**张子玥Markdown笔记及其他**

一级标题#空格 二级标题##空格 以此类推

加粗\*\*文字\*\*空格斜体\*文字\*空格

下划线~文字~空格 删除线~~文字~~空格 分割线---或\*\*\*

有序列表1. 空格 无序列表-空格或\*空格

任务列表【】空格

增加缩进tab 减少缩进shift+tab

引用》空格

代码‘代码’

代码块‘’‘（？

**10.30**

**git学习**

1. 配置用户名和邮箱：

* git config --global user.name ""
* Git config --global user.email xxx@ .com

1. 创建目录/仓库

* mkdir learn-git
* cd learn-git
* git init

（在learn-git下创建一个仓库）

git init my-repo

1. 远程仓库

git clone +地址

1. 查看仓库的状态：git status
2. 查看本地内容：ls
3. 查看暂存区内容：git ls-files
4. 添加到暂存区：

* git add
* 可以使用通配符：git.add\*.xxx(文件类型)（将所有这类文件添加到暂存区）
* 也可以使用目录，例如：git add.

1. 提交：

* git commit（只能提交暂存区，不会提工作区）
* git.commit-m"哔哩吧啦”来指定提交信息

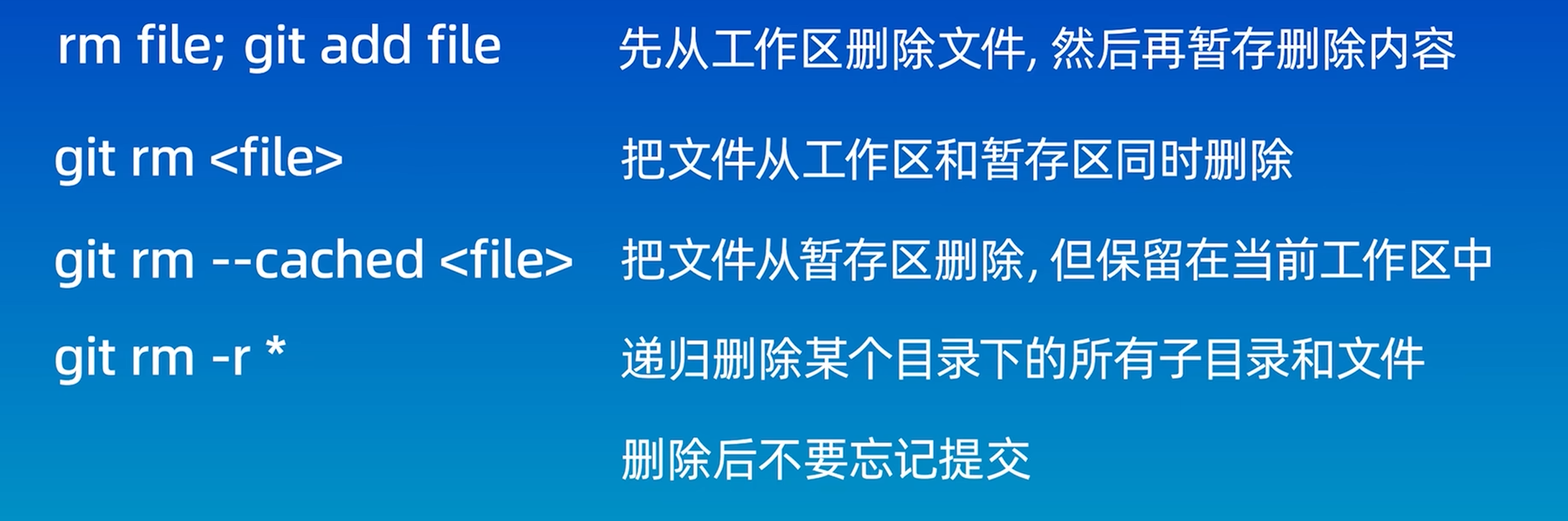
1. 查看仓库提交历史记录：git log
2. Git reset回退版本

* Git reset--soft 保留工作区与暂存区内容
* Git reset--hard 不保留工作区和暂存区内容
* Git reset--mixed 保留工作区不保留暂存区内容

