

# BAwiki 程序/文件规范 (v1.1)

---

## 概括

- 本wiki是一个基于mediawiki引擎的静态网页，程序由html/css/js构建而成。由于此网页的建设只能操作前端，并不能在服务器端做任何安装和修改或者使用package manager一类的功能，再加上一些后面会展开的技术问题，此网站的搭建基本禁止导入现成的模组，要实现的功能要从零编写。
- Bwiki的后台比较繁乱，不能有效地对文件进行管理。再加上css/js的部分bwiki的封装不是很到位，所以此网站上所有的页面、页面内的css class和上传的文件都必须遵循严格的命名规范，并且必须使用驼峰法则用英语命名。

## 页面命名

- Bwiki的页面分为普通页面、模板、MediaWiki、特殊四类，其他的命名空间目前不考虑使用。页面命名必须使用英语的原因是为了保证代码的可维护性，中文字符将导致网页链接出现无法读懂的%字符串。常用的含中文字符的网页链接中，大部分都有指向性相同的英语链接。
- 目前bwiki会自动将具体页面的第一个字母大写，并将只有首字母大小写不同的页面视作同一个网址，比如.../ba/Aru 和 .../ba/aru会跳转至同一个页面，但是其他位置的字母全都区分大小写。因此保险起见所有的页面首字母大写，其余部分遵循驼峰法则。(.../ba/StudentList)
- 普通页面
  - 用户访问的页面均为普通页面，主要包括首页和wiki内的各个内容页。首页目前没有找到mediawiki原有的英语链接，不过此跳转应该只会在导航类模块中出现。内容页遵循具体类别的命名规范。
    - 学生档案：使用罗马字名称加上活动名称后缀，保证词条名称可以对应至数据库内角色。
    - 物品：
    - Sandbox前缀的页面为内容测试页面。
- 模板
  - 这类页面的链接是.../ba/Template:XXX。这类页面的命名必须清晰的说明模板的作用，如果主要功能是在另一个模板中被调用，也一定要在名称中体现。比如学生档案的模板为.../ba/Template:StudentProfile，此模板调用的组件是.../ba/Template:StudentProfileBasicInfo，.../ba/Template:StudentProfileSkillEx等。
    - 建设中如果遇到了功能非常接近的模板请考虑优化成同一个模板，不方便的情况下命名要能体现调用情况的区别。
    - 在对已经投入使用的模块进行修改的时候，请新创建模板进行测试。新模板的名称要加上创建者的字母代号后缀，以便区分和页面整理。
- MediaWiki
  - 这类页面的链接是.../ba/MediaWiki:XXX。
    - .css/.js：这两种页面一定在某个模板中被调用，因此程序名称必须和模板名称完全一致。
    - .json：虽然每个json文件通常只会在需要的js程序中被导入，但是为了保证数据易于整理，json文件都按照数据进行命名。
  - 其他的命名空间基本上没有使用的需求，因此统一不使用。

- 特殊页面
  - 这里列举一些常用的特殊页面
    - .../ba/特殊:文件列表
    - .../ba/特殊:所有页面
    - .../ba/特殊:特殊页面

## 文件命名&管理

- 此版块包含.png/.mp3文件的命名规范，所有的文件无论是上传至bwiki文件列表、github CDN还是备份都必须严格遵守规范。文件名称错误不仅会导致部分网页无法正常运行，也会给文件管理工作造成巨大麻烦。所有的文件都需要进行定期的备份和整理。
- .mp3目前会涉及bgm和语音文件。由于我们无法自己上传mp3文件，所有的音频文件都存储在一个Github repo内。CDN链接使用的是Github Pages功能，不会被屏蔽并且已经投入使用。此Github Repo也会对数据和程序进行定期备份。
  - BGM：目前路径为.../audio/bgm/XXX.mp3，使用驼峰法则首字母大写，文件名中不加空格。
  - 语音
- .png文件是bwiki文件列表内的主要内容。图片在bwiki中的实际路径是包含patchwiki的链接，链接后半部分是文件名生成的哈希码。由于常用的哈希算法经过测试都无法生成bwiki所用的链接，图片路径需要使用爬虫进行抓取保存。图片文件名称更改也会导致链接变化。因此强烈建议在需要使用的时候再上传图片，并确认文件名是否正确。此外所有图片使用驼峰法则，不使用任何空格/下划线/其他标点。
- 图片内容种类较多而且会有重叠的类型，因此在这里罗列每套图片的命名规范。在这里也会给出Github repo中文件的存储位置，作为一种整理的思路。
  - 学生
    - 学生名字使用罗马字+活动名称，保证全名和图鉴页以及数据库一致。特别注意活动名称只有第一个字母大写。
    - 立绘：.../students/portrait/XXXPortrait.png 如果立绘有差分（如阿里乌斯小队的面罩），则添加.../students/portrait/XXXPortraitSecondary.png。主立绘为游戏内默认立绘，要注意有面罩的角色默认状态不一定是配戴面罩。
    - 回忆大厅：.../students/portrait/XXXLobby.png 保存时注意文件像素大小，由于截图的时候不能保证大小一致，所以文件大小的控制在上传之前进行。图片最终比例设为4:3。
    - 图鉴：.../students/portrait/XXXCollection.png 长方形头像图，用于学生图鉴页的头像显示。
    - MomoTalk：.../students/portrait/XXXIcon.png 正方形头像图，预计用于MomoTalk等头像上不显示额外信息的位置。
    - 武器：.../students/weapons/XXXWeapon.png 武器图片，大部分情况下活动角色使用的武器和平时的武器一致，保存一个没有活动名的角色武器即可。如有不同需要按照角色名和活动名命名武器。
  - 装备图纸
    - 统一命名格式：.../items/equipment/EquipmentXXXTX.png 前面位置为英语装备类型，后面为等级数字。
    - 对应关系：
      - 帽子：Hat，手套：Glove，鞋子：Shoes
      - 发夹：Hairpin，包：Bag，徽章：Badge
      - 手表：Watch，项链：Necklace，护身符：Charm

