

项目技术说明文档：深海仿生概念网站

1. 技术栈与核心插件概览

本项目采用 **原生 HTML5 + CSS3 + Vanilla JavaScript** 作为核心基础，以确保最高的定制化自由度。为了实现极致的视觉体验和高级交互，引入了以下核心第三方插件库：

插件名称	版本 / 文件	应用场景与功能描述
Lenis	<code>lenis.min.js</code>	全局平滑滚动： 替代浏览器生硬的原生滚动，提供具有物理惯性的丝滑滚动体验，特别是在“生态共生”和“首页视差”等长图文页面中大幅提升了浏览的高级感。
MathJax	<code>tex-svg.js</code> (v3)	数学公式渲染： 应用于数理秩序页面（ <code>formula.html</code> ）。将复杂的 LaTeX 宏包代码（如流体力学、能斯特-普朗克方程）实时渲染为高精度的 SVG 矢量图形。
Swiper.js	<code>swiper- bundle.min.js</code>	3D 封面流轮播： 应用于团队介绍页面（ <code>team.html</code> ）。构建了具有深度透视感的 3D 卡片阵列，用于展示 10 名团队成员，并支持丝滑的拖拽与分组跳转。

2. 遇到的核心问题与解决方案

在开发多页面、重视觉交互的网站时，我们遇到了几个典型的技术阻碍，以下是解决过程的详细记录：

问题一：全局 CSS 污染导致 MathJax 公式重叠错位

- 问题描述：** 在编写公式展示页时，全局样式表中设置的 `font-family`（艺术字体）、`letter-spacing`（字间距）和 `line-height`（行高）强行作用到了 MathJax 生成的 DOM 元素上，导致复杂的数学公式符号互相挤压、重叠，无法阅读。
- 解决过程：** 1. 初步尝试通过 CSS `!important` 强制重置数学容器的字体属性，但由于浏览器字体渲染引擎的差异，依然存在像素级的错位。2. **最终方案（底层替换）：** 放弃传统的 HTML-CHTML 渲染模式，将 MathJax 的引入脚本更换为 **SVG 模式**（`tex-svg.js`）。SVG 是纯粹的矢量路径图形，完全免疫文本相关的 CSS 属性。修改后，复杂的积分符号和上下标变成了固定的矢量坐标，公式渲染变得极其清晰锐利，彻底根除了排版重叠问题。

问题二：网格画廊图片比例失调与裁剪

- 问题描述：** 细胞美学页面的网格画廊中，不同尺寸的原始图片在放入统一的 Grid 网格时，会被强制拉伸或裁剪。此外，带有白色背景的原始图片与网站的“玻璃拟态”透明背景显得格格不入。
- 解决方案：**

1. **布局约束**: 强制设定固定的高度 (如 `height: 280px`) , 确保所有卡片等高, 规避瀑布流的错乱。
2. **视觉融合**: 对图片使用 `object-fit: contain` 确保完整展现; 最关键的一步是引入了 `mix-blend-mode: multiply` (**正片叠底**) 混合模式。这使得白底图片中的白色部分被自动过滤, 直接透出底部的玻璃卡片背景, 实现了类似“显微镜载玻片标本”的高级融合感。

问题三：长页面导航栏状态不同步

- **问题描述**: 在生态共生页面中, 设计了一个仿生神经索侧边栏。我们需要当用户滚动页面时, 侧边栏能够自动展开对应的子菜单, 且“发光神经索”的高度要实时精准跟随。
- **解决方案**: 弃用性能低下的 `window.onscroll` 监听, 改用现代浏览器原生 API `IntersectionObserver`。
 1. 将阈值设定为视口中部 (`rootMargin: '-20% 0px -60% 0px'`), 精准探测哪个内容区块进入了视野。
 2. 触发回调后, 使用 JavaScript 动态移除其他菜单的激活类, 实现“手风琴式”的自动收纳。
 3. 动态计算当前激活节点 (如水滴形角标) 相对于容器的 Y 轴偏移量, 并实时赋予进度条的高度属性, 实现物理联动的视觉反馈。

问题四：团队介绍卡片的机械感过重

- **问题描述**: 团队介绍页最初的 Swiper 排版过于规矩, 方形头像框和亮条分割线让页面看起来像传统的企业后台, 违背了项目“深海生物学”的有机主题。
- **解决方案**: 进行“有机化改造”:
 1. **呼吸形变头像**: 摈弃常规的圆形倒角, 利用复杂的百分比圆角 (如 `border-radius: 40% 60% 70% 30% / 40% 50% 60% 50%`) 结合 `@keyframes` 动画, 让头像框像水母的伞盖一样进行 8 秒周期的缓慢“蠕动”变形。
 2. **分组联动**: 在顶部增加了圆润的胶囊形过滤按钮, 并在 JS 中双向绑定了 Swiper 的 API。点击按钮时触发 `swiper.slideTo()` 跳转到对应组别; 当用户手动滑动卡片触发 `slideChange` 事件时, 反向计算当前索引, 高亮顶部的对应组别按钮。

3. 核心视觉规范

本项目在样式上高度统一, 提炼出了几项可复用的 CSS 设计模式:

- **Glassmorphism (玻璃拟态)**: 大量运用半透明白色背景 (`rgba(255, 255, 255, 0.6)`) 配合强效背景模糊 (`backdrop-filter: blur(20px)`) , 在深色动态背景 (如水母和孢子) 前营造通透的呼吸感和微观感。
- **Bioluminescence (生物荧光)**: 摈弃生硬的实体边框, 转而使用低透明度的 `box-shadow` 和径向渐变 (`radial-gradient`) 来模拟深海发光生物的光晕。色板严格限制在莫兰迪蓝 (`#8ec5fc`) 、紫 (`#c490e4`) 、粉 (`#f8cdda`) 等梦幻色系区间。
- **GPU 硬件加速**: 对于悬浮、放大、平移等高频动画, 优先使用 `transform: translate3d()` 或 `scale()` , 避免触发浏览器的重排, 保证在移动端也能维持丝滑体验。

