sakuraRenderer

注意

1. 本项目必须要安装element plus并且全局挂载（包括css）

样式顺序

articleGroup < articleContainer < option < articleConfig < elementClass < elementStyle < style

规范

1. 组件的name须以sr开头，单词全小写，单词之间通过-分割
2. css类名字按照BEM规范，须以sa开头，css样式不用写在vue下，写在配套的css文件里面
3. 组件用到的颜色变量在css文件开始的地方进行变量定义，颜色变量按照--sa-color+组件识别名字+颜色的位置
4. 被articleContainer直接调用的组件有且只有data一个props，类型是Object
5. 每个组件的css文件都不在组件内部，而存放在theme-chalk下面的css文件，文件名字和组件名字相同

渲染流程

1. 文档加载
    1. 预处理（暂无）
2. 划分区间
    1. 拆分配置项、文章主体
        - 通过\n|-\n划分
3. 配置项初始化
    1. 拆分配置项
    2. 合并同类配置项
    3. 与默认配置项合并
4. 拆分主体结构
    1. 确定不需要渲染的区域
        1. 将其从文章里面用特殊句子替换
    2. 确定代码区域
        1. 将其从文章里面用特殊句子替换
    3. 拆分成组件（大体）
5. 处理每一个主体架构
    1. 处理模板语法
    2. 将代码区域插回文章
    3. 对文章代码区域进行渲染
    4. 将不需要渲染的区域插回文章
6. 合并主体架构成为符合要求的渲染数组
7. 加入到布局管理器中（文章承载器）
8. 使用vue的component进行循环渲染
9. 渲染完毕之后进行数学公式渲染

页面结构

- 头部（可有可无）
  - 页面基础信息介绍
- 主体
  - 标题/段落/表格/列表/图片显示器
- 尾部（可有可无）
  - 贡献者显示器
  - 引用显示器
  - 页面信息显示器

组件（大体）

1. 标题 title
2. 段落 paragraph
3. 目录 catalogue
   1. 左右上下侧目录
   2. 浮动目录
4. 表格 table
5. 列表 list
6. 图片显示器 imageShower
   1. 全部图片显示器 allImageShower
   2. 走马灯图片显示器 carouselImageShower
   3. 相册图片显示器 albumImageShower
7. 引用显示器
8. 贡献者显示器
9. 页面信息显示器

注意：

1. 以上组件都不能相互嵌套（暂时）
2. 以上组件两辆之间必须要两个换行符

语法

注意，此部分和md有语法冲突，按照源代码为准！！

1. ~~包围 ~~删除线~~
2. **包围 **加粗**
3. 4个- 分割线
4. *包围 *斜体*
5. __包围 __下划线__
6. ~*包围 ~*粗斜体*~
7. 包围行内数学公式$

8. 包围 $$行间数学公式

- 删除线、加粗、下划线更加详细的设定需要在模板中编写

模板{{}}

- 模板都是被{{}}括起来
- 模板基础语法：{{模板名字|参数1|参数3=red}}
  - 模板名字是确定不可变的，如果填写不存在的模板名字会将去掉{{}}的代码放到界面上
  - 参数之间通过|分割
  - 模板的参数可能会有不同名字，在文档中用#分割
  - 不加=则参数按照顺序填写
  - 加上=则参数按照=进行赋值
  - 参数是可以选填的，详细会在各各模板介绍
  - 模板的对外显示内容（会有说明）里面的内容可以嵌套其他部分模板（会有说明）
  - 模板内部内容允许换行
  - 后设置的参数会覆盖前面设置的参数
  - 假定现有模板：{{测试#test|句子#content|颜色#color|大小#size}}，且参数：大小#size默认值是20px，则以下写法效果完全相同：
    - {{测试|测试的句子|red|20px}}
    - {{测试|颜色=red|句子=测试的句子|20px}}
    - {{测试|测试的句子|3=20px|2=red}}
    - {{test|color=red|size=20px|content=测试的句子}}
    - {{测试|color=red|content=测试的句子}}
    - {{测试|color=blue|content=测试的句子|color=red}}

1. {{黑幕#heimu#hide|内容#content|注释#title}}
2. {{模糊#blur|内容#content|注释#title|范围#size}}

3. {{文本#font|内容#content|颜色#color|大小#size|行高#height|字体#fontfamily|粗细#weight|背景色#bgcolor|斜体#italic|位置#align}}
4. {{删除线#del|内容#content|颜色#color|粗细#size}}
5. {{下划线#und|内容#content|颜色#color|粗细#size}}
6. {{注解#ruby|内容#content|解释#explain}}
7. {{超链接#a|链接#href|内容#content|提示#title}}
8. {{id|内容#content|ID}}
9. {{func|内容#content|click|mousein|mouseout|doubleclick}}
10. {{显示框#box|内容#content|框内内容#boxcontent|图片#src}}
11. {{引用#ref|内容#content|来源#from}}

