# License Document Draft

## Dot

继承 License 有 不 可 逆 / 不 可 篡改 基本 特征

```plain

1. ^ 萌铺子（杭州）科技有限公司 遵循 先立后破 不立不破 原则

2. ^ 约定 定义 省部 (行政划分) 发起 经济区域 (区域经济) 有 轮值 链长 城市 / 成员 城市 轮值 链长 城市 职责 是 轮值期 负责 指导 / 监督 各 成员 城市 全面 工作 成员 城市 对 轮值 链长 城市 负责 轮值 链长 城市 因 省部 产生 对 省部 负责 允许 依 法定 程序 直接 指导 / 监督 成员 城市 各 社区 全面 工作 遵循 共同 参与 / 共同 协商 / 共同 治理 / 共同 建设 / 共同 分享 原则 依 法定 程序 针对 一切 / 所有 / 衍生 背离 经济区域协调发展 战略 言论 / 行为 追责 / 处罚 / 清理 零 容忍

3. ^ 最小可行性研究及试验发展 (Minimum Viable Research and Development) 有 最小可行性验证 (Minimum Viable Proof) / 最小可行性产品 (Minimum Viable Product) 最小可行性验证 因 企业实验室 法定代表人 授权 发布 产生 对 企业实验室 法定代表人 授权 发布 负责 最小可行性产品 因 企业运营室 法定代表人 授权 发布 产生 对 企业运营室 法定代表人 授权 发布 负责

4. ^ 企业 内部 遵循 最小 派出 组织 发展 原则 就 是 说 必要 保留 / 适度 超前 / 坚决 精简

5. ^ 企业 会议 遵循 全过程人民民主 原则 有 股东代表大会 / 职工代表大会 [6]

6. ^ 股东代表大会 职责 是 接收 职工代表大会 职工代表 (高级管理人员) 报告 / 由 股东代表大会 股东代表 (法定代表人) 依 法定 程序 办理 事项 且 记录 / 存档 / 修订 / 审议 章程 因 股东代表大会 股东 表决权 产生 对 股东代表大会 股东 表决权 负责 股东代表大会 因 中华人民共和国公司法 产生 对 中华人民共和国公司法 负责 [7]

7. ^ 职工代表大会 职责 是 股东代表大会 股东代表 (法定代表人) 听取 职工代表大会 职工代表 (高级管理人员) 意见 / 建议 由 职工代表大会 职工代表 (高级管理人员) 依 法定 程序 整理 需 进一步 办理 事项 向 股东代表大会 股东代表 (法定代表人) 提交 报告 因 股东代表大会 产生 对 股东代表大会 负责 属于 企业 内部 群众 自治 组织 自愿 接受 各级 党委 及 工会 领导

6. ^ 以 中国共产党 中央 文献 -> 行政 历史 档案 为 基础 全面 整理 各级 地方 文献 -> 行政 历史 档案 确保 公开级 行政 历史 档案 原子性 一致

7. ^ 企业 资金 方向 有 企业 资金 / 自有 资金 企业 资金 追踪 有 股东 实缴 / 业务 投资 / 工薪 (职工 薪资) 代发

8. ^ Serving legal representatives of startups in their ultra-early phase / Services for legal representatives of ultra-early-phase startups [9]

9. ^ Promote the comprehensive development of business models and legal representatives for enterprises of all ownership types / Pursuing both with equal rigor and resolve

10. ^ 像 做 广告 / 媒体 / 新闻 / 文章 一样 做 教育 及 培训 让 教育 思想 客观 落地 / 促进 教育 文化 及 企业 文化 深度 融合 开辟 教育学 新 篇章

