链滴

思源特性提议：外挂样式

作者：EndlessErrors
原文链接：https://ld246.com/article/1649779886927
来源网站：链滴

背景：为什么需要这个特性

我观察到下面这些现象：

1. 用户如果 **需要定制一些细微的样式，需要深入到 theme.css 中修改代码**，这对非代码用户极不好，而这个需求是很广泛的（不然也不需要开一个装修QQ群了）

2. **自定义属性+样式**，对于主题开发来说已经成为一个特别实用的技巧了，尤其是 Zhang-Light 主题将其发挥到极致

这种方法可以实现很多很多功能，甚至添加原本没有的新样式。

实际上，对于用户，可以**把添加自定义属性这个过程给封装为更友好的形式**（见下文）

3. **主题作者之间的代码复用**问题：一个主题作者有时会开发出适用于任何主题的样式（比如说标题居、限制页面宽度……），假如其他作者想在自己的主题里引入这个样式，必须修改 theme.css，实际这个动作是不必要的，万用的样式应该可以独立出主题之外，**成为类似于“库”或者“包”一样的存在**

于是，我认为外挂样式一定是主题的一个趋势。

下面先介绍对于用户来说，外挂样式这一功能意味着什么。

对于用户

在集市中可以方便地下载和配置外挂样式（参考VSCode的插件）

具体能够配置的内容是：样式是否启用，样式的优先级（下文详述）（图里没有，是我懒得画了）

对于“自定义属性+样式”的功能，可以像修改块的外观一样，一键添加自定义属性来用指定的样式示。

对于思源和主题开发者：实现细节

由于下面是需要真正落实到代码的一些细节，所以我尽量考虑到现实中会遇到的复杂度。

外挂样式如何工作

原文链接：思源特性提议：外挂样式

外挂样式可以简单理解为是一个CSS文件，可以有多个外挂样式，它们可以在主题的 theme.css 之前之后加载，并且可以覆盖 theme.css，或被 theme.css 覆盖，外挂样式之间也可以相互覆盖。

另外主题也可以将其中的功能提取为外挂样式，这样这些功能也可以在其它主题中应用。

下面举一个例子：（从上到下依次加载css文件，删除线代表该样式被后续加载的CSS覆盖）

1. 外挂样式 A：
    - 将自定义属性"f"=="table"的列表显示为表格样式
    - ~~令上述表格背景为白色~~
2. 主题T：
    - 由于主题为红色调，将~~标记背景色改为红色~~
3. 下载 T 时附带的外挂样式 T.a ：
    - 提供一个可开可关的功能：H1标题居中
4. 下载 T 时附带的外挂样式 T.b ：
    - 提供一个可定制（覆盖）主题 T 的功能：把 T 的主色调改为蓝色
5. 下载 T 时附带的外挂样式 T.c ：
    - 为主题 T 适配外挂样式A，
    - 由于主题为红色调，将自定义属性"f"=="table"的列表的表格背景改为淡红色 **(覆盖外挂样式A)**
6. 外挂样式 C：
    - 最后加载，优先级最高，未被别的样式覆盖
    - 将标记的背景色强制改为粉红色 **(覆盖主题T)**

开发外挂样式的接口参考

外挂样式可以上传到集市，也可以像挂件一样解压使用。

准确来说，外挂样式并非单独的 .css 文件，下载下来的实际是一个外挂样式模块，

下载或解压到SiYuanWorkspace\conf\appearance\styles后，文件结构如下

- SiYuanWorkspace\conf\appearance\styles
    - 某个外挂样式模块的文件夹
        - 若干个 .css 文件
            - 一些 .css 提供了其他 .css 的前置依赖
            - 一些.css 可以在白天主题加载，而另一些则在夜间加载
        - style.json 对外挂样式进行配置
        - 可选的 .js 文件

