端团队规范](https://alloyteam.github.io/CodeGuide/)

### 标准文件结构

[mainfolder]

|--[component] //所有组件

|   |-- [component\_name] //组件文件夹

|   |   |-- img1.png //组件依赖的图片

|   |   |-- index.css //组件依赖的css文件

|   |   |-- index.js //组件依赖的js文件

|   |-- [component\_name] //组件文件夹

|   |   ...

|   |   |-- index.css //组件依赖的css文件

|   |   |-- index.js //组件依赖的js文件

|--[js] //js文件夹

| |-- main.js

|--[style] //所有样式相关的css和image

| |-- [image] //主要image文件夹

| | |-- img1.png

| | ...

| |-- [style\_name\_a] //皮肤A的文件夹

| | |-- [image] //皮肤A的image

| | |-- style\_name\_a.css //皮肤A的css文件

| | ...

| |-- [style\_name\_b] //皮肤B的文件夹

| |-- [style\_name\_c] //皮肤C的文件夹

| |-- main.css //主要css文件

|--[audio] //所有样式相关的audio音频文件

| |-- sound.mp3 //audio文件

|-- index.html //index文件

|-- page1.html //其他页面

...

文件名全部小写，用下划线分隔。

### 标准html5代码

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta name="author" content="Tencent.AlloyTeam.Jetyu" />

<meta name="copyright" content="Tencent.AlloyTeam" />

<meta name="keywords" content="腾讯 Alloy 团队" />

<meta name="description" content="Tencent.AlloyTeam" />

<title>Tencent AlloyTeam 标准文档</title>

<link href="./style/main.css" rel="stylesheet" type="text/css" />

</head>

<body>

<!-- 注释 -->

<h1 id="title" class="title">Tencent AlloyTeam 标准文档</h1>

<div>

<h3>Title</h3>

<p>

标准文档

</p>

</div>

<script type="text/javascript" src="./js/jx.min.js"></script>

<script type="text/javascript">

var J = new Jx();

</script>

</body>

</html>

### 命名规则

1. 项目命名全部小写，以下划线分隔
2. 目录有复数结构，要采用复数命名法。

### HTML

1. 缩进使用soft tab（4个空格）；
2. 在属性上，使用双引号，不要使用单引号；
3. 属性名全小写，用中划线做分隔符；
4. 不要在自动闭合标签结尾处使用斜线（HTML5规范指出它是可选的。）（eg:img,input）；
5. doctype按照惯例大写；
6. lang(会给语音工具和翻译工具帮助) eg:<html lang="en-us">
7. [字符编码](https://msdn.microsoft.com/en-us/library/ms533052(v=vs.85).aspx)通过通常指定为'UTF-8'; eg:<meta charset="UTF-8">
8. IE兼容模式用<meta>标签可以指定页面应该用什么版本的IE来渲染。
   1. eg:<meta http-equiv="X-UA-Compatible" content="IE=Edge">
   2. [了解更多](http://stackoverflow.com/questions/6771258/whats-the-difference-if-meta-http-equiv-x-ua-compatible-content-ie-edge-e)
9. 引入CSS，JS,不需要指明type,因为text/css和text/javascript分别是他们的默认值。
   1. <!--External CSS--> <link rel="stylesheet" href="code\_guide.css">
   2. <!--In-document CSS--> <style>…</style>
   3. <!-- External JS--> <script src="code\_guide"></script>
   4. <!—In-document JS --> <script>… </script>
10. 属性顺序
    1. class class是为了高可复用组件设计的。
    2. id 更加具体且应该尽量少使用。
    3. name
    4. data-\*
    5. src,for,type,href,value,max-length,max,min,pattern
    6. placeholder,title,alt
    7. aria-\*,role
    8. required,readonly,disabled
    9. eg:
       1. <a class="" id="" data-modal="" href="#"></a>
       2. <input class="form-control" type="text">
       3. <img src="…" alt="…">
11. boolean属性
    1. boolean属性指不需要声明取值的属性，XHTML需要每个属性声明取值，但是HTML5不需要。
    2. eg
       1. <input type="text" disabled>
       2. <input type="checkbox" value="1" checked>
       3. <selected><option value="1" selected>1</option></select>
12. JS生成标签让内容变得更难查找，更难编辑，性能更差。应该尽量避免这种情况出现。
13. 减少标签的数量
    1. 尽量避免多余的父节点；
    2. 通过迭代和重构来使HTML变得更少。
14. 实用高于完美，尽量遵循HTML标准和语义，但是不应该以浪费实用性作为代价；
15. 任何时候都要用尽量少的复杂度和尽量少的标签来解决问题。

