README

General（通用）

Chinese problem（中文语言问题和网站访问问题）

- 语言问题：文档是多语言的（zh/en），不用担心
- 访问失败/被墙：本文的默认网站链接指向 github.io，如果国内有不能访问的朋友，将网站链接部分的 *linczero.github.io* 替换成 *linczero-github-io.pages.dev* 就可以了

More Links（相关链接）

Related links: (tutorial、use skill、contribution、secondary development)

- Online Wiki - github.io
- Online Wiki - cloudflare

Effects warrior（效果展示）

（推荐使用暗黑模式进行查看）

下面在展示中，会使用标签页将几个版本进行对照。以便来回查看以体会用法和语法：

::: tabs

@tab en

- Plugin effect
  - Effect: It is the result of the actual rendering after using the plugin, not the screenshot
  - It will show: Plugin real effect
- No plugin effect
  - Effect: Render the result after you close the plugin
  - It will show: The plugin syntax is almost non-intrusive, and in a plugin-free environment, md documents remain elegantly readable rather than becoming code boxes
- Md source code
  - Effect: Open with Notepad/See the result in Ob's source mode
  - It will show: Convenient for you to see how this effect is written, showing the elegant and efficient plugin syntax

@tab zh

- 插件效果
  - 效果：是使用插件后真实渲染出来的结果而不是图片截图
  - 展示了：插件的真实效果
- 无插件效果
  - 效果：你关闭了插件后的渲染结果
  - 展示了：该插件语法几乎无入侵性，在无插件环境下，md文档依然优雅可读，而非变成代码框
- md源码
  - 效果：用记事本打开/在Ob的源码模式中看到的结果
  - 展示了：方便你看这种效果是怎么写出来的，表现了插件语法的优雅高效

:::

list2table（列表转多叉表格）

This table, I call a `multiWay table` / `multiCross table` / `Cross table` (这种表格我命名为 `多叉表格` / `跨行表格`)

It is characterized by `multi-tree` structure, The table allows cross rows. (特点是结构就像 `多叉树` 一样，表格允许换行)

> **Allows cells to cross rows and inline styles、'|' grammar candy (允许单元格跨行、内嵌样式、"|"语法糖)**

md-demo Describe common angiosperms and gymnosperms (描述一下常见被子植物与裸子植物)

::: tabs

@tab Plugin effect (插件效果)

[list2table]

- Gymnosperm
  裸子植物
  - Cypress
    松树
    - Chinese pine
      油松
    - Buddhist pine
      罗汉松
    - masson pine
      马尾松
    - Pinus koraiensis
      红松
  - Ginkgo
    柏树| (This is ~~just~~ **a** `style` *test*)
    (这 ~~仅仅~~ 是 **一个** `样式` *测试*)
  - Angiosperms
    银杏
- Angiosperm
  被子植物
  - Sunflower
    向日葵
  - Lotus
    荷花
  - Chrysanthemum
    菊花 | Chamomile
    甘菊

@tab No plugin effect (无插件效果)

(noPlugin)[list2table]

- Gymnosperm
  裸子植物
  - Cypress
    松树
    - Chinese pine
      油松
    - Buddhist pine
      罗汉松
    - masson pine
      马尾松
    - Pinus koraiensis
      红松
  - Ginkgo
    柏树| (This is ~~just~~ a `style` *test*)

(这 仅仅 是一个 样式 测试)

- Angiosperms
  银杏

- Angiosperm
  被子植物
  - Sunflower
    向日葵
  - Lotus
    荷花
  - Chrysanthemum
    菊花 | Chamomile
    甘菊

@tab md source code (md源码)

```
[list2table]

- Gymnosperm<br> 裸子植物
  - Cypress<br> 松树
    - Chinese pine<br> 油松
    - Buddhist pine<br> 罗汉松
    - masson pine<br> 马尾松
    - Pinus koraiensis<br> 红松
  - Ginkgo<br> 柏树| (**This** is ~~just~~ a `style` *test*)<br> (**这** ~~仅仅~~ 是一个 `样式` *测试*)
  - Angiosperms<br> 银杏
- Angiosperm<br> 被子植物
  - Sunflower<br> 向日葵
  - Lotus<br> 荷花
  - Chrysanthemum<br> 菊花 | Chamomile<br> 甘菊
```

:::

Allow rotation table (允许旋转/转置表格)

(DEV TODO: 目前是纯css实现，间距不理想，且如果复杂的会有bug (如多叉时)，等待通过js手段真正意义上实现)

This is a comparison table between Chinese and English (这是一个中英对照表)

::: tabs

@tab Plugin effect (插件效果)

[list2tableT]

- Cypress | 松树
- Ginkgo | 柏树
- Angiosperms | 银杏
- Sunflower | 向日葵
- Lotus | 荷花
- Chrysanthemum | 菊花

@tab No plugin effect (无插件效果)

(noPlugin)[list2tableT]

- Cypress | 松树
- Ginkgo | 柏树
- Angiosperms | 银杏
- Sunflower | 向日葵
- Lotus | 荷花
- Chrysanthemum | 菊花

@tab md source code (md源码)

```
[list2tableT]

- Cypress | 松树
- Ginkgo  | 柏树
- Angiosperms | 银杏
- Sunflower | 向日葵
- Lotus | 荷花
- Chrysanthemum | 菊花
```

:::

> **Allows other blocks to be embedded、Allows tabel header (允许内嵌其他块、允许表头)**

Describe print statements for various common programming languages (描述一下各种常见编程语言的打印语句)