```

## Part 0

### Point 0.0.2

```plain

国家税务总局杭州市拱墅区税务局

责令限期改正通知书 [10]

杭拱墅税 限改 [2025] 22293 号

萌铺子（杭州）科技有限公司（纳税人识别号 91330105MAEAA1M33D）

你（单位）2025-02-01 至 2025-02-28 个人所得税（工资薪金所得）未 按期 进行 申报 根据 中华人民共和国税收征收管理法 (62) 规定 限 你（单位）于 2025-04-17 前 携带 相关 资料 至 我 局 申报 办理 有关 事项

2025-03-27

```

### Point 0.1.2

```plain

首违不罚承诺书 [10]

本 单位（个人）（名称）萌铺子（杭州）科技有限公司 纳税人识别号（身份证号码）91330105MAEAA1M33D 2025-02-01 至 2025-02-28 个人所得税 未 按期 进行 申报 违反 了 税收 法律 法规 的 相关 规定 因 系 初次 违法 危害 后果 轻微 并 及时 改正 根据 中华人民共和国行政处罚法 (33) 规定 税务 机关 已 对 该 违法 事项 不 予 处罚

本 单位（个人）承诺 如下

1. 本 单位（个人）已 知晓 税务 行政 处罚 首违不罚 事项 清单 及 相关 规定 将 履行 相关 义务

2. 本 单位（个人）将 加强 税收 相关 法律 法规 的 学习 不 再 违反 该 事项 的 相关 规定

3. 本 单位（个人）办理 首违不罚 事项 后 如 因 相同 事项 再次 违反 相关 规定 将 依 法 按照 规定 接受 相应 行政 处罚

4. 本 单位（个人）将 依 法 诚信 纳税 积极 履行 纳税人（扣缴义务人）义务

5. 上 述 承诺 是 本 单位（个人）的 真实 意思 表示

承诺人 萌铺子（杭州）科技有限公司

2025-03-30

```

### Point 0.2.2

```plain

国家税务总局杭州市拱墅区税务局

不予税务行政处罚决定书 [10]

杭拱墅税不罚 [2025] 3855 号

萌铺子（杭州）科技有限公司（纳税人识别号 91330105MAEAA1M33D）

经 我 局（所）对 你（单位）未 按 规定 的 期限 向 主管 税务 机关 办理 纳税 申报 及 报送 纳税 资料 的 情况 进行 检查 核实 你 单位 存在 以 下 违法 事实 2025-02-01 至 2025-02-28 个人所得税 未 按期 进行 申报 上 述 事实 主要 有 以 下 证据 证明

1. 责令限期改正通知书（杭拱墅税 限改 [2025] 22293 号） 及 送达回证

2. 金税核心征管系统纳税申报查询结果

上 述 行为 违反 中华人民共和国税收征收管理法 (62) 规定 鉴 于 上 述 税收 违法 行为 初次 违法 且 危害 后果 轻微 并 及时 改正 按照 中华人民共和国行政处罚法 (33) 规定 现 决定 不 予 行政 处罚

如 对 本 决定 不 服 可以 自 收到 本 决定书 之 日 起 60 天 内 依 法 向 国家税务总局杭州市税务局 申请 行政 复议 或者 自 收到 本 决定书 之 日 起 6 月 内 依 法 向 人民法院 起诉

税务 机关（印章）

2025-03-30