标签

- 真实标签代表会在h5注册为自定义标签，通过h5渲染
- 虚假标签不是自定义标签，只是进行标记

1. sr-box/sr-b 行内鼠标浮动显示框
    - 真实标签
2. sr-ref/sr-r/ref 行内引用标签文档区域
    - 虚假标签
3. sr-code/sr-c/code 行内代码区域标签
    - 虚假标签
4. sr-poem/sr-p/poem 保留换行符区域标签
    - 虚假标签
    - 通过这个标签替换为pre
5. sr-ignore/sr-i/ignore 不对文本进行渲染
    - 虚假标签
6. sr-ref-item 行内引用到页面底部的引用区域的链接
    - 真实标签
    - 通过sr-ref/sr-r替换而来
    - 不对外开放
7. sr-ref-node 页面底部引用区域的链接
    - 真实标签
    - 通过sr-ref/sr-r标签被js识别之后添加
    - 不对外开放

模块[[]]

组件{||}

行内图像

- 行内图像是被[[]]括起来的
- 模块基础语法（格式）：[[img:文件路径|title=标题|选项]]
    - 参数之间通过|分割
    - 模块的参数可能会有不同名字，在文档中用#分割
    - 标题必须要以title进行赋值
- 参数
    - 支持的文件格式与html的img组件支持的格式相同
    - 标题可以包含模板
        - 标题默认值为：图片路径
    - 选项可选值：
        - 格式
        - 尺寸
            - 宽度px，例如：200px
            - x高度px，例如：x100px
            - 宽度x高度px
            - **默认值：300px**
        - 水平对齐
            - inline，行内元素
            - none，块状元素
            - left，块状元素，float:left
            - right，块状元素，float:right
            - center，块状元素，居中对齐
            - **默认值：inline**
        - 链接
            - link=链接，点击之后进行页面跳转
            - **默认值：link=，就是空值**

        - 替代文本
            - alt=替代文本
            - **默认值：alt显示title**

行内音乐（不重要，之后写）

- 行内音乐是被[[]]括起来的
- 模块基础语法（格式）：[[music:文件路径|title=标题|选项]]
    - 参数之间通过|分割
    - 模块的参数可能会有不同名字，在文档中用#分割
    - 标题必须要以title进行赋值

组件（详细）

- 部分组件后跟参数，参数填写方式与模板相同

标题

1. 格式：
    - = 第一标题 ?style ta=center|bp=l|ha=true
    - 前后至少有一个换行符
    - 通过=数量来判断是第几标题
    - 后跟?style可以设置样式
    - 如果希望标题有?style可以设置为?style，（见md源码）
    - 当然也可以通过格式来进行书写，以便于加入配置

```
{|title#标题|type=h1
|style=font-size:20px;width:100%
|class=class1;class2
|ta=center|bp=l|ha=true
|-
| 标题
|}
```

2. 参数（?style后的）
    1. 意义：

1. ta:text-align
    - 会将此值赋给标题css的text-align属性
    - 默认值：left
    - 可选值：
        - left
        - center
2. bp:border-position
    - 不会在标题上直接添加边框
    - 可选值：
        - l/left：左侧边框
        - b/bottom：底部边框
        - n/none：没有边框
    - 默认值：l
3. ha:hover-animation
    - 是否存在鼠标浮动动画，动画效果是边框从无到有
    - 此值只要存在即认为有动画，也就是说ha等同于ha=true
    - 可选值：
        - true：有动画
        - false：没动画
    - 默认值：false
4. hl:has-link
    - 鼠标浮动时是否出现可以点击的#，进行页面滚动
    - 此值只要存在即认为有动画，也就是说hl等同于hl=true
    - 可选值：
        - true：有可以点击的#
        - false：没有可以点击的#
    - 默认值：false

段落

1. 格式：
    - 通过两个换行符来进行区分
    - 段落为了简写可以直接写内容
    - 当然也可以通过格式来进行书写，以便于加入配置
    - 段落会存在类型type
        - default 默认，一般段落
        - success 成功，（warning的配色改变）

        - warning 警告，样式见https://element-plus.org/zh-CN/component/button.html#%E9%93%BE%E6%8E%A5%E6%8C%89%E9%92%AE
        - info 信息，（warning的配色改变）
        - tip 要点，（danger的配色改变）
        - custom 自定义，必须填入边框颜色和背景颜色
        - 注意！！！！warning的颜色用的sa-color-danger的配色！！！
    - 类型可以设置类型标题

```
{|para#段落|style=font-size:20px;width:100%
|class=class1;class2
|-
| 段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。
| 段落第二句话，这部分与上面一句相比，已经换行了。段落第二句话，这部分与上面一句相比，已经换行了。
段落第二句话，这部分与上面一句相比，已经换行了。
|}
```

```
{|para#段落|type=warning
|title=警告
|-
| 段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。
| 段落第二句话，这部分与上面一句相比，已经换行了。段落第二句话，这部分与上面一句相比，已经换行了。
段落第二句话，这部分与上面一句相比，已经换行了。
|}
```

```
{|para#段落|type=custom
|title=警告|bc=red|bgc=white
|-
| 段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。段落第一句话，这不会换行。
| 段落第二句话，这部分与上面一句相比，已经换行了。段落第二句话，这部分与上面一句相比，已经换行了。
段落第二句话，这部分与上面一句相比，已经换行了。
|}
```