::: tabs

@tab Plugin effect (插件效果)

[list2mdtable]

- < Language
  语言
  - Print statement
    打印语句
    - characteristic
      特点

- Java
  - ```
    System.out.
        println("Hello World");
    ```
    - This sentence is a little long
      这语句有点长

- C
  - ```
    printf("Hello World");
    ```
    - The raw C output
      原始的C输出

- C++
  - ```
    std::cout<<"Hello Wrold";
    // <<std::end;
    ```
    - Stream output, but this thing has a high performance overhead

      流输出，但是这东西开销大

      > - newline: `<<std::end;`
      > - 换行: `<<std::end;`

- Python
  - ```
    print("Hello World")
    ```
    - Note that Python2 and Python3 have different print statements
      需要注意一下Python2和Python3的打印语句不同
      |python2|python3|

|---|---|
| print "" | print("") |

- JavaScript

  - ```
    console.log("Hello World");
    ```

    - Console printing
      控制台打印

@tab No plugin effect (无插件效果)

(noPlugin)[list2mdtable]

- < Language
  语言
  - Print statement
    打印语句
    - characteristic
      特点

- Java

  - ```
    System.out.
        println("Hello World");
    ```

    - This sentence is a little long
      这语句有点长

- C

  - ```
    printf("Hello World");
    ```

    - The raw C output
      原始的C输出

- C++

  - ```
    std::cout<<"Hello Wrold";
    // <<std::end;
    ```

    - Stream output, but this thing has a high performance overhead

      流输出，但是这东西开销大

      > - newline: `<<std::end;`
      > - 换行: `<<std::end;`

- Python

  - ```
    print("Hello World")
    ```

    - Note that Python2 and Python3 have different print statements

      需要注意一下Python2和Python3的打印语句不同
python2	python3
print ""	print("")

- JavaScript

  - ```
    console.log("Hello World");
    ```

- Console printing
  控制台打印

@tab md source code (md源码)

```
[list2mdtable]

- < Language<br>语言
  - Print statement<br>打印语句
    - characteristic<br>特点
- Java
  - ```java
    System.out.
        println("Hello World");
    ```
    - This sentence is a little long
这语句有点长
- C
  - ```c
    printf("Hello World");
    ```
    - The raw C output
原始的C输出
- C++
  - ```cpp
    std::cout<<"Hello Wrold";
    // <<std::end;
    ```
    - Stream output, but this thing has a high performance overhead
      流输出，但是这东西开销大
      > - newline: `<<std::end;`
      > - 换行: `<<std::end;`
- Python
  - ```python
    print("Hello World")
    ```
      - Note that Python2 and Python3 have different print statements
需要注意一下Python2和Python3的打印语句不同
python2	python3
`print ""`	`print("")`
- JavaScript
  - ```js
    console.log("Hello World");
    ```
      - Console printing
控制台打印
```

:::

> It makes it look like a list instead of a table (可以让他看起来像列表而不是表格)

::: tabs

@tab Plugin effect (插件效果)

[list2table|addClass(ab-table-fc)|addClass(ab-table-likelist)]

- A1
  - B1
  - B2
- A2
  - B1
  - B2
    - C1
    - C2
      - D1
  - B3
- A3
  - B1
```

    - B2

@tab No plugin effect (无插件效果)

注意：这里的头部可以用%%注释不显示，也可以用别名机制使其简短，这里只是为了展示可以串联多个ab转换器进行修改。
正常来说没有这么长的

(noPlugin)[list2table|addClass(ab-table-fc)|addClass(ab-table-likelist)]

- A1
    - B1
    - B2
- A2
    - B1
    - B2
        - C1
        - C2
            - D1
    - B3
- A3
    - B1
    - B2

@tab md source code (md源码)

```
[list2table|addClass(ab-table-fc)|addClass(ab-table-likelist)]

- A1
  - B1
  - B2
- A2
  - B1
  - B2
    - C1
    - C2
      - D1
  - B3
- A3
  - B1
  - B2
```

:::
or

::: tabs

@tab Plugin effect (插件效果)

[list2c2t|addClass(ab-table-fc)|addClass(ab-table-likelist)]

- A1
    - B1
    - B2
- A2
    - B1
    - B2
        - C1
        - C2
            - D1

    - B3
- A3
    - B1
    - B2

@tab No plugin effect (无插件效果)

(noPlugin)[list2c2t|addClass(ab-table-fc)|addClass(ab-table-likelist)]

- A1
    - B1
    - B2
- A2
    - B1
    - B2
        - C1
        - C2
            - D1
    - B3
- A3
    - B1
    - B2

@tab md source code (md源码)

```
[list2c2t|addClass(ab-table-fc)|addClass(ab-table-likelist)]

- A1
  - B1
  - B2
- A2
  - B1
  - B2
    - C1
    - C2
      - D1
  - B3
- A3
  - B1
  - B2
```

:::

list2listTable (列表转列表格)

This table, I call a `ListTable` / `TreeTable` / `TreeGrid` (这种表格我命名为 `列表格` / `树形表格`)

The characteristic is that it basically conforms to multi-tree, but each tree node allows annotations. (特点是基本符合多叉树，但每个树节点允许存在注释)

When used to represent a file directory, it is also called a `CatalogTable` (当用于表示文件目录时，也叫 `目录表`)

Allow Folding (允许折叠)

md-demo This is the hierarchy of a certain company (这是某个公司的层次结构)