### CSS,SCSS

1. 缩进
   1. 使用soft tab(4个空格)。
2. 分号
   1. 每个属性末尾都要加分号。
3. 空格
   1. 不需要空格的情况：
      1. 属性名后
      2. 多个规则的分隔符','前
      3. ！important '!'后
      4. 属性值中'（'后和'）'前
      5. 行末不要有多余的空格
   2. 属性值前
   3. 选择器'>','+','~'前后
   4. '{'前
   5. ！important '!'前
   6. @else前后 ？
   7. 属性值中的','后
   8. 注释'/\*'后和'\*/'前
4. 空行
   1. 文件最后保留一个空行
   2. '}'后最好有一个空行，包括scss中嵌套的规则
   3. 属性之间需要适当的空行
5. 注释统一用'/\* \*/'（scss中也不要用'//'），可位于代码行的末尾，与代码间隔一个空格。
6. 引号
   1. 最外层统一使用双引号
   2. url中的内容要用引号
   3. 属性选择器中的属性值需要引号
7. 命名
   1. 类名使用小写字母，以中划线分隔
   2. id采用驼峰式命名
   3. scss中的变量、函数、混合、placeholder采用驼峰式命名。 ？
8. 属性声明顺序

.declaration-order {

display: block;

float: right;

position: absolute;

top: 0;

right: 0;

bottom: 0;

left: 0;

z-index: 100;

border: 1px solid #e5e5e5;

border-radius: 3px;

width: 100px;

height: 100px;

font: normal 13px "Helvetica Neue", sans-serif;

line-height: 1.5;

text-align: center;

color: #333;

background-color: #f5f5f5;

opacity: 1;

}

1. 颜色
   1. 颜色十六进制用小写字母；
   2. 颜色16进制尽量用简写
2. 属性简写
   1. 属性简写需要你非常清楚属性值的正确顺序
      1. font:-style,-variant,-weight,-size,line-height,-family
      2. background:-color,-image,-repeat,-attachment,-position
      3. transition:-property,-duration,-timing-function,-delay
      4. animation:-name,-duration,-timing-function,-delay,-iteration-count,-direction,-fill-mode,-play-state.
   2. 建议尽量分开声明会更加清晰
   3. margin和padding尽量使用简写。
3. 媒体查询 ？
   1. 尽量把媒体查询的规则靠近他们相关的规则
4. SCSS相关 ？
   1. 提交的代码中不要有@debug
   2. 声明顺序：
      1. @extend
      2. 不包含@content的@include
      3. 包含@content的@include
      4. 自身属性
      5. 嵌套规则
   3. @import引入的文件不需要开头的'\_'和结尾的'.scss';
   4. @extend中使用placeholder选择器;去掉不必要的父级引用符号'&'
5. 杂项
   1. 不允许有空的规则
   2. 元素选择器用小写字母
   3. 去掉小数点前面的0;
   4. 去掉数字中不必要的小数点和末尾的0;
   5. 属性值'0'后面不要加单位;
   6. 同个属性不同前缀的写法需要在垂直方向保持对齐
   7. 无前缀的标准属性应该写在有前缀的属性后面
   8. 不要在同个规则里面出现重复的属性
   9. 如果重复的属性是连续的则没关系
   10. 不要在一个文件里面出现两个相同的规则
   11. 选择器不要超过4层（scss中如果超过4层应该考虑用嵌套的方式来写）
   12. 发布的代码中不要有 @import
   13. 尽量少用\*选择器。

