需求

1.多语言切换：能够在至少两种语言之间进行切换，展示不同语言版本的内容。2.数学公式支持：内容中包含数学公式，且在语言切换后，数学公式能正确渲染和显示。

3.离线可用：通过本地化部署 MathJax，确保在没有网络连接的情况下，组件的所有功能（包括语言切换和数学公式渲染）都能正常工作。

实现思路

1.HTML 结构：创建一个包含按钮和内容展示区域的 HTML 页面。按钮用于触发语言切换操作，内容展示区域用于呈现不同语言版本的文本和数学公式。同时，在 HTML 中引入本地化的 MathJax 脚本文件，并配置其基本参数，如公式的分隔符等。

2.CSS 样式：为按钮和内容展示区域设置基本的样式，如按钮的背景颜色、字体颜色，内容区域的边距和字体样式等，以提供简洁美观的视觉效果。

3.JavaScript 逻辑：

◦定义一个对象来存储不同语言的文本内容，包括标题、描述和数学公式。

◦获取页面中的按钮和内容展示区域的 DOM 元素。

◦为按钮添加点击事件监听器，当按钮被点击时，根据当前语言切换到另一种语言，并更新内容展示区域的 innerHTML。

◦在更新内容后，手动调用 MathJax 的渲染方法，确保新的数学公式能够正确渲染。同时，通过MathJax.startup.promise确保在 MathJax 完全加载和启动后再执行语言切换相关操作。

**优点**

**现阶段虽然有些浏览器能给提供翻译功能，但是由于翻译的不准确性和对某些内容无法翻译，从而带来阅读上的障碍，通过内置语言选项按钮，能够实现全文的语言翻译，从而达到不管是国内还是国外的浏览者，都能有很好的阅读体验，犹如阅读本国的网站**

1.独立性强：由于本地化部署了 MathJax，无需依赖外部 CDN，增强了组件在离线环境或网络不稳定情况下的可用性。

2.可维护性高：代码结构清晰，HTML 负责页面结构，CSS 负责样式，JavaScript 负责交互逻辑，各个部分职责明确，便于后续的维护和扩展。例如，若要添加新的语言，只需在 JavaScript 定义的语言文本对象中增加相应的语言项即可。

3.灵活性好：对 MathJax 的配置更加灵活，可根据项目需求定制渲染参数，如公式的分隔符、字体缓存模式等，以满足不同场景下的数学公式展示需求。