(DEV TODO: fix bug: 多行时，折叠功能似乎存在问题)

::: tabs

@tab Plugin effect (插件效果)

[list2lt]

- < Company name
  公司名| Superior section
  上级部门| Principal
  负责人| Phone
  电话
- **ABC head office**| | |
  - Shanghai branch| **ABC head office** | ZhangSan| 13&xxxxxxxx
    - Marketing section| Shanghai branch | LiSi|
self	father	mother
201xxxxx	202xxxxx	203xxxxx
      - Marketing Division 1| | |
      - Marketing Division 2| | |
    - Sales section| Shanghai branch | WangWu| 15&xxxxxxxx
  - *Beijing branch*| **ABC head office** | ChenLiu| 16&xxxxxxxx
    - Technical division| *Beijing branch* | OuYang| 17&xxxxxxxx
    - Finance| *Beijing branch* | HuangPu| 18&xxxxxxxx

@tab No plugin effect (无插件效果)

(noPlugin)[list2lt]

- < Company name
  公司名| Superior section
  上级部门| Principal
  负责人| Phone
  电话
- **ABC head office**| | |
  - Shanghai branch| **ABC head office** | ZhangSan| 13&xxxxxxxx
    - Marketing section| Shanghai branch | LiSi|
self	father	mother
201xxxxx	202xxxxx	203xxxxx
      - Marketing Division 1| | |
      - Marketing Division 2| | |
    - Sales section| Shanghai branch | WangWu| 15&xxxxxxxx
  - *Beijing branch*| **ABC head office** | ChenLiu| 16&xxxxxxxx
    - Technical division| *Beijing branch* | OuYang| 17&xxxxxxxx
    - Finance| *Beijing branch* | HuangPu| 18&xxxxxxxx

@tab md source code (md源码)

```
[list2lt]

- < Company name<br>公司名| Superior section<br>上级部门| Principal<br>负责人| Phone<br>电话
- ==ABC head office==| | |
  - **Shanghai branch**| ==ABC head office==  | ZhangSan| 13&xxxxxxxx
    - Marketing section| **Shanghai branch** | LiSi|
    |self    |father  |mother  |
    |--------|--------|--------|
    |201xxxxx|202xxxxx|203xxxxx|
      - Marketing Division 1| | |
      - Marketing Division 2| | |
    - Sales section| **Shanghai branch** | WangWu| 15&xxxxxxxx
  - *Beijing branch*| ==ABC head office==  | ChenLiu| 16&xxxxxxxx
    - Technical division| *Beijing branch* | OuYang| 17&xxxxxxxx
    - Finance| *Beijing branch* | HuangPu| 18&xxxxxxxx
```

:::

> **It makes it look like a list instead of a table（可以让他看起来像列表而不是表格）**

This is also called "optimizing list"（这也叫优化列表）

The essence is "listtable" based on the addition of a mock list style（本质是 "列表格" 的基础上增加仿列表样式）

::: tabs

@tab Plugin effect（插件效果）

[list]

- vue-demo/
  - build/， 项目构建(webpack)相关代码
  - config/， 配置目录，包括端口号等。我们初学可以使用默认的
  - node_modules/， npm 加载的项目依赖模块
  - src/， 这里是我们要开发的目录
    - assets/， 放置一些图片，如logo等
    - components， 目录里面放了一个组件文件，可以不用
    - App.vue， 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录
    - main.js， 项目的核心文件。
  - static/， 静态资源目录，如图片、字体等
  - test/， 初始测试目录，可删除
  - .eslintignore
  - .gitignore， git配置
  - .index.html， 首页入口文件，你可以添加一些 meta 信息或统计代码啥的
  - package.json， 项目配置文件
  - READED.md， 项目的说明文档，markdown 格式
    手动换行测试
    自动换行测试k
  - ...

@tab No plugin effect（无插件效果）

(noPlugin)[list]

- vue-demo/
  - build/， 项目构建(webpack)相关代码
  - config/， 配置目录，包括端口号等。我们初学可以使用默认的
  - node_modules/， npm 加载的项目依赖模块
  - src/， 这里是我们要开发的目录
    - assets/， 放置一些图片，如logo等
    - components， 目录里面放了一个组件文件，可以不用
    - App.vue， 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录
    - main.js， 项目的核心文件。
  - static/， 静态资源目录，如图片、字体等
  - test/， 初始测试目录，可删除
  - .eslintignore
  - .gitignore， git配置
  - .index.html， 首页入口文件，你可以添加一些 meta 信息或统计代码啥的
  - package.json， 项目配置文件

    - READED.md， 项目的说明文档，markdown 格式
      手动换行测试
      自动换行测试k

    - ...