```

### Point 0.3.2

framework\_base\_02.html

```plain

<!--

以下是将装饰器系统整合到MVP框架后的完整代码实现：

主要改进点说明：

1. 装饰器翻译系统

- 通过 `registerServices` 实现 装饰器工厂清单

- 通过 `registerService` 实现 装饰器工厂方法

- 通过 `withDecorators` 实现 装饰器工厂流水线

- 服务间支持自动依赖解析（如UserService自动获取StorageService）

- Presenter构造函数自动注入已注册服务

2. 服务整合

- StorageService提供本地存储功能

- UserService处理用户相关业务逻辑

- 服务实例在应用启动前自动初始化

3. Presenter增强

- 同时支持原有MVP参数(view/model)和依赖注入

- 业务方法中可直接使用注入的服务

- 保持原有事件处理逻辑不变

4. 数据持久化示例

- 用户提交时自动保存到localStorage

- 商品保存时记录时间戳

- 可通过StorageService扩展更多存储方式

使用示例：

1. 提交用户表单后可在控制台查看存储结果

2. 保存商品后可在localStorage查看"lastProduct"数据

3. 所有服务依赖自动解析，无需手动实例化

该实现保持了MVP模式的核心优势，同时增加了现代化的依赖注入和服务管理能力，适合需要扩展复杂业务逻辑的中大型应用。

-->

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<style>

[data-id] { margin: 5px; padding: 5px; }

.container { border: 1px solid #ccc; padding: 20px; max-width: 500px; }

.error { color: red; display: none; }

</style>

</head>

<body>

<!-- 组件1 -->

<div class="container" data-component="userForm" data-model="UserModel">

<input type="text" placeholder="Enter username" data-id="username" data-bind="model.username">

<button data-id="submitBtn" data-event="click:handleSubmit">Submit</button>

<div data-id="usernameError" class="error"></div>

<div data-id="output"></div>

</div>

<!-- 组件2 -->

<div class="container" data-component="productForm" data-model="ProductModel">

<input type="text" placeholder="Enter productName" data-id="productName" data-bind="model.name">

<button data-event="click:handleSave">Save Product</button>

</div>

<script>

// ====================================

// Model 类 - 数据管理

// 基础模型类

// ====================================

class Model {

constructor() {

this.\_data = {};

this.listeners = [];

}

get data() {

return this.\_data;

}

setData(key, value) {

this.\_data[key] = value;

this.notify(key);

}

subscribe(callback) {

this.listeners.push(callback);

}

notify(key) {

this.listeners.forEach(cb => cb(key));

}

}

// 用户模型

class UserModel extends Model {

constructor() {

super();

this.setData('username', '');

}

validateUsername(value) {

if (value.length < 3) throw new Error("用户名至少3个字符");

}

}

// 商品模型

class ProductModel extends Model {

constructor() {

super();

this.setData('name', '');

}

}

// ====================================

// View 类 - UI管理

// ====================================

class View {

constructor(dataComponent) {

this.component = document.querySelector(`[data-component="${dataComponent}"]`);

this.elements = {};

this.bindings = {};

// 自动发现元素

this.component.querySelectorAll('[data-id]').forEach(el => {

this.elements[el.dataset.id] = el;

});

// 处理数据绑定

this.component.querySelectorAll('[data-bind]').forEach(el => {

const binding = el.dataset.bind;

const [\_, prop] = binding.split('.');

if (!this.bindings[prop]) this.bindings[prop] = [];

this.bindings[prop].push(el);

// 初始化值

el.value = this.presenter?.model.data[prop] || '';

// 双向绑定

el.addEventListener('input', (e) => {

this.presenter.model.setData(prop, e.target.value);

});

});

// 处理事件绑定

this.component.querySelectorAll('[data-event]').forEach(el => {

const [eventType, handler] = el.dataset.event.split(':');

el.addEventListener(eventType, () => {

if (this.presenter && this.presenter[handler]) {

this.presenter[handler]();

}

});

});

}

bindPresenter(presenter) {

this.presenter = presenter;

this.presenter.model.subscribe((key) => this.updateBindings(key));

}

updateOutput(text) {

this.elements.output.textContent = text;

}

updateBindings(key) {

const value = this.presenter.model.data[key];

this.bindings[key]?.forEach(el => {

el.value = value;

});

}

showError(message) {

this.elements.usernameError.style.display = 'block';

this.elements.usernameError.textContent = message;

}

hideError() {

this.elements.usernameError.style.display = 'none';

}

}

// ====================================

// 装饰器翻译系统 依赖注入

// ====================================

// 装饰器工厂清单

const registerServices = new Map();

// 装饰器工厂方法

function registerService(serviceName) {

return function(target) {

registerServices.set(serviceName, new target());

target.prototype.\_\_serviceName = serviceName;

return target;

};

}

// 装饰器工厂流水线

function withDecorators(...decorators) {

return function(target) {

return decorators.reduceRight((acc, decorator) => {

return decorator(acc);

}, target);

};

}

// ====================================

// 装饰器翻译系统 服务定义

// ====================================

const StorageService = withDecorators(

registerService('storageService')

)(

class {

save(key, value) {

localStorage.setItem(key, JSON.stringify(value));

}

load(key) {

return JSON.parse(localStorage.getItem(key));

}

}

);

const UserService = withDecorators(

registerService('userService')

)(

class {

constructor() {

this.storage = registerServices.get('storageService');

}

saveUser(user) {

this.storage.save('currentUser', user);

}

}

);

// ====================================

// Presenter 类 - 业务逻辑

// ====================================

class Presenter {

constructor(view, model) {

this.view = view;

this.model = model;

this.view.bindPresenter(this);

// 依赖注入

registerServices.forEach((service, name) => {

if (this.constructor === service.constructor) return;

this[name] = service;

});

}

handleSubmit() {

try {

this.model.validateUsername(this.model.data.username);

this.view.updateOutput(`欢迎, ${this.model.data.username}!`);

this.view.hideError();

// 使用注入的服务保存用户

if (this.userService) {

this.userService.saveUser({

username: this.model.data.username,

timestamp: new Date()

});

}

} catch (error) {

this.view.showError(error.message);

}

}

handleSave() {

console.log('保存商品: ', this.model.data.name);

// 使用存储服务保存商品

if (this.storageService) {

this.storageService.save('lastProduct', {

name: this.model.data.name,

savedAt: new Date().toISOString()

});

}

}

}

// ====================================

// 组件初始化系统

// ====================================

class App {

static dataComponents = {};

static registerModel(name, modelClass) {

this.dataComponents[name] = modelClass;

}

static init() {

document.querySelectorAll('[data-component]').forEach(dataComponent => {

const modelName = dataComponent.dataset.model;

const modelClass = this.dataComponents[modelName];

if (!modelClass) throw new Error(`未注册的模型: ${modelName}`);

const model = new modelClass();

const view = new View(dataComponent.dataset.component);

new Presenter(view, model);

});

}

}

// 注册模型

App.registerModel('UserModel', UserModel);

App.registerModel('ProductModel', ProductModel);

// 启动应用

document.addEventListener('DOMContentLoaded', () => App.init());

</script>

</body>

</html>