2. 参数
    1. 意义：
        1. lh:line-height
            - 行高
            - 可选值：

        - 会将此值直接赋值给css的line-height属性
    - 默认值：1
    - 值填写出错(不为正数)：1
2. bc:border-color
    - 边框颜色
    - 如果type是default，则是整个段落的边框色，如果type是custom，则是左边框色，其余的此值无效
    - 可选值：
        - 颜色
    - 默认值：rgba(0, 0, 0, 0)
    - 值填写出错：无法检测，不做处理
3. border
    - 边框
    - 如果type是default，则是整个段落的边框设置，其余的此值无效
    - 可选值：
        - 直接将此值赋值给css的border属性
    - 默认值：-
    - 值填写出错：无法检测，不做处理
4. bgc:background-color
    - 背景颜色
    - 如果type是default或者type是custom，则是整个段落的背景色，其余的此值无效
    - 可选值
        - 颜色
    - 默认值：rgba(0, 0, 0, 0)
    - 值填写出错：无法检测，不做处理
5. type
    - 段落类型
    - 可选值：
        - default 默认
        - success 成功
        - warning 警告
        - tip 提示
        - info 信息
    - 默认值：default
    - 值填写出错：default
6. tips
    - 段落的提示，当且仅当段落类型不为default时生效

- 可选值：
    - 空，此时根据类型使用默认值（见下）
        - TIPS 提示 type=tip
        - WARNING 警告 type=warning
        - SUCCESS 成功 type=success
        - INFO 信息 type=info
    - 自定义值，当类型不为default时，替换界面上的提示信息
        - 此值允许使用模板
        - 默认颜色、大小、粗细和提示值的css相同
        - 或许要做一下字符串检查？
- 默认值：-
- 值填写出错：-

表格

1. 格式：

```
{|table#表格|fold|style=font-size:20px;width:100%
|class=class1;class2
|-
|+ 标题 c=8/t
|-
| 这会占用四个格子 c=2/r=2/t
| 早上 c=2/t | 下午 c=2/t | 晚上 c=2/t
|-
| [[img:test.jpg|width=100px]]
| {{font|学习|red}}
| 睡觉
| 打游戏 | 看番 | 睡觉
|}
```

```
{|table
|-
|+ 标题 | a | d | d |
|-
| sa | as | as |
|}
```

2. 第一行后面填写的区域是代表表格的设置
    - fold代表表格是否可以折叠
    - style会直接赋给表格

3. 第一个|-出现之后才是表格信息区域
4. 关键字c：代表单元格横向占多少格子
5. 关键字r：代表单元格竖向占多少个格子
6. 关键字t：代表单元格是否是标题
7. 第一个|-出现之后的第一行如果带上+说明是thead
8. 之后的|-代表表格换行

列表

1. 格式：

```
{|list#列表|
|class=class1;class2
|-
|+有序列表1
|+有序列表2
|++有序列表2.1
|++有序列表2.2
|+有序列表3
|*无序列表
|**无序列表
|*+
|}
```

图片显示器

1. 格式：
    - 规则

```
{|图片显示器名字
| 参数列表
|-
| 图片地址列表
| 图片地址列表
|}
```

2. 共有参数
    - 尺寸
        - 宽度px，例如：200px，每一张图片的宽度

        - x高度px，例如：x100px，每一张图片的高度
        - 宽度x高度px，例如：200x100px，每一张图片的宽高
        - 图片下标?尺寸，例如：1?200px，代表：第一张图片宽度是200px
        - **上述参数不同组件默认值略有不同，详细参考下分的组件**
        - width=容器宽度/w=容器宽度
            - **默认值：100%**
        - height=容器宽度/h=容器高度
            - **默认值：空（自适应），此处填写%单位数值无效**
    - 水平对齐
        - none，块状元素
        - left，块状元素，float:left
        - right，块状元素，float:right
            - **设置float时，width可能会发生改变**
        - center，块状元素，居中对齐
        - **默认值：center**
    - 样式
        - style=css语句，不可换行，每条语句之间通过;分割（会直接赋值给最外层组件的style
        - class=类名，不可换行，每个类之间通过;分割

1. allImageShower 类似于9宫格那样的图片展示器，（当然并不是九宫格

```
{|allIS#图片展示框
| width=500px
|-
| img1.jpg
| img2.png
| img3.jpg
| img4.jpeg
|}
```

2. carouselImageShower

```
{|carouselIS#走马灯图片展示框
| width=500px
|-
| img1.jpg
| img2.png
| img3.jpg
| img4.jpeg
|}
```

3. albumImageShower

```
{|albumIS#相册图片展示框
| width=500px
|-
| img1.jpg
| img2.png
| img3.jpg
| img4.jpeg
|}
```