

ERP系统前端开发规范

编码

HTML

- `tab`键用 4 个空格代替
- `html` 标签

1. `id`, `class` 等以 ``-`` 分隔, 全部小写
2. 必要时子标签加父标签前缀

- 注释 : `<!--注释-->`
-

CSS

- 选择器 声明独占一行
 - 属性 定义另起一行, 以分号结尾
-

JS

- 变量

1. 驼峰命名法, 即首词小写, 后每词首字母大写
2. 如变量是`bool` 类型, 加前缀 `is/has` 等

- 常量 : 全大写命名, 各词间以 `_` 分隔
- 枚举 : 同上
- 函数

1. 驼峰命名
2. 如返回值是 `bool` , 加前缀 `is/has` 等
3. 函数任务单一原则 , 函数过长时<超过80行>, 考虑拆为多个函数
4. 参数: 驼峰命名

- 类 : 驼峰命名, 首字母大写

- **注释** : 每个js 页面必须注释功能用途, 有复杂逻辑时, 有必要对函数添加注释

```
//函数注释
/**
 * @desc    函数描述
 * @param   参数说明
 * @param
 */

//文件 注释
/**
 *
 * 描述功能
 */
```

其它

- 文件命名 : 下划线命名, 全部小写
- 图片命名 : 同上
- 代码格式化插件 HTML/CSS/JS Prettify

GIT

基本操作

- `git branch` 查看本地分支
- `git branch -a` 查看所有分支
- `git branch -d` 删除本地分支
- `git checkout` 切换分支
- `git branch name` 新建分支
- `git status` 查看分支状态

关于项目GIT流程

- 提交本地代码

```
git add .
git commit -m " "
```

- 获取远程代码

```
git pull --rebase
```

- 若有冲突

1. 解决冲突
2. `git add .`
3. `git rebase --continue`

确保在本地解决冲突以后且页面正常再 `Push`

- `git push origin name` 将本地分支推送到指定远程分支上

将远程分支拉到本地

- `git branch name` 创建dev分支
- `git pull origin dev` 将远程分支dev 拉取到本地

其它

- `git config --global alias.co checkout` 修改命令
- `git log` 查看提交历史
- `git show HEAD~3` 查看往前3次的提交
- `git reset --hard 哈希值` 回退到指定版本
- `git checkout -- filename` 取消对文件的修改
- `git branch -d` 删除本地分支

项目部署

- webpack
- `./deploy.sh`

开启本地环境

- webpack-dev-server
- `node --harmony app.js`

新接入此项目

- `npm install`
- 启动webpack-dev-server , node
- `localhost:8080/index.html`