```

2025-03-30

```

### Point 0.4.2

framework\_base\_01.html

```plain

<!--

以下是基于MVP模式的扩展实现，包含数据绑定、智能事件绑定和组件化支持：

主要改进点说明：

1. 数据绑定系统：

- 通过`data-bind="model.property"`实现双向绑定

- 使用观察者模式自动同步数据变化

- 模型变化时自动更新所有绑定元素

2. 智能事件绑定：

- 通过`data-event="事件类型:处理方法"`格式

- 自动绑定到Presenter的对应方法

- 支持多个不同事件类型绑定

3. 组件化系统：

- 使用`data-model`指定组件关联的模型

- 中央注册系统管理组件模型

- 支持多个独立组件并存

4. 模型继承体系：

- 基础Model类提供数据管理能力

- 支持通过继承创建领域专用模型

- 每个组件可以指定自己的模型类型

扩展后的框架使用示例：

1. 定义组件：

```html

<div data-component="userForm" data-model="UserModel">

<input data-id="username" data-bind="model.username">

<button data-event="click:handleSubmit">提交</button>

</div>

```

2. 创建专用模型：

```javascript

class UserModel extends Model {

constructor() {

super();

this.setData('username', '');

}

validateUsername(value) {

if (value.length < 3) throw new Error("用户名太短");

}

}