@tab md source code（md源码）

```
[list]

- vue-demo/
  - build/， 项目构建(webpack)相关代码
  - config/， 配置目录，包括端口号等。我们初学可以使用默认的
  - node_modules/， npm 加载的项目依赖模块
  - src/， 这里是我们要开发的目录
    - assets/， 放置一些图片，如logo等
    - components， 目录里面放了一个组件文件，可以不用
    - App.vue， 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录
    - main.js， 项目的核心文件。
  - static/， 静态资源目录，如图片、字体等
  - test/， 初始测试目录，可删除
  - .eslintignore
  - .gitignore， git配置
  - .index.html， 首页入口文件，你可以添加一些 meta 信息或统计代码啥的
  - package.json， 项目配置文件
  - READED.md， 项目的说明文档，markdown 格式<br>手动换行测试<br>自动换行测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试k
  - ...
```

:::

Allow presentation a file directory（允许表示为文件目录）

The essence is "listtable" based on the addition of imitation directory style（本质是"列表格"的基础上增加仿目录样式）

::: tabs

@tab Plugin effect（插件效果）

[list2dt]

- vue-demo/
    - build/， 项目构建(webpack)相关代码
    - config/， 配置目录，包括端口号等。我们初学可以使用默认的
    - node_modules/， npm 加载的项目依赖模块
    - src/， 这里是我们要开发的目录
        - assets/， 放置一些图片，如logo等
        - components， 目录里面放了一个组件文件，可以不用
        - App.vue， 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录
        - main.js， 项目的核心文件。
    - static/， 静态资源目录，如图片、字体等
    - test/， 初始测试目录，可删除
    - .eslintignore
    - .gitignore， git配置
    - .index.html， 首页入口文件，你可以添加一些 meta 信息或统计代码啥的
    - package.json， 项目配置文件
    - READED.md， 项目的说明文档，markdown 格式
      手动换行测试
      自动换行测试k
    - ...

@tab No plugin effect（无插件效果）

(noPlugin)[list2dt]

- vue-demo/
  - build/， 项目构建(webpack)相关代码
  - config/， 配置目录，包括端口号等。我们初学可以使用默认的
  - node_modules/， npm 加载的项目依赖模块
  - src/， 这里是我们要开发的目录
    - assets/， 放置一些图片，如logo等
    - components， 目录里面放了一个组件文件，可以不用
    - App.vue， 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录
    - main.js， 项目的核心文件。
  - static/， 静态资源目录，如图片、字体等
  - test/， 初始测试目录，可删除
  - .eslintignore
  - .gitignore， git配置
  - .index.html， 首页入口文件，你可以添加一些 meta 信息或统计代码啥的
  - package.json， 项目配置文件
  - READED.md， 项目的说明文档，markdown 格式
手动换行测试
自动换行测试k
  - ...

@tab md source code (md源码)

```
[list2dt]

- vue-demo/
  - build/， 项目构建(webpack)相关代码
  - config/， 配置目录，包括端口号等。我们初学可以使用默认的
  - node_modules/， npm 加载的项目依赖模块
  - src/， 这里是我们要开发的目录
    - assets/， 放置一些图片，如logo等
    - components， 目录里面放了一个组件文件，可以不用
    - App.vue， 项目入口文件，我们也可以直接将组件写这里，而不使用 components 目录
    - main.js， 项目的核心文件。
  - static/， 静态资源目录，如图片、字体等
  - test/， 初始测试目录，可删除
  - .eslintignore
  - .gitignore， git配置
  - .index.html， 首页入口文件，你可以添加一些 meta 信息或统计代码啥的
  - package.json， 项目配置文件
  - READED.md， 项目的说明文档，markdown 格式<br>手动换行测试<br>自动换行测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试测试k
  - ...
```

:::

Ascii Tree

::: tabs

@tab Plugin effect (插件效果)

[list2astreeH|code()]

- vue-demo/
  - build/
  - config/
  - src/
    - assets/

            - a/
                - b

        - components
    - .babelrc
    - .editorconfig
    - ...

@tab No plugin effect (无插件效果)

(noPlugin)[list2astreeH|code()]

- vue-demo/
    - build/
    - config/
    - src/
        - assets/
            - a/
                - b

        - components
    - .babelrc
    - .editorconfig
    - ...

@tab md source code (md源码)

```
[list2astreeH|code()]

- vue-demo/
        - build/
        - config/
        - src/
                - assets/
                        - a/
                                - b
                - components
        - .babelrc
        - .editorconfig
        - ...
```

:::

WBS (Work Breakdown Structure, 工作分解结构)

::: tabs

@tab Plugin effect (插件效果)

[list2pumlWBS]

- vue-demo/
    - build/
    - config/
    - node_modules/
    - src/
        - < assets/
            - < a
                - b
                - < c

            - d
            - e
        - components
        - App.vue
        - main.js
    - static/
    - test/

@tab No plugin effect (无插件效果)

(noPlugin)[list2pumlWBS]

- vue-demo/
    - build/
    - config/
    - node_modules/
    - src/
        - < assets/
            - < a
                - b
                - < c
            - d
            - e
        - components
        - App.vue
        - main.js
    - static/
    - test/

@tab md source code (md源码)

```
[list2pumlWBS]

- vue-demo/
  - build/
  - config/
  - node_modules/
  - src/
    - < assets/
      - < a
        - b
        - < c
      - d
      - e
    - components
    - App.vue
    - main.js
  - static/
  - test/
```

:::

list2ut (列表转数据表格)

This table, I call a `DataTable` (这种表格我命名为 `数据表格`)

The feature is that cells are not allowed to cross rows or columns (特点是单元格不允许跨行或跨列)

> Allow filter sorting and other operations, but developing (允许筛选排序等操作，但该功能开发中)

描述一下你对下面各种水果的看法

::: tabs

@tab Plugin effect (插件效果)

[list2ut]

- < 水果
  - 颜色
  - 英文
- 苹果
  - 红色
  - apple
- 香蕉
  - 黄色
  - banana
- 橘子
  - 橙色
  - orange

@tab No plugin effect (无插件效果)

(noPlugin)[list2ut]