style.json 文件像下面这样（后面会对每个字段有说明）：

```
{
    "moduleName": "便签",    // 模块名，是会显示给用户的
    "author": "XX",
    "url": "https://github.com/XXX",
    "version": "1.0.0",
    "enable": true,        // 是否启用该样式模块，可以在思源设置界面配置
    "priority": "low",     // 样式模块的优先级，可以在思源设置界面配置

    "cssFiles": [
        {
            "path": "base.css",
            // 该外挂样式模块的依赖css
            // 实际上，可以不在该文件中写出，而直接在主要的.css中import即可
        },
        {
            "path": "stickyNote_DayLight.css",
            "dayOrNight": "dayLight",
            // 日间主题的便签
        },
        {
            "path": "stickyNote_MidNight.css",
            "dayOrNight": "midNight"
            // 夜间主题的便签
        }
    ],
    "click2Add": [
        {
            "buttonLabel": "便签（红）",
            "actionWhenAdding": [
                ["parcelWith", "quote"]
            ],
            "customAttr": [
                {
                    "name": "css",
                    "val": "stickyNote"
                },
                {
                    "name": "stickyNoteCfg",
                    "val": "redBg"
                }
            ]
        },
        {
            "buttonLabel": "便签（蓝）",
            "actionWhenAdding": [
                ["parcelWith", "quote"]
            ],
            "customAttr": [
                {
                    "name": "css",
                    "val": "stickyNote"
                },
```

```
            {
                "name": "stickyNoteCfg",
                "val": "blueBg"
            }
        ]
    }
  ]
}
```

其中：

"priority": 优先级（决定了加载顺序）
"final": 优先级超过主题和其它外挂样式，实现方式是最后才加载（只允许本外挂样式修改其它外挂样式，不允许其他外挂样式修改本样式。当然这只是一个像!important一样未必总是兑现的承诺

"high": 优先级高，一定保证在主题样式之后加载，该样式可以覆盖主题样式，但可能会被其它挂样式覆盖
"low" （默认）: 优先级低，最先（在主题样式之前）加载，该样式可以被主题和其它外挂样式盖

"cssFiles"[]: 给出外挂样式模块中的所有css文件，并设置它在白天/夜间主题下启用性，思源主程序按顺序依次加载对应主题下的css文件
[
{
"dayOrNight": 白天/夜间主题启用设置
"both"（默认）: 对于日间/夜间主题，均启用css
"dayLight" || "midNight": 单独为日间/夜间主题启用css

"loadOnlyIf"[]: 如果这个css是为了适配（覆盖）其它外挂样式或主题的，那么可以把对方的模名或主题名加在这里，
只有当思源主程序检测到列表里的样式都加载后，才会加载该css
}
]

"click2Add"[]: 通过点击为块添加特殊的自定义属性，以令该块显示为外挂样式。这个字段为数组，出可以由用户点击添加的样式自定义属性
[
{
"buttonLabel": 显示给用户的，样式按钮标签
"relatedBlock": ["p", ...] : 可以给那些种类的块添加这种样式。缺省时为对所有块均启用
"customAttr"[]: 想要添加的自定义属性列表
[
{
"name": 自定义属性名
"val": 自定义属性值
}
]
"actionWhenAdding"[]: 添加自定义属性的同时，会执行的特殊行为的列表
假如该列表为空，则默认行为是：给每个选中的块添加“customAttr”中指的属性
假如该列表不为空，则默认行为被禁用，然后该列表中的各个特殊行为按下标序从小到大执行

其中的每个特殊行为均以数组给出，第一个元素为操作名，后面的元素为参数：

```
        ["replace", regExp , newExp ] :    对选中的块的 Kramdown（或Json）源码进行一次正则
换
        ["parcelWith", "quote"]:    为一个非引述块或多个块添加外挂样式时，先将这些块用一个引
块包裹，然后对包裹的引述块添加自定义属性，并结束特殊行为
        ["parcelWith", "super"]:    同上，只不过改为用超级块包裹
        // 感觉对源码的若干次正则替换已经可以解决绝大多数情景了，后两者也可以用正则替换实
，不过单独列出来方便主题开发者开发
    }
]
```

可以看出，虽然一个外部样式模块可能包含多个.css文件，但是在日间/夜间模式确定的情况下，大多模块里主要的.css文件只有一个，因此多数时候可以把外部样式模块简单视为一个.css文件。