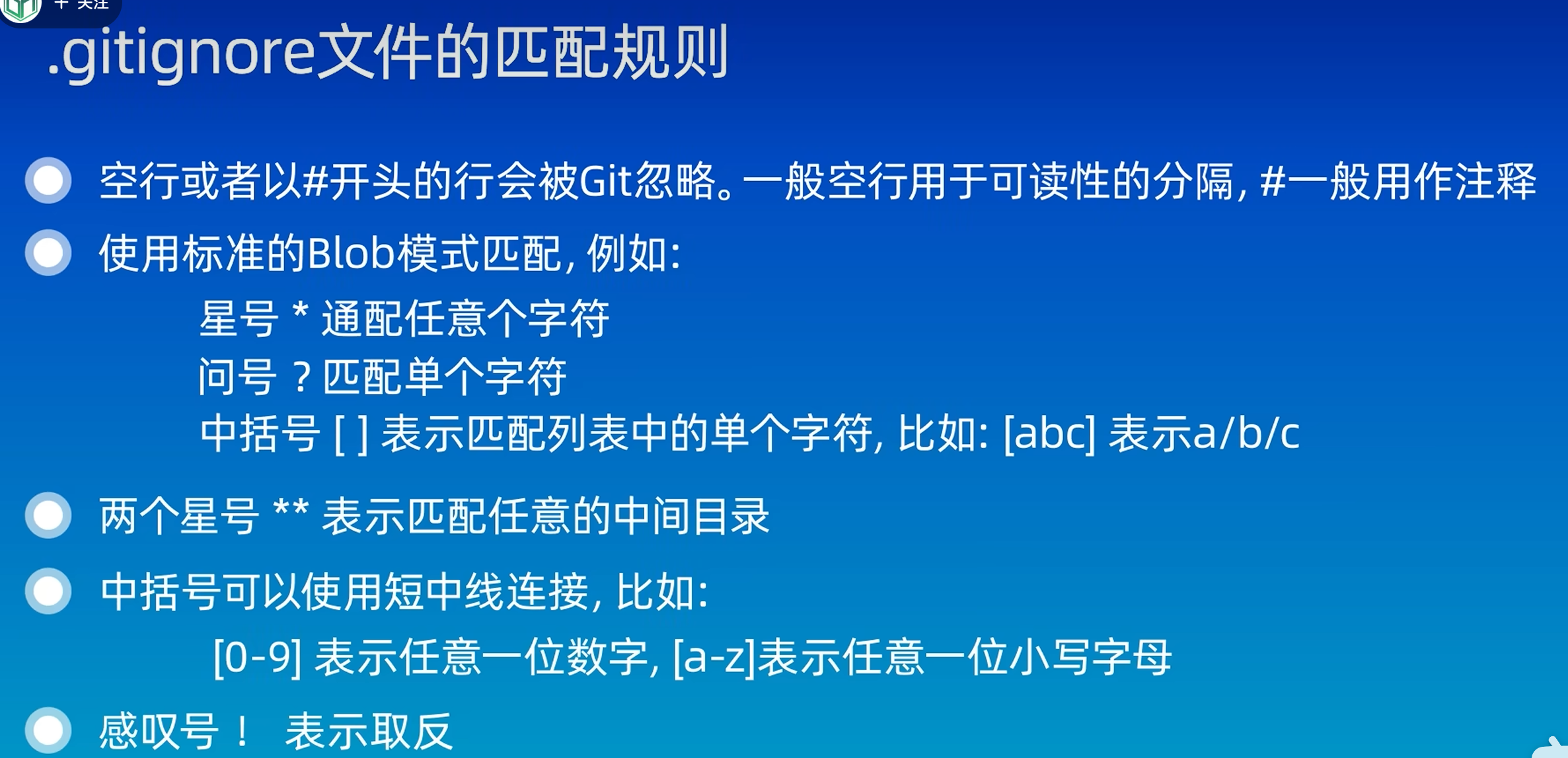
1. 比较差异：Git diff

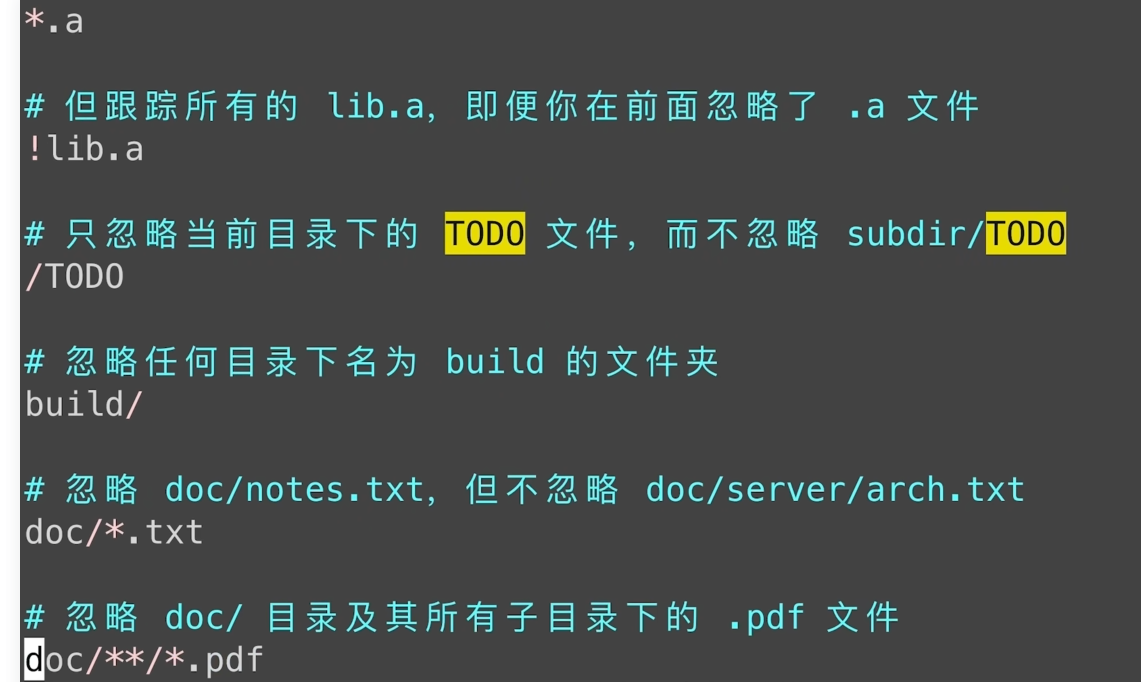
* 工作区vs暂存区：Git diff
* 工作区+暂存区vs本地仓库：Git diff HEAD
* 暂存区vs本地仓库：Git diff--cached/git diff --staged
* 比较提交之间的差异：Gid diff <commit\_hash><commit\_hash>/git diff HEAD~HEAD
* 比较分支之间的差异：Git diff<branch\_name><branch\_name>