- 技能素材

- 稀有度等级用颜色区分，分别为Grey、Blue、Gold、Purple，以下文件名中用Color代替。
- 学院名称如下：Abydos、Arius、Gehenna、Hyakkiyako、Millennium、RedWinter、Shanhaijing、Trinity、Valkyrie。以下文件名中用School代替，目前没有SRT素材。
- 技能书：.../items/skills/NotesSchoolColor.png
- BD：.../items/skills/BdSchoolColor.png
- 升级技能用材料：.../items/skills/XXXColor.png 由于不同稀有度的素材中文名字可能会有较大差别，所以素材种类不定义中英对照关系，按照英语种类名整理即可。具体可查询数据库。
- 奥秘之书：.../items/skills/SecretTechNotes.png

- 技能图标

- 由于很多技能的图标通用于ex和不是ex的技能里，以及命名过于复杂，所有的技能图标都用数字进行编号。这个编号对应的是图标中心的白色图案，文件名中还涉及了是否为ex技能以及攻击类型。攻击类型表示目前定为Red、Yellow、Blue，未来角色有新的攻击类型的时候再进行添加。
- 目前技能图标使用的是白色技能图案和背景图重叠生成，由于素材质量不完全统一，会有图案不居中的情况。优化的时候直接覆盖原图即可。
- 技能图标名称示例：
  - .../icons/skills/skill34Red.png
  - .../icons/skills/skillEx49Yellow.png
- 技能图标只上传需要的背景和图案组合。可以考虑备份无背景的技能图案。

- 漫画

- 漫画目前分为日服和国际服两套，预计未来会出现国服漫画。国际服使用In，日服使用Jp，国服待定。以下路径中用region代替
- 主漫画：.../Region/RegionComicX.png 这是漫画的主图
- 副图和彩蛋：在原先的章节数后面添加小写字母，比如JpComic105a.png。字母排序的时候永远把副图排在前，比如国际服的漫画经常会有副图。
- 漫画配字还未处理完毕，目前暂时存在bwiki上。

- 学院图标

- 保存在.../schools/ 文件夹中。预计以后会需要添加各个学院的图片素材。

- 其他

- 有一些比较杂乱的图标，比如地形图标和展开按钮等。目前保存在.../icons/ 内，但是尽量避免无用的素材堆积的情况。

## 程序概况

- 如前面提到的，bwiki上都是静态网站，我们在搭建的时候不能操作服务器后端的任何东西。不过项目开始的时候，我修好了mediawiki的import功能。因此在BAwiki的搭建过程中，我们可以使用独立的css和js文件，json数据也可以摆脱mw语法在js程序内导入。不过import功能还是仅限于独立文件的导入，比如导入每个模板对应的css和js程序，不能够导入模组。导入现成的js包不仅会导致网站非常卡顿，而且在没有Node.js这种管理器的情况下，很容易发生冲突，而且以后很难维护。因此BAwiki的搭建过程中，所有的功能要从零写起。
- Bwiki上的mediawiki引擎应该有一段时间没有更新了，而且被b站自己修改了很多。所以在mediawiki用来搭网站本来就不完善的语法的基础上，又有一大部分在bwiki上用不了。最经典的一个例子就是 <button> 在bwiki上无法使用。这种情况有两种解决办法，这两种思路都会在网页建设

中用到。第一种就是自己通过jquery锁定html元素，然后自己手写onClick。这样的好处就是按钮的动画效果可以做细节的调整，按钮的设计也会灵活很多。第二种就是运用mediawiki语法中图片默认是超链接的特性，在需要的位置覆盖透明图片并修改图片的默认超链接。第一种在很多情况下都会好用很多，不过在功能需求相对简单或者按钮组件出现频率很高的时候（比如漫画板块本质上会有上百个按钮），会考虑使用第二种写法，降低需要import的次数。