```

3. 注册并使用组件：

```javascript

App.registerModel('UserModel', UserModel);

```

该实现保持MVP模式核心思想，同时增加了现代前端框架的常用特性，适合构建中等复杂度的Web应用。

-->

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<style>

[data-id] { margin: 5px; padding: 5px; }

.container { border: 1px solid #ccc; padding: 20px; max-width: 500px; }

.error { color: red; display: none; }

</style>

</head>

<body>

<!-- 组件1 -->

<div class="container" data-component="userForm" data-model="UserModel">

<input type="text" placeholder="Enter username" data-id="username" data-bind="model.username">

<button data-id="submitBtn" data-event="click:handleSubmit">Submit</button>

<div data-id="usernameError" class="error"></div>

<div data-id="output"></div>

</div>

<!-- 组件2 -->

<div class="container" data-component="productForm" data-model="ProductModel">

<input type="text" placeholder="Enter productName" data-id="productName" data-bind="model.name">

<button data-event="click:handleSave">Save Product</button>

</div>

<script>

// ====================================

// Model 类 - 数据管理

// 基础模型类

// ====================================

class Model {

constructor() {

this.\_data = {};

this.listeners = [];

}

get data() {

return this.\_data;

}

setData(key, value) {

this.\_data[key] = value;

this.notify(key);

}

subscribe(callback) {

this.listeners.push(callback);

}

notify(key) {

this.listeners.forEach(cb => cb(key));

}

}

// 用户模型

class UserModel extends Model {

constructor() {

super();

this.setData('username', '');

}

validateUsername(value) {

if (value.length < 3) throw new Error("用户名至少3个字符");

}

}

// 商品模型

class ProductModel extends Model {

constructor() {

super();

this.setData('name', '');

}

}

// ====================================

// View 类 - UI管理

// ====================================

class View {

constructor(dataComponent) {

this.component = document.querySelector(`[data-component="${dataComponent}"]`);

this.elements = {};

this.bindings = {};

// 自动发现元素

this.component.querySelectorAll('[data-id]').forEach(el => {

this.elements[el.dataset.id] = el;

});

// 处理数据绑定

this.component.querySelectorAll('[data-bind]').forEach(el => {

const binding = el.dataset.bind;

const [\_, prop] = binding.split('.');

if (!this.bindings[prop]) this.bindings[prop] = [];

this.bindings[prop].push(el);

// 初始化值

el.value = this.presenter?.model.data[prop] || '';

// 双向绑定

el.addEventListener('input', (e) => {

this.presenter.model.setData(prop, e.target.value);

});

});

// 处理事件绑定

this.component.querySelectorAll('[data-event]').forEach(el => {

const [eventType, handler] = el.dataset.event.split(':');

el.addEventListener(eventType, () => {

if (this.presenter && this.presenter[handler]) {

this.presenter[handler]();

}

});

});

}

bindPresenter(presenter) {

this.presenter = presenter;

this.presenter.model.subscribe((key) => this.updateBindings(key));

}

updateOutput(text) {

this.elements.output.textContent = text;

}

updateBindings(key) {

const value = this.presenter.model.data[key];

this.bindings[key]?.forEach(el => {

el.value = value;

});

}

showError(message) {

this.elements.usernameError.style.display = 'block';

this.elements.usernameError.textContent = message;

}

hideError() {

this.elements.usernameError.style.display = 'none';

}

}

// ====================================

// Presenter 类 - 业务逻辑

// ====================================

class Presenter {

constructor(view, model) {

this.view = view;

this.model = model;

this.view.bindPresenter(this);

}

handleSubmit() {

try {

this.model.validateUsername(this.model.data.username);

this.view.updateOutput(`欢迎, ${this.model.data.username}!`);

this.view.hideError();

} catch (error) {

this.view.showError(error.message);

}

}

handleSave() {

console.log('保存商品: ', this.model.data.name);