1. 删除文件：git rm



1. Git ignore



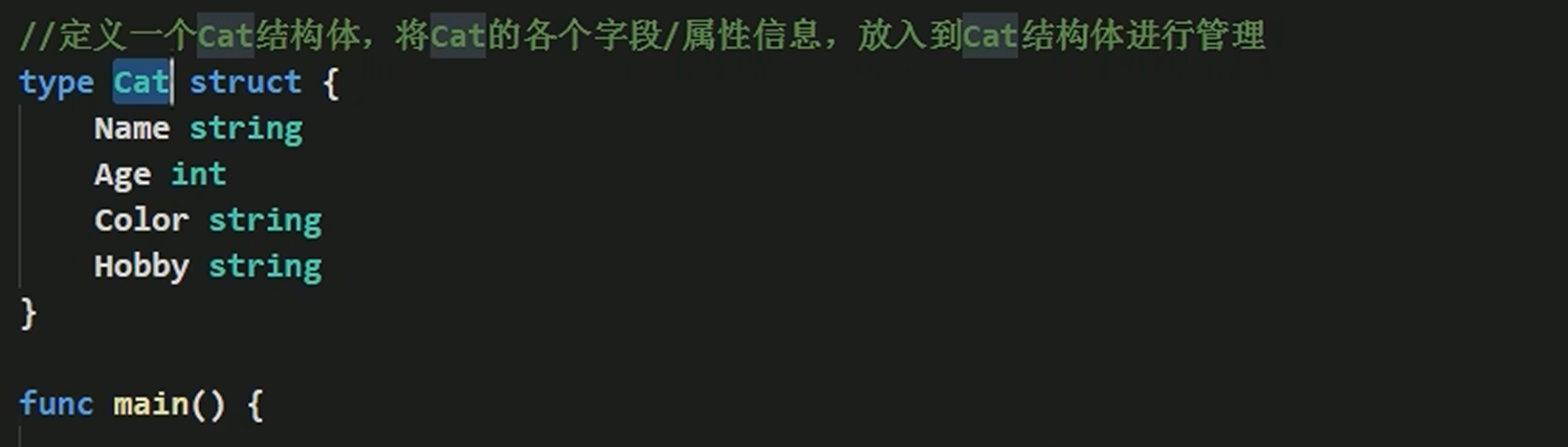


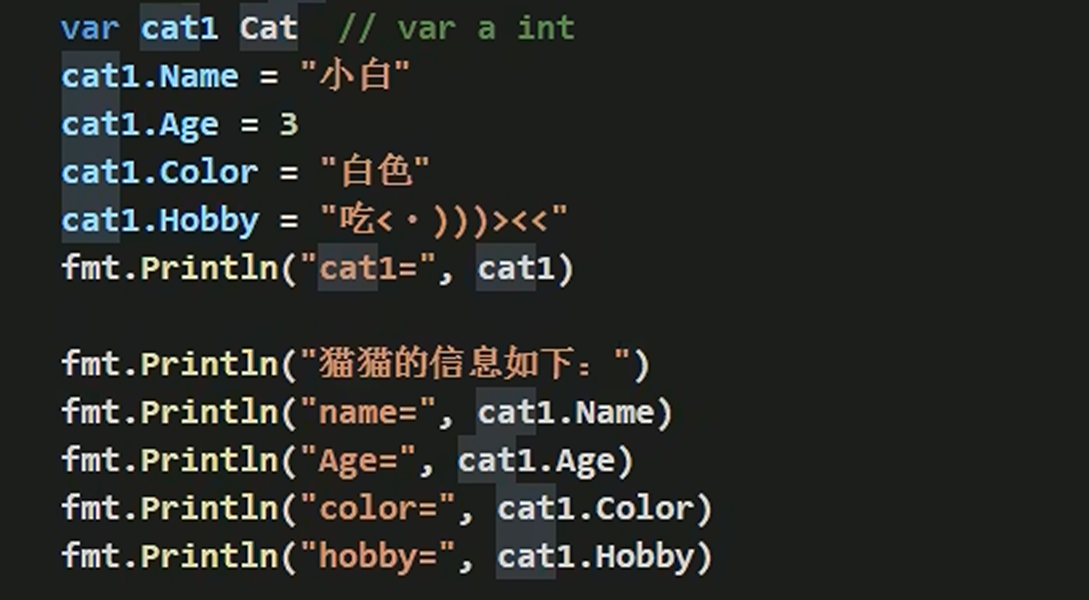
**11.1**

**结构体**

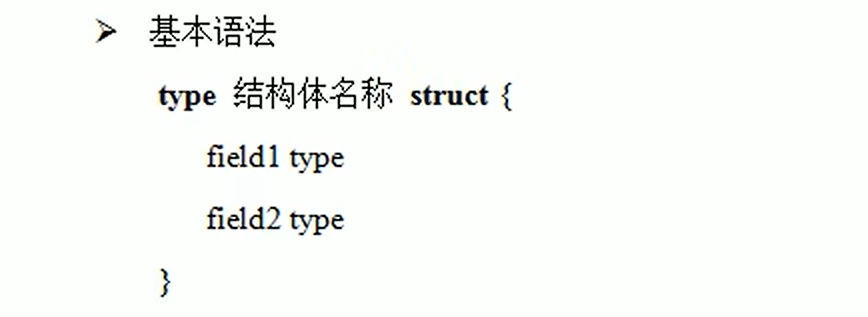
1. 将一个事物的特性提取出来，形成一个新的数据类型，就是一个结构体
2. 根据这个结构体，我们可以创建多个变量