### JavaScript

1. 缩进 ：使用soft tab（4个空格）
2. 单行超过80，如果编辑器开启word wrap可以不考虑单行长度
3. 分号
   1. 以下几种情况后需要加分号
      1. 变量声明
      2. 表达式
      3. return
      4. throw
      5. break
      6. continue
      7. do-while
4. 空格
   1. 不需要空格的情况：
      1. 对象的属性名后
      2. 前缀一元运算符后
      3. 后缀一元运算符前
      4. 无论函数声明还是函数表达式'('前不要空格
      5. 数组'['后和']'前
      6. 对象的'{'和'}'前
      7. 运算符'('和')'前
   2. 需要空格：
      1. 二员运算符前后
      2. 三元运算符'?:'前后
      3. 代码块'{'前
      4. 下列关键字前：else, while, cathc, finally
      5. 下列关键字后：if, else, for, while, do, switch, case, try, catch, finally, with, return, typeof
      6. 单行注释'//'后（若单行注释和代码同行，则'//'前也需要），多行注释'\*'后
      7. 对象的属性值前
      8. for循环，分号后留一个空格，前置条件如果有多个，逗号后留一个空格。
      9. 无论是函数声明还是函数表达式，'{'前一要有空格
      10. 函数的参数之间
5. 空行
   1. 变量声明后（当变量在代码块的最后一行时，则无需空行）
   2. 注释前（当注释在代码块的第一行时，则无需空行）
   3. 代码块后（在函数调用、数组、对象中则无需空行）
   4. 文件保留最后一个空行
6. 换行
   1. 换行的地方，行末必须要有','或者运算符。
   2. 以下几种情况不需要换行；
   3. 下列关键字后：else，catch，finally
   4. 代码块'{'后和'}'前
   5. 变量赋值后
7. 单行注释
   1. 双斜线后，必须跟一个空格；
   2. 缩进与下一行代码保持一致；
   3. 可位与一个代码行的末尾，与代码间隔一个空格。
8. 多行注释
   1. 最少三行，'\*后面跟一个空格，具体参照右边的写法；
   2. 建议使用的情况：
      1. 难遇理解的代码段
      2. 可能存在错误的代码段
      3. 浏览器特殊额HACK代码
      4. 业务逻辑相关的代码
9. 文档注释
   1. 各类标签@param,@method等
   2. 请参考usejsdoc和JSDoc Guide;
   3. 建议在以下情况使用：
      1. 所有常量
      2. 所有函数
      3. 所有类
10. 引号
    1. 最外层统一使用单引号
11. 变量命名
    1. 标准变量采用驼峰式命名（除了对象的属性外，主要考虑的到cgj返回的数据）
    2. 'ID'在变量名中全大写
    3. 'URL'在变量名中全大写
    4. 'Android'在变量名中大写第一个字母
    5. 'iOS'在变量名中小写第一个，大写后两个字母
    6. 常量全大写，用下划线连接
    7. 构造函数，大写第一个字母
    8. jQuery对象必须以'$'开头命名
12. 变量声明
    1. 一个函数作用域中的所有的变量声明尽量提到函数首部，用一个var声明，不允许出现连续的var声明。
13. 函数
    1. 无论函数声明还是函数表达式，'('前不要有空格，但'{'一定要有空格；
    2. 函数调用括号前不需要空格；
    3. 立即执行函数外必须包一层括号。
    4. 不要给inline function命名。
    5. 参数之间用','风格，注意逗号后有一个空格。
14. 数组、对象
    1. 对象属性不需要加引号；
    2. 对象以缩进的形式书写，不要写在一行；
    3. 数组、对象最后不要有逗号。
15. 括号
    1. 下列关键字后必须有大括号（即使代码块的内容只有一行）：if,else,for,while,do,switch,try,catch,finally,with。
16. null
    1. 适用场景
       1. 初始化一个将来可能被赋值的对象的变量
       2. 与已经初始化的变量做比较
       3. 作为一个参数为对象的函数的调用传参
       4. 最为一个返回对象的函数的返回值
    2. 不适用场景
       1. 不要用null来判断函数调用时有无传参。
       2. 不要与未初始化的变量做比较
17. undefined
    1. 永远不要直接使用undefined进行变量判断。
    2. 使用typeof和字符串'undefined'对变量进行判断。
18. jshint
    1. 用'===','!=='代替'==','!=';
    2. for-in里一定要有hasOwnProperty的判断
    3. 不要在内置对象的原型上添加方法，如Array,Date;
    4. 不要在内层的用域的代码里声明了变量，之后却访问到了外层作用域的同名变量；
    5. 变量不要先使用后声明；
    6. 不要在一句代码中单单使用构造函数，记得将其赋值给某个变量；
    7. 不要在同个作用域下声明同名变量；
    8. 不要在一些不需要的地方加括号，例：delete(a,b);
    9. 不要使用未声明的变量（全局变量需要加到，jshintrc文件的globals属性里面）；
    10. 不要声明了变量确不使用；
    11. 不要在应该做比较的地方做赋值；
    12. debugger不要出现在提交的代码里；
    13. 数组中不要存在空元素；
    14. 不要在循环内部声明构造函数，例：new function() {……},new Object;
19. 杂项
    1. 不要混用tab和space。
    2. 不要在移出使用多个tab或space。
    3. 换行符统一用'LF'。
    4. 对上下文this的引用只能使用'\_this','that','self'其中一个来命名。
    5. 行尾不要有空白字符。
    6. switch的falling through和no。
    7. default的情况一定要有注释特别说明。
    8. 不允许有空的代码行。