**JavaScript/TypeScript 技术管理与代码规范**

**2022141461099 赵智翔**

a. 强制（必须遵守）

1. 每个文件必须声明严格模式（'use strict';，TypeScript 默认严格）。
2. 变量声明必须使用 const 或 let，禁止使用 var。
3. 禁止未使用的变量和函数。
4. 禁止直接修改函数参数。
5. 禁止在同一作用域内重复声明变量。
6. 必须使用全等（===）和不全等（!==），禁止使用 == 和 !=。
7. 禁止在循环体内声明函数。
8. 禁止使用 eval、with、alert。
9. 禁止直接操作 DOM，除非在框架允许的生命周期内。
10. 必须为所有导出的函数、类、组件写 JSDoc/TS 注释。
11. 异步操作必须使用 async/await 或 Promise，禁止回调地狱。
12. 必须使用 ESLint/TSLint 进行代码静态检查，CI 阶段强制通过。
13. 所有新代码必须有单元测试，覆盖率不得低于 80%。
14. 禁止在生产代码中出现 console.log、debugger。
15. 必须使用 Prettier 或等价工具统一代码风格。
16. 禁止魔法数字，必须用常量或枚举代替。
17. 禁止直接在模板/JSX 中拼接 HTML，防止 XSS。
18. 必须使用模块化（ESM 或 CommonJS），禁止全局变量污染。
19. 必须在 catch 语句中处理异常，禁止空 catch。
20. 禁止在循环中使用 await，应使用 Promise.all 并发。

b. 推荐（建议遵守）

1. 每个文件只包含一个主要导出（类/组件/函数）。
2. 函数长度不超过 40 行，类不超过 300 行。
3. 组件/函数参数不超过 4 个，超出请用对象传参。
4. 优先使用解构赋值。
5. 优先使用箭头函数，除非有 this 绑定需求。
6. 类型定义优先使用 TypeScript，JS 项目建议用 JSDoc。
7. 优先使用不可变数据结构（如 Object.freeze、readonly）。
8. 优先使用模板字符串，避免字符串拼接。
9. 优先使用命名导出，避免默认导出。
10. 优先使用函数式编程风格，减少副作用。
11. 优先使用 ESLint 推荐规则和 Prettier 默认格式。
12. 优先使用 async/await 处理异步，避免 then 链。
13. 优先使用 const 定义变量，只有需要变更时才用 let。
14. 优先使用解构 props/state，避免 this.props.xxx。
15. 优先使用现代 API（如 fetch、Map、Set），避免老旧 API。
16. 优先使用国际化（i18n）方案处理文本。
17. 优先使用响应式/声明式编程（如 Vue/React 的响应式特性）。
18. 优先使用自动化工具（如 Husky、lint-staged）保证提交质量。
19. 优先在 PR 中进行代码评审，避免直接合并主分支。
20. 优先在项目根目录维护统一的 README、CHANGELOG、CONTRIBUTING 文档。

c. 允许（可选）

1. 允许在开发环境下使用 console.log，但需在提交前移除。
2. 允许在特殊场景下使用 any，但必须注明原因。
3. 允许在小型工具函数中省略类型注释，但需保证可读性。
4. 允许在性能敏感场景下使用 for 循环替代高阶函数。
5. 允许在 demo 或实验性分支中适当放宽 lint 规则。
6. 允许在必要时使用第三方库，但需评估安全性和维护性。
7. 允许在 UI 组件中使用内联样式，但建议优先用 CSS/SCSS。
8. 允许在单元测试中 mock 全局对象。
9. 允许在项目初期阶段适当降低测试覆盖率，但需逐步提升。
10. 允许在文档、注释中使用中文，但代码和命名必须为英文。