* 下面为神秘网课截图展示：可爱猫猫身平信息大展示



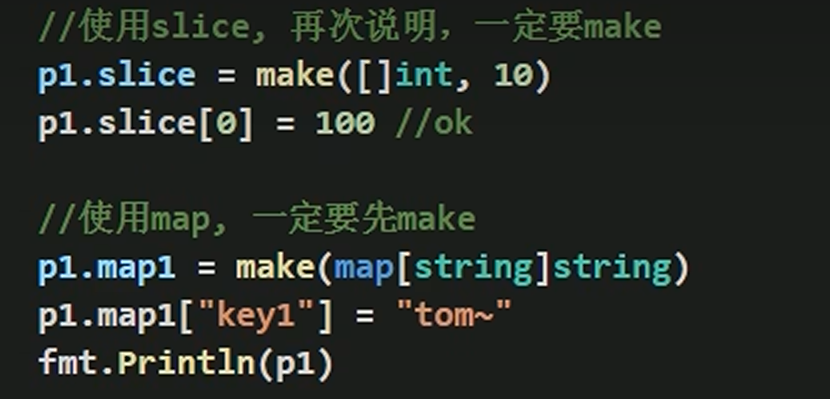


1. 声明结构体



1. 注意细节：

* 字段的类型可以是基本数据类型，数组，引用类型
* 若没给字段赋值，会是默认值；指针，slice，map的零值都是nil（即还没分配空间，需要make）



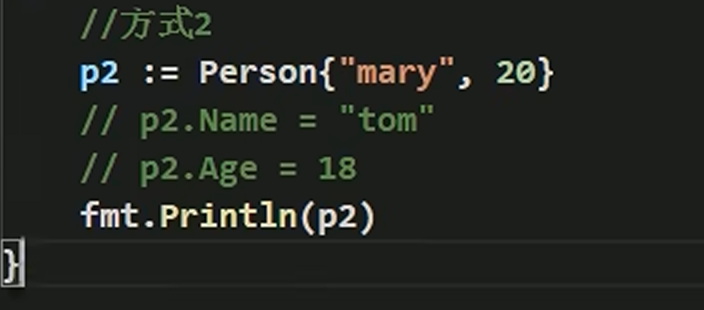
* 结构体是值类型，默认使用值传递



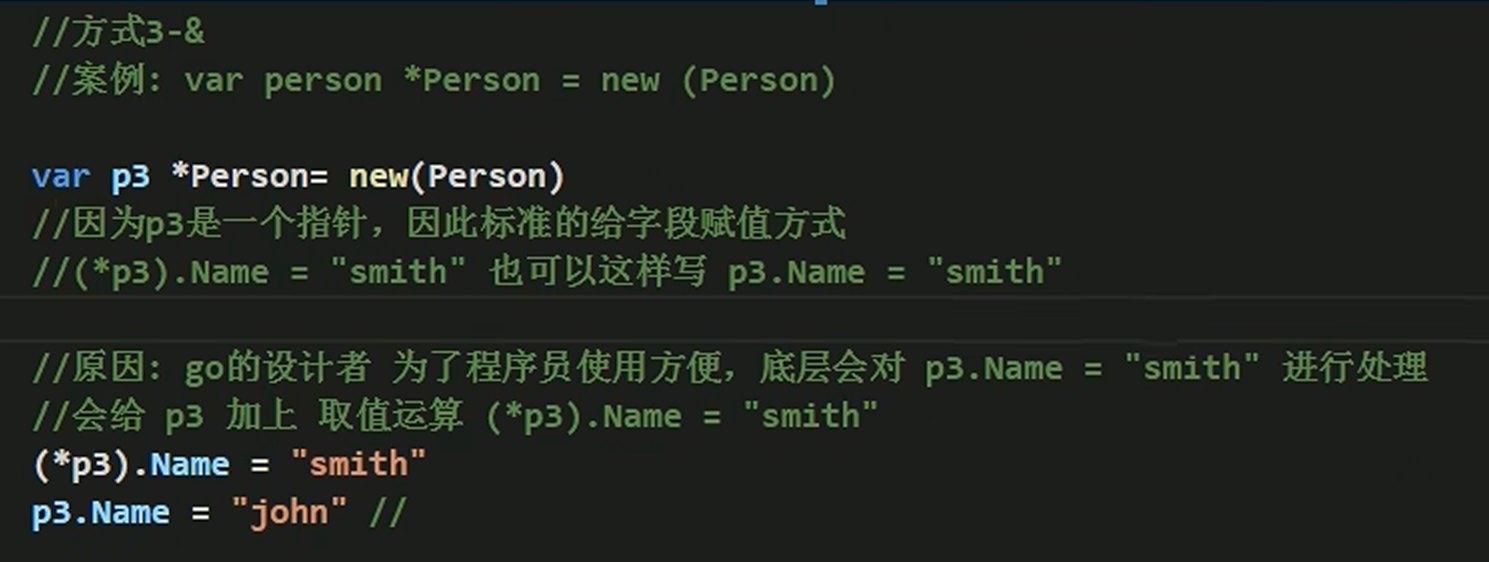
1. 如何创建结构体变量和访问字段

* 法一：前面内个

法二：



法三：指针？



法四：



1. 内存分配机制案例：