}

}

// ====================================

// 组件初始化系统

// ====================================

class App {

static dataComponents = {};

static registerModel(name, modelClass) {

this.dataComponents[name] = modelClass;

}

static init() {

document.querySelectorAll('[data-component]').forEach(dataComponent => {

const modelName = dataComponent.dataset.model;

const modelClass = this.dataComponents[modelName];

if (!modelClass) throw new Error(`未注册的模型: ${modelName}`);

const model = new modelClass();

const view = new View(dataComponent.dataset.component);

new Presenter(view, model);

});

}

}

// 注册模型

App.registerModel('UserModel', UserModel);

App.registerModel('ProductModel', ProductModel);

// 启动应用

document.addEventListener('DOMContentLoaded', () => App.init());

</script>

</body>

</html>

```

## Reference

0. ^ 好 产品 是 偷 不 来 / 抢 不 来 的 偷 来 / 抢 来 只 会 是 刹那 烟火 无法 成 气候

1. ^ [2020.01 刘晓春 是 时候 重新 审视 银企 关系 | 第一财经](https://www.yicai.com/news/100459082.html)

2. ^ "民企 融资 问题 之 所以 久 攻 难 破 是 因为 银行 / 企业 间 没有 形成 良性 有效 的 银企 关系 且 是 一个 深 层次 问题 2019年 底 中央 为 民企 发放 政策 大 礼包 [2019.12 中共中央国务院关于营造更好发展环境支持民营企业改革发展的意见 | 求是网](http://www.qstheory.cn/zdwz/2019-12/22/c\_1125375095.htm) 提出 改善 银企 关系 鼓励 银行 同 民营 企业 建立 中 / 长 期 银企 关系"

3. ^ 沁园春 长沙 说 "独立 寒 秋 湘江 北 去 橘子洲 头 看 万 山 红 遍 层 林 尽 染 漫 江 碧 透 百 舸 争 流 鹰 击 长 空 鱼 翔 浅 底 万 类 霜 天 竞 自由 怅 寥廓 问 苍茫 大 地 谁 主 沉浮 携 来 百 侣 曾 游 忆 往 昔 峥嵘 岁月 稠 恰 同学 少年 风 华 正 茂 书 生 意 气 挥 斥 方 遒 指 点 江山 激 扬 文字 粪土 当 年 万 户 侯 曾 记 否 到 中 流 击 水 浪 遏 飞 舟"

4. ^ DeepSeek 客观 实现 各 行 / 各 业 / 各 领域 专家 一张 桌 共同 参与 的 中国特色 现代企业 制度 衍生 的 平台经济 新 篇章 为 开辟 中国特色 社会主义 现代化 (中国式 现代化) 即 公有制经济 为 主体 / 多 种 所有制经济 共同 发展 及 按 劳 分配 方式 为 主体 / 多 种 分配 方式 并存 同 中国特色 社会主义 市场经济 体制 转型 为 中国特色 社会主义 基本经济 制度 作 出 新 贡献

5. ^ 不 以 学位 / 学历 为 目的 教育 没有 意义 同样 不 以 创新 为 目的 创业 / 就业 没有 意义 自我 感动 是 集体主义 的 寄生虫 必须 干净 / 彻底 消灭

6. ^ 1962.04 国际歌 "从 来 就 没有 什么 救世主 也 不 靠 神仙 皇帝 要 创造 人类 的 幸福 全 靠 我们 自己 我们 要 夺回 劳动 果实 让 思想 冲破 牢笼 快 把 那 炉火 烧得 通红 趁热 打铁 才 能 成功 是 谁 创造 了 人类 世界 是 我们 劳动 群众 一切 归 劳动者 所有 哪 能 容得 寄生虫 最 可恨 那些 毒蛇 猛兽 吃 尽 了 我们 的 血肉 一 旦 把 它们 消灭 干净 鲜红 的 太阳 照 遍 全球"

7. ^ 设计 不 一定 是 语言 也 可以 是 别 的 什么 / 设计 基础 是 逻辑学 本来 就 存在 可 验证性 及 不 可 验证性 / 不 可 验证性 不 等于 失去 有效性 / 一刀 切 问题 因 临床心理 认知行为 疗法 失去 有效性 产生 对 错误 认知 不 可 验证性 负责 依 临床医学 循证法 辩证 是否 需要 同时 采取 低 剂量 / 短 周期 药物 治疗 根据 临床数据 作 进一步 治疗 及 康复 训练