- < 水果
  - 颜色
  - 英文
- 苹果
  - 红色
  - apple
- 香蕉
  - 黄色
  - banana
- 橘子
  - 橙色
  - orange

@tab md source code (md源码)

```
[list2ut]

- < 水果
  - 颜色
  - 英文
- 苹果
  - 红色
  - apple
- 香蕉
  - 黄色
  - banana
- 橘子
  - 橙色
  - orange
```

:::

Another way to write it（另一种写法）

或者（*该功能未完善，等待后续开发*）

demo 另一种写法

::: tabs

@tab Plugin effect（插件效果）

[list2ut]

- 苹果
  - 颜色：红色
  - 英文：apple
- 香蕉
  - 颜色：黄色
  - 英文：banana
- 橘子
  - 颜色：橙色
  - 英文：orange

@tab No plugin effect（无插件效果）

(noPlugin)[list2ut]

- 苹果
  - 颜色：红色
  - 英文：apple
- 香蕉
  - 颜色：黄色
  - 英文：banana
- 橘子
  - 颜色：橙色
  - 英文：orange

@tab md source code（md源码）

```
[list2ut]

- 苹果
  - 颜色：红色
  - 英文：apple
- 香蕉
  - 颜色：黄色
  - 英文：banana
- 橘子
  - 颜色：橙色
  - 英文：orange
```

:::

Lists above level 2 are considered part of the content（二级以上的列表视为内容的一部分）

[list2mdut]

- 该模式下只能用二层列表01
  - 多了无效1
    - 多了无效11
    - 多了无效12

    - 多了无效2
        - 多了视为内容21
            - kll（这里有bug，没缩进）
        - 多了视为内容22
- 该模式下只能用二层列表02

List2tctable（列表转两列表格）

This table, I call a `Two-column table` （这种表格我命名为 `两列表格`）

The feature is that the table does not allow cross-row and cross-column, and only two columns（特点是表格不允许跨行和跨列，且只有两列）

timeline（时间线）

Can be used as a timeline, but developing（可以当时间线使用，但开发中）

@todo 这里样式可以再优化一下，目前看起来是和2ut的效果一样的

@attension 这里在渲染模式可能与table-extended插件冲突

::: tabs

@tab Plugin effect（插件效果）

[list2mdtimeline]

- 1840年6月
  英军发动鸦片战争
- 1842年8月
  清政府与英国签订《南京条约》：
  1. 中国割让香港岛给英国；
  2. 赔款洋银2100万元；
  3. 开放广州、厦门、福州、宁波、上海五处为通商口岸；

  > **Note**
  > 《南京条约》影响
  > 1. 中国近代史上第一个不平等条约，给中国人民带来深重的灾难，开创了列强以条约形式侵略和奴役中国的恶例；
  > 2. 中国的国家主权和领土完整遭到破坏，逐步沦为半殖民地半封建社会；
  > 3. 中国社会的主要矛盾由地主阶级与农民阶级的矛盾，演变为帝国主义和中华民族的矛盾（主要矛盾）、封建主义和人民大众的矛盾；
  > 4. 反侵略反封建成为中国人民肩负的两大历史任务；
  > 5. 中国逐渐开始了反帝反封建的资产阶级民主革命。

- 1841年5月
  三元里人民的抗英斗争，是中国近代史上中国人民第一次大规模的反侵略武装斗争。

- 1851年1月
  洪秀全金田村发动起义，建号太平天国。1853年3月，占领南京,定为首都,改名天京，正式宣告太平天国农民政权的建立。颁布《天朝天亩制度》、天平军北伐

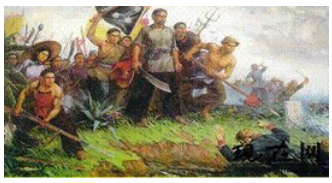
@tab No plugin effect（无插件效果）

(noPlugin)[list2mdtimeline]

- 1840年6月
  英军发动鸦片战争
- 1842年8月
  清政府与英国签订《南京条约》：
  1. 中国割让香港岛给英国；
  2. 赔款洋银2100万元；
  3. 开放广州、厦门、福州、宁波、上海五处为通商口岸；

  > **Note**
  > 《南京条约》影响
  > 1. 中国近代史上第一个不平等条约，给中国人民带来深重的灾难，开创了列强以条约形式侵略和奴役中国的恶例；
  > 2. 中国的国家主权和领土完整遭到破坏，逐步沦为半殖民地半封建社会；
  > 3. 中国社会的主要矛盾由地主阶级与农民阶级的矛盾，演变为帝国主义和中华民族的矛盾（主要矛盾）、封建主义和人民大众的矛盾；
  > 4. 反侵略反封建成为中国人民肩负的两大历史任务；
  > 5. 中国逐渐开始了反帝反封建的资产阶级民主革命。

- 1841年5月
  三元里人民的抗英斗争，是中国近代史上中国人民第一次大规模的反侵略武装斗争。

  > 

- 1851年1月
  洪秀全金田村发动起义，建号太平天国。1853年3月，占领南京,定为首都,改名天京，正式宣告太平天国农民政权的建立。颁布《天朝天亩制度》、天平军北伐