- 由于BAwiki网站上mediawiki的import是我自己修好的，所以其实在bwiki上，js程序的功能不是很完善。最重要的一点就是关于变量的封装。虽然很多情况下，js程序中涉及的是写在\$(function() {}内的jQuery，但还是会有在js程序中直接被定义的变量。经过测试这些变量虽然可能只是在某一个模板下的定义的，但是对于任意调用了这个模板的页面，无论嵌套多少层都会被当作全局变量。因此目前为止所有的开发都是以“所有的程序全在同一个文件内”的心态去搭建的。虽然一定有永远不可能调用到A模板的B页面，但是为了杜绝侥幸心理和代码的一致性，任何代码在被保存的时候必须能做到永远不会与其他程序冲突。具体的一些要求会在下面展开。
- 另外一点是bwiki的服务器不是很稳定，经常会有缓存更新非常缓慢的情况。这个问题在独立模板的编写中不是很突出，但是但凡出现了使用其他模板的情况就会非常明显。经常会出现代码需要五到十分钟甚至更长才能体现效果的情况，个别情况还会更久。因此非常建议在检查重要的程序的版本的时候，console.log程序保存时间并且找别人一起测试。

## 程序规范

- 不能修改common.css和common.js。这两个文件会被所有页面调用，而且是mediawiki默认程序。
- css内禁止使用id。否则一定会有莫名其妙id重复的一天。
- 所有的模块必须有wrapper进行封装。虽然这会让css和js程序中class的名字非常长，但这是最稳妥的办法。
  - 这里的封装指的是在html(mediawiki)文本最外层多套一层<div>。这个div的className必须是模板名字加上"Wrapper"作为后缀，比如StudentProfileWrapper，同时要求首字母大写。
  - 对应的css文件里所有的css也必须只能在这个wrapper下生效。方便起见可以在css完成以后使用scss/css转换器，自动添加class。
  - js程序中，内层模块应该尽量避免全局变量的使用。负责完整页面的js程序很多会不可避免地用到全局变量，通常是为了做出动态网页的效果。这种情况下务必对变量进行详细命名，json数据保存在和表格名称相同的变量内，其余标注页面名称。
- 所有的程序必须有工整的缩进。
- 尽可能保证代码的可维护性，避免大段的重复代码。
- 禁止使用中文变量名、程序名。
- import功能使用格式为<div class="loader">xxx</div>，支持导入css和js文件。XXX不需要写MediaWiki前缀、loader class会自动隐藏这个div。
- css尽量不在模板的mediawiki内编写。

## 程序debug

- 被发现的bug都会被列在bug收集表中，新发现的bug请按照格式填入。
- bug修复以后更新修复状态。如果为翻译内容的错误有可能进行批量上传，以及涉及到可能需要讨论的内容，需要先更新翻译修正状态，在网站上上传新版本后再更新修复状态。
- Bwiki的bug有很强的不确定性，所以额外增加验收环节以降低bug没有完全修复的可能性。

## json数据库

- BAwiki网站上所有的json文件都用于js程序生成动态页面，因此全部保存在MediaWiki命名空间下，并且禁止不负责程序代码的人进行编辑。同时也建议在需要增加网页内容的时候再调整数据库，以避免数据的重复和混淆。
- 数据库全都在js程序中的ajax request内进行调用，并储存在全局变量中。读取链接为json文件的纯文本格式，/ba/index.php?title=MediaWiki:XXX.json&action=raw。
- 由于bwiki服务器的不确定性，尽量在数据库内只保存对应页面所需要的数据，无法重复使用的数据过多或者文件多大是需要分开储存。

## 注意事项

- 网页链接中的querySelector值变化不会导致bwiki后台刷新，此类页面必须完全从js生成。
- 在调用模板的时候，如果遇到一直不刷新的情况，可以尝试添加一个额外的变量并更改赋值。

## 重要css元素

- BAwiki字体使用的是PuHuiTi v2，此字体已经被设置成默认。
- 倾斜的方框、按钮等，倾斜角为8deg。
- 常用颜色：
  - 普鲁士蓝：#003153
  - 爆炸属性：#8B0000
  - 贯通属性：#B8860B
  - 神秘属性：#4682B4
- 页面宽度设置：
  - 手机端 (<600px)：页面min-width设置为350px，宽度自适应。
  - 平板 (>=600px, 1350px>)：页面min-width设置为600px，宽度上限待定，自适应。
  - 电脑 (>=1350px)：margin自适应，注意漂浮窗位置。

## 常用Mediawiki语法及功能

- 调用模板：{{ 模板名 | 变量A=x | 变量B=y }}
- 模板内变量：{{{ 变量A }}}
- 图片：[[ File:XXX.png | link= ]]
- 链接：[[ 链接文字 | 链接 ]]
- Alt+Shift+E：查询页面源代码，编辑代码按钮被隐藏的页面同样适用