8. ^ 终身学习计划 (Lifelong Learning Plan) 是 设计 不 可 分割 的 一 部分 因 联合国教科文组织 (联合国 教育 / 科学 / 文化 组织) 产生 对 联合国宪章 负责

9. ^ 企业 是 联合国 各 主权 成员国 间 共同 参与 / 共同 协商 / 共同 治理 / 共同 建设 / 共同 分享 基本 集体主义 外观 不 存在 脱离 集体主义 的 个人主义

10. ^ 文书 应 体现 全 心 全 意 为 人民 服务 理念 言语 处处 体现 冷战 思维 背离 中央 8 项 规定 精神 依 法定 程序 针对 背离 中央 8 项 规定 精神 的 言论 / 行为 追责 / 处罚 / 清理 零 容忍 [11]

11. ^ [2025.03 锲 而 不 舍 落实 中央 8 项 规定 精神 改 文风 / 会风 / 话风 是 改 作风 题 中 之 义 | 中国军网](http://www.81.cn/yw\_208727/16377678.html) [12]

12. ^ "当 前 全 党 上 下 正 开 展 深入 贯彻 中央 8 项 规定 精神 学习 教育 / 纵 深 推进 作风 建设 / 改 作风 涉及 方方 面面 / 形成 好 文风 / 会风 / 话风 是 检视 作风 建设 成效 重要 标尺 更 是 持续 改 作风 题 中 之 义 改 作风 / 文风 / 会风 / 话风 要 一 体 推进 / 相互 促进" [13]

13. ^ "文章 怎么 写 / 会 怎么 开 / 话 怎么 说 都 直接 映射 党员 干部 作风 状况 直接 关乎 党 / 政府 在 群众 心 中 形象 / 早 在 中央 8 项 规定 出 台 之 初 / 党 中央 对 精 文 简 会 等 方面 提出 明确 要求 以 上 率 下 作 示范 / 树立 标杆 / 深入 学习 贯彻 中央 8 项 规定 精神 / 要 时时 对 标 对 表 / 在 改 文风 / 会风 / 话风 绵绵 用 力 / 以 群众 看 得 见 的 变化 体现 作风 建设 新 成效" [14]

14. ^ [2025.03 八一 锐评 作风 正派 公道 处事 | 中国军网](http://www.81.cn/yw\_208727/16377724.html) [15]

15. ^ "我们 共产党人 为 的 是 大 公 / 守 的 是 大 义 / 求 的 是 大 我 作风 正派 / 公道 处事 是 我 党 我 军 作风 建设 内 在 要求 也 是 共产党人 行为 准则 焦裕禄 / 谷文昌 / 杨善洲 / 杨业功 等 我 党 优秀 领导 干部 坚持 原则 / 不 搞 特权 / 既 造 福 一 方 也 成就 自我 永远 是 我们 学习 榜样 / 军队 领导 干部 只 有 一 心 为 公 / 时时 洁 身 自 好 / 事事 公 私 分 明 / 处处 以 身 作 则 才 能 真正 为 战斗力 建设 及 官兵 利益 尽 到 职 / 负 好 责" [16]

16. ^ 日常 小 事 就 是 常常 大 事 必然 会 导致 慢 作为 / 乱 作为 / 不 作为 牢记 为 人民 服务 / 受 人民 监督 / 人民 心 中 有 账本 才 能 打 得 一 拳 开 免 得 百 拳 来 人民 队伍 产生 寄生虫 一 经 发现 必然 主动 消灭 因 中华人民共和国 国防 建设 产生 对 中华人民共和国 国防 建设 负责 自觉 / 主动 实名 言论 / 行为 应 常态化 [17]

17. ^ 卜算子 咏梅 说 "读 陆游 咏梅 词 反 其 意 而 用 之 风 雨 送 春 归 飞 雪 迎 春 到 已 是 悬崖 百 丈 冰 犹 有 花 枝 俏 俏 也 不 争 春 只 把 春 来 报 待 到 山 花 烂漫 时 她 在 丛 中 笑" [18]

18. ^ 前 苏联 精英 教育 糟粕 思想 终究 会 导致 帝国主义 拥护者 亲手 消灭 自身 因 玩 火 者 必 自 焚

19. ^ Sexy Signal 是 事业 / 家业 双 丰收 前提 / 歪 门 邪 道 不 可 取

20. ^ 有 些 人 日常 / 常常 因 站位 环境 不 能 被 阳光 覆盖 / 期待 能 被 阳光 覆盖 成 了 执念 / 哎 聚光灯效应 注定 无缘 这 一 类 站位 拿不起 / 想不开 / 放不下 就 不 存在 这 一 类 站位 / 说 什么 也 改变 不 了 客观 事实 / 只 有 丰富 / 完善 自身 综合 实力 去 克服 自身 消极 / 抵触 情绪 别 无 选择 / 责任 就 在 那 拿起 / 放下 自己 决定 自己 负责

21. ^ 临床 缩短 病程 means 缩短 直 至 消灭 临床 急性期 延长 临床 恢复期 / 一代 又 一代 人 何尝 不 是 舍 小 家 顾 大 家

## Signature

