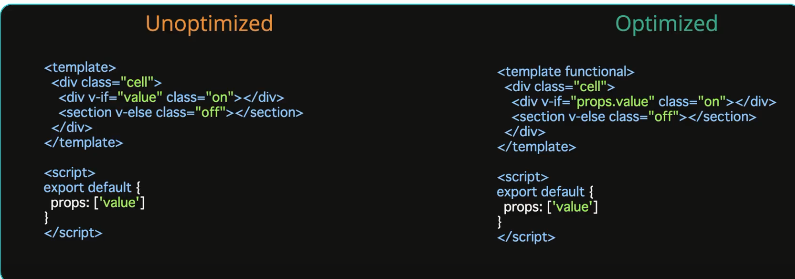