@tab md source code (md源码)

```
[list2mdtimeline]

- 1840年6月

  英军发动鸦片战争

- 1842年8月

  清政府与英国签订《南京条约》：
  1. 中国割让香港岛给英国；
  2. 赔款洋银2100万元；
  3. 开放广州、厦门、福州、宁波、上海五处为通商口岸；

  > [!note]
  > 《南京条约》影响
  >
  > 1. 中国近代史上第一个不平等条约，给中国人民带来深重的灾难，开创了列强以条约形式侵略和奴役中国的恶例；
  > 2. 中国的国家主权和领土完整遭到破坏，逐步沦为半殖民地半封建社会；
  > 3. 中国社会的主要矛盾由地主阶级与农民阶级的矛盾，演变为帝国主义和中华民族的矛盾（主要矛盾）、封建主义和人民大众的矛盾；
  > 4. 反侵略反封建成为中国人民肩负的两大历史任务；
  > 5. 中国逐渐开始了反帝反封建的资产阶级民主革命。

- 1841年5月

  三元里人民的抗英斗争，是中国近代史上中国人民第一次大规模的反侵略武装斗争。

  > ![ ](https://tse1-mm.cn.bing.net/th/id/R-C.4bbce1406f4442c1360edde26baa894d?rik=iHeUeby0jS5lnw&riu=http%3a%2f%2fp8.qhmsg.com%2fdr%2f270_500_%2ft01dbb76ff833d0a159.jpg&ehk=yggnC0t62%2b6DEVjvBgs%2fXJuuexBucd66FTc5gL0Gw%2fA%3d&risl=&pid=ImgRaw&r=0)

- 1851年1月
```

```
洪秀全金田村发动起义，建号太平天国。1853年3月，占领南京,定为首都,改名天京，正式宣告太平天国农民政权的建立。颁布《天朝天亩制度》、天平军北伐
```

:::

Tabs（标签页）

Can be used as a TAB page, but developing

可以点击标签栏切换

How to install python on each platform (python在各平台上的安装方法):

::: tabs

@tab Plugin effect (插件效果)

[list2tab]

- linux
  - 可以通过执行以下命令在终端中使用 apt 包安装程序:

    ```
    apt-get install python3.6
    ```

- windows
  - 转到官方 Python 站点，并导航到最新版本。在撰写本文时，即 `3.10.6`。
  - 下载适用于您平台的二进制文件。执行二进制。
  - 除了将 Python 添加到 `PATH` 之外，您不需要选择任何选项，因为默认安装程序具有您需要的一切。只需单击“安装”即可。
- macOS
  - 转到官方 Python 站点，并导航到最新版本。在撰写本文时，即 `3.10.6`。
  - 下载适用于您平台的二进制文件。执行二进制。
  - 在 Mac 上，这将默认在 dmg 安装程序中完成。

@tab No plugin effect (无插件效果)

(noPlugin)[list2tab]

- linux
  - 可以通过执行以下命令在终端中使用 apt 包安装程序:

    ```
    apt-get install python3.6
    ```

- windows
  - 转到官方 Python 站点，并导航到最新版本。在撰写本文时，即 `3.10.6`。
  - 下载适用于您平台的二进制文件。执行二进制。
  - 除了将 Python 添加到 `PATH` 之外，您不需要选择任何选项，因为默认安装程序具有您需要的一切。只需单击“安装”即可。
- macOS
  - 转到官方 Python 站点，并导航到最新版本。在撰写本文时，即 `3.10.6`。
  - 下载适用于您平台的二进制文件。执行二进制。
  - 在 Mac 上，这将默认在 dmg 安装程序中完成。

@tab md source code (md源码)

````
\[list2tab]

- linux
  - 可以通过执行以下命令在终端中使用 apt 包安装程序:
    ```shell
````
````

```
    apt-get install python3.6
    ```
- windows
  - 转到官方 Python 站点，并导航到最新版本。在撰写本文时，即 `3.10.6`。
  - 下载适用于您平台的二进制文件。执行二进制。
  - 除了将 Python 添加到 `PATH` 之外，您不需要选择任何选项，因为默认安装程序具有您需要的一切。只需单击“安装”即可。
- macOS
  - 转到官方 Python 站点，并导航到最新版本。在撰写本文时，即 `3.10.6`。
  - 下载适用于您平台的二进制文件。执行二进制。
  - 在 Mac 上，这将默认在 dmg 安装程序中完成。
```

:::

## Card（卡片）

::: tabs

@tab Plugin effect（插件效果）

[list2card|addClass(ab-col3)]

- card1
  card1_item
  $1+1=2$
- card2
  card2_item

  ```
  var a = 1
  ```

- card3
  card3_item
  Bold *italics* **highlight** ~~delete~~
  - list1
  - list2
  - list3

@tab No plugin effect（无插件效果）

(noplugin)[list2card|addClass(ab-col3)]

- card1
  card1_item
  $1+1=2$
- card2
  card2_item

  ```
  var a = 1
  ```

- card3
  card3_item
  Bold *italics* **highlight** ~~delete~~
  - list1
  - list2
  - list3

@tab md source code（md源码）

```
\[list2card|addClass(ab-col3)]

- card1
  card1_item<br>$1+1=2$
- card2
```
```

```
  card2_item
  ```js
  var a = 1
  ```
- card3
  card3_item
  Bold *italics* ==highlight== ~~delete~~
  - list1
  - list2
  - list3
```

:::

list2pic 转图像

to flow（转mermaid流程图）

（补充：其本质是转化为 `graph TB` 语法.所以除了常规操作，你还可以进行一些其他操作：
例如指定别名、进行树结构以外的连接）

demo：描述一下树设计的脑图

::: tabs

@tab Plugin effect（插件效果）

[list2mermaid]