```plain

0. ^ 依 [2011.10 中华人民共和国居民身份证法 | 中国人大网](http://www.npc.gov.cn/npc/c2/c12435/201905/t20190522\_81831.html) 执行

1. ^ 中华人民共和国居民身份证法 (1) 为 证明 居住 在 中华人民共和国 境 内 的 公民 身份 保障 公民 的 合法 权益 便利 公民 进行 社会 活动 维护 社会 秩序 制定 本 法

2. ^ 中华人民共和国居民身份证法 (6) 居民身份证 式样 由 国务院 公安 部门 制定 居民身份证 由 公安 机关 统一 制作 / 发放 居民身份证 具备 视读 / 机读 两种 功能 视读 / 机读 的 内容 限 本法 (3.1) 规定 的 项目 公安 机关 及 其 人民 警察 对 因 制作 / 发放 / 查验 / 扣押 居民身份证 而 知悉 公民 的 个人 信息 应 予 以 保密

3. ^ [2024.08 学习 决定 每 日 问答 | 怎样 理解 建立 专家 实名 推荐 的 非 共识 项目 筛选 机制 | 中国政府网](https://www.gov.cn/zhengce/202408/content\_6970604.htm)

4. ^ 专家 实名 推荐 的 非 共识 项目 筛选 机制 允许 中华人民共和国 公民 依 法 主动 报告 个人 有关 事项 且 主动 报告 个人 有关 事项 因 中华人民共和国 国防 建设 产生 对 中华人民共和国 国防 建设 负责 主动 报告 个人 有关 事项 遵循 尽力而为 / 量力而行 / 适可而止 原则 不 允许 强迫 善意 第三方 执行 谁 发起 / 谁 监督 / 谁 执行 为 人民 服务 / 受 人民 监督 / 人民 心 中 有 账本

5. ^ Signature 格式 为 "Signature> 信仰 信仰人完整国家籍贯 信仰人姓名 信仰人公民身份号码"

```

Signature> 中国特色共产主义 中华人民共和国 唐铭泽 210202199702194932