主题和外挂样式的交互

主题的 theme.json 配置文件中可以添加一个字段"externStyles"：

```
{
    "name": "daylight",
    "author": "Vanessa",
    "url": "https://github.com/Vanessa219",
    "version": "1.0.0",
    "modes": [
      "light"
    ],
    "externStyles":[
        {
            "moduleName": "foo",    // 和对应的外挂样式模块的命名一致，不需要在这里指定路径，
思源主程序负责寻找本地路径加载或联网下载
            "url": "https://github.com/...",
            "isEnablingSuggested": true    // 主题建议，该外挂样式模块是否推荐启用
                                    // 如果建议启用，则从集市下载主题的时候同时也下载该外挂样式，并
在每次启用主题的时候顺带会启用该外挂样式
                                    // 如果不建议启用，则从集市下载主题的时候不会同时下载该外挂样式
并且在每次启用主题的时候顺带会禁用这个外挂样式
        }
    ]
}
```

下载主题时“附带”的外挂样式并不隶属于主题，仍然是独立的外挂样式，可以单独设置外挂样式的用与否。之所以写在主题配置文件里，单纯是因为主题作者认为这个样式符合主题风格或者很漂亮。就像你下载了思源，里面用到了 Pandoc，并不意味着 Pandoc 隶属于思源）

假如想对其它外挂样式进行适配，应该引用一个专门用于定制其它外挂样式的自定义外挂样式模块，个样式模块应该像下面这样：

```
{
    "moduleName": "XX主题：定制外部样式", // 样式模块名，是会显示给用户的
    "author": "XX",
    "url": "https://github.com/XXX",
    "version": "1.0.0",
    "enable": true, // 是否启用该样式模块，可以在思源设置界面配置
    "priority": "high" || "final", // 为了覆盖别的样式模块，应该提高优先级
    "cssFiles": [
```

```
        {
            "path": "stickyNote_ForXXX.css",
            "loadOnlyIf"[
                "XXX（本主题）", "便签"     // 需要本主题和对应模块均加载时才进行定制
            ]
        },
        {
            "path": "list2Table_ForXXX.css",
            "loadOnlyIf"[
                "XXX（本主题）", "列表显示为表格"
            ]
        }
    ]
}
```

需要特别注意的是：主题附带的外挂样式对于主题来说均是**可选**的，**用户可以选择把这些外挂样式全关闭。**假如主题依赖于某个css，**那不应该把它分离成外挂样式**，这也不是外挂样式这个概念出现想解决的问题。**外挂样式是面向用户而非开发者的，不应该完全等同于编程中的“库”，它更像思源的件，每个外挂样式应该提供一个完整的样式功能。**

另外也不要在一个样式模块里面添加多种样式，一个模块对应一个功能即可，方便开启关闭。

（就像卡片式笔记一样，追求独立化）

甚至可以为“H1标题居中”、"H2居中"、"H3居中"……分别写一个模块。（主要是为了方便非码农户定制，能不进.css改代码就是最大的成功）

对主题大佬的一些希望

- 希望一些主题大佬率先把一些常见的功能作为外挂样式分离出来，作为公用的样式
    - （例子太多了，比如便签、语雀提示区块、额外的色彩、额外的字体、像挖空、波浪线、备注这类特殊的样式）
    - 这一方面能节省其它主题开发者的时间，一方面也尽早把与样式相关的自定义块属性统一下来
- 编写外挂样式的时候尽量使用 `:root{...}` 中的配色
- 编写外挂样式的时候，优先级尽量低，方便别人定制

结语

假如可以调用 JS 脚本的话，那肯定会方便不少，比如"loadOnlyIf"接口就可以简化了。不过我不太楚对于思源，JS的界限在哪里，所以上文都是假设不使用样式定制的JS。不过这么设计也行，可以减开发样式的代码量，毕竟大多数代码由思源开发者搞定了。

对于用户来说，很友好，不用再到theme.css里面修改什么了，也不用手动添加自定义块属性了。

而且自定义块属性配合样式，确实可以实现很多功能，像是标题居中之类的功能，其实没必要用插件现，用样式刚刚好。

而且对于主题开发者来说，模块化和结构化肯定是必然趋势，毕竟不是每个人都要造轮子的。

上面给出的接口设计，由于是空想的，仅供参考，欢迎大家讨论。