- 树结构
  - 基本术语
    - A
    - B(BB)
    - C(CC)
      - A
  - 性质
  - 基本运算
  - 二叉树
    - 分支1
    - 分支2

@tab No plugin effect（无插件效果）

(noPlugin)[list2mermaid]

- 树结构
  - 基本术语
    - A
    - B(BB)
    - C(CC)
      - A
  - 性质
  - 基本运算
  - 二叉树
    - 分支1
    - 分支2

@tab md source code（md源码）

```
[list2mermaid]

- 树结构
  - 基本术语
    - A
    - B(BB)
    - C(CC)
      - A
  - 性质
  - 基本运算
  - 二叉树
    - 分支1
    - 分支2
```

:::

to pumlMindmap (转plantuml思维导图)

推荐轻量级使用（不内嵌md）

::: tabs

@tab Plugin effect (插件效果)

[list2pumlMindmap]

- vue-demo/
  - build/
  - config/
  - node_modules/
  - src/
    - assets/
      - a/
        - b
    - components
    - App.vue
    - main.js
  - static/
  - test/

@tab No plugin effect (无插件效果)

(noPlugin)[list2pumlMindmap]

- vue-demo/
  - build/
  - config/
  - node_modules/
  - src/
    - assets/
      - a/
        - b
    - components
    - App.vue
    - main.js
  - static/

    - test/

@tab md source code (md源码)

```
[list2pumlMindmap]

- vue-demo/
        - build/
        - config/
        - node_modules/
        - src/
                - assets/
                        - a/
                                - b
                - components
                - App.vue
                - main.js
        - static/
        - test/
```

:::

to nodes (转节点树图, AnyBlock版思维导图)

::: tabs

@tab Plugin effect (插件效果)

[list2node]

- Links
  - Website
  - GitHub
- Related Projects
  - coc-markmap for Neovim
  - markmap-vscode for VSCode
  - eaf-markmap for Emacs
- Features
  - Lists
    - strong ~~del~~ *italic* **highlight**
    - `inline code`
    - [x] ~~checkbox~~
    - Katex: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
    - Katex_mutilLine:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

    - More Katex Examples
    - > **Note**
      > Now we can wrap very very very very long text based on `maxWidth` option

  - Blocks
    - ```
      console.log("hello, JavaScript")
      ```
Products	Price
Apple	4
      ```

Products	Price
Banana	2

- 

@tab No plugin effect (无插件效果)

(noPlugin)[list2node]

- Links
  - Website
  - GitHub
- Related Projects
  - coc-markmap for Neovim
  - markmap-vscode for VSCode
  - eaf-markmap for Emacs
- Features
  - Lists
    - strong ~~del~~ *italic* **highlight**
    - `inline code`
    - [x] ~~checkbox~~
    - Katex: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
    - Katex_mutilLine:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

    - More Katex Examples
    - > **Note**
      > Now we can wrap very very very very long text based on `maxWidth` option

  - Blocks
    - ```
      console.log("hello, JavaScript")
      ```
Products	Price
Apple	4
Banana	2

    - 

@tab md source code (md源码)

```
\\[list2node]

- Links
  - [Website](https://markmap.js.org/)
```
```

```
  - [GitHub](https://github.com/gera2ld/markmap)
- Related Projects
  - [coc-markmap](https://github.com/gera2ld/coc-markmap) for Neovim
  - [markmap-vscode](https://marketplace.visualstudio.com/items?itemName=gera2ld.markmap-vscode) for VSCode
  - [eaf-markmap](https://github.com/emacs-eaf/eaf-markmap) for Emacs
- Features
  - Lists
    - **strong** ~~del~~ *italic* ==highlight==
    - `inline code`
    - [x] checkbox
    - Katex: $x = {-b \pm \sqrt{b^2-4ac} \over 2a}$
    - Katex_mutilLine:
      $$
      x = {-b \pm \sqrt{b^2-4ac} \over 2a}
      $$
    - [More Katex Examples](#?d=gist:af76a4c245b302206b16aec503dbe07b:katex.md)
    - > [!note]
      > Now we can wrap very very very very long text based on `maxWidth` option
  -
    ```javascript
    console.log("hello, JavaScript")
    ```
Products	Price
Apple	4
Banana	2
  -
```

:::

to markmap（转markmap思维导图）

推荐需要内嵌md时使用

::: tabs

@tab Plugin effect（插件效果）

[list2markmap]

- Links
  - Website
  - GitHub
- Related Projects
  - coc-markmap for Neovim
  - markmap-vscode for VSCode
  - eaf-markmap for Emacs
- Features
  - Lists
    - strong ~~del~~ *italic* **highlight**
    - `inline code`
    - [x] ~~checkbox~~
    - Katex: $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ <!-- markmap: fold -->
    - More Katex Examples
    - Now we can wrap very very very very long text based on `maxWidth` option
  - Blocks
  -
    ```javascript
    console.log("hello, JavaScript")
    ```
  -

Products	Price
Apple	4

Products	Price
Banana	2

- 

@tab No plugin effect (无插件效果)

(noPlugin)[list2markmap]

- Links
  - Website
  - GitHub
- Related Projects
  - coc-markmap for Neovim
  - markmap-vscode for VSCode
  - eaf-markmap for Emacs
- Features
  - Lists
    - strong ~~del~~ *italic* ==highlight==
    - `inline code`
    - [x] ~~checkbox~~
    - Katex: $x = {-b \pm \sqrt{b^2-4ac} \over 2a}$ <!-- markmap: fold -->
    - [More Katex Examples](#?d=gist:af76a4c245b302206b16aec503dbe07b:katex.md)
    - Now we can wrap very very very very long text based on `maxWidth` option
  - Blocks
  - ```
    console.log("hello, JavaScript")
    ```
Products	Price
Apple	4
Banana	2
  - 

@tab md source code (md源码)

```
[list2markmap]

- Links
  - [Website](https://markmap.js.org/)
  - [GitHub](https://github.com/gera2ld/markmap)
- Related Projects
  - [coc-markmap](https://github.com/gera2ld/coc-markmap) for Neovim
  - [markmap-vscode](https://marketplace.visualstudio.com/items?itemName=gera2ld.markmap-vscode) for VSCode
  - [eaf-markmap](https://github.com/emacs-eaf/eaf-markmap) for Emacs
- Features
  - Lists
    - **strong** ~~del~~ *italic* ==highlight==
    - `inline code`
    - [x] checkbox
```
```

```
    - Katex: \$x = {-b \pm \sqrt{b^2-4ac} \over 2a}$ <\!-- markmap: fold -->
    - \[More Katex Examples](#?d=gist:af76a4c245b302206b16aec503dbe07b:katex.md)
    - Now we can wrap very very very very very long text based on `maxWidth` option
  - Blocks
  -
    ```javascript
    console.log("hello, JavaScript")
    ```
Products	Price
Apple	4
Banana	2
  -
```

:::

to mermaid mindmap (转mermaid思维导图)

(由于ob的mermaid版本较低，没有mindmap，所以这里插件内置了一个新的mermaid)
(当然缺点是：插件大小从200KB变为了9MB多，等到ob更新mermaid版本我会将插件内置的那份给去除掉的)

很多符号限制，而且样式我觉得一般。这是旧版anyblock中所使用到的思维导图，最新版本个人不再推荐使用

这里我就直接使用mermaid官方给的例子了：

::: tabs

@tab Plugin effect (插件效果)

[listroot(root((mindmap)))|list2mindmap]

- Origins
  - Long history
  - ::icon(fa fa-book)
  - Popularisation
    - British popular psychology author Tony Buzan

- Research
  - On effectiveness
    and features
  - On Automatic creation
    - Uses
      - Creative techniques
      - Strategic planning
      - Argument mapping

- Tools
  - Pen and paper
  - Mermaid

@tab No plugin effect (无插件效果)

(noPlugin)[listroot(root((mindmap)))|list2mindmap]

- Origins
  - Long history
  - ::icon(fa fa-book)
  - Popularisation
    - British popular psychology author Tony Buzan

- Research

    - On effectiveness
      and features
    - On Automatic creation
        - Uses
            - Creative techniques
            - Strategic planning
            - Argument mapping

- Tools
    - Pen and paper
    - Mermaid

@tab md source code (md源码)

```
[listroot(root((mindmap)))|list2mindmap]

- Origins
  - Long history
  - ::icon(fa fa-book)
  - Popularisation
    - British popular psychology author Tony Buzan
- Research
  - On effectiveness<br/>and features
  - On Automatic creation
    - Uses
      - Creative techniques
      - Strategic planning
      - Argument mapping
- Tools
  - Pen and paper
  - Mermaid
```

:::

Other (其他)

加列表根

如果列表有root，我们可以写列表时将root省略掉，在头部信息中加上。
在转流程图和思维导图时该技巧很好用

例如:

::: tabs

@tab Plugin effect (插件效果)

[listroot(树结构)]

- 基本术语
    - A
    - B
    - C
- 性质
- 基本运算
- 二叉树

@tab No plugin effect (无插件效果)

(noPlugin)[listroot(树结构)]

- 基本术语

    - A
    - B
    - C
- 性质
- 基本运算
- 二叉树

@tab md source code (md源码)

```
\[listroot(树结构)]

- 基本术语
  - A
  - B
  - C
- 性质
- 基本运算
- 二叉树
```

:::

> json

[X|json2pumlJson]

```
{
  "a": 10,
  "b": {
    "b1": 11,
    "b2": 12
  }
}
```

More (更多)

::: tabs

@tab en

- There are dozens of different processors, not given here, that can be self-explored:
- Older documentation for V2 contains more information about the processor:
  https://linczero.github.io/MdNote_Public/ProductDoc/AnyBlock/v2%20old%20docs/
- You can install 'Any Block' in Ob and open the plug-in's Settings panel to see all supported processors
  (Except for the new processor and mdit instructions that are not added in the new version, the instructions in the old version are actually more complete)
- After installation in Obsidian, you can also view all supported processors through the '[info]' processor

@tab zh

- 有几十个不同的处理器，这里没有给出，可以自探索:
- V2的旧文档包含更多处理器的介绍： https://linczero.github.io/MdNote_Public/ProductDoc/AnyBlock/v2%20old%20docs/
- 可以在Ob中安装 `Any Block` 并打开该插件的设置面板，以查看所有支持的处理器
  （除了没有新版本增加的新处理器和mdit说明外，旧版文档的说明其实会更全）
- 在Obsidian中安装以后，也可以通过 `[info]` 处理器，以查看所有支持的处理器

:::

Selector (选择器)

::: tabs

@tab en

This part is very important! Recommended to finish!

url: Selector

@tab zh

这部分内容非常重要！推荐看完！

链接：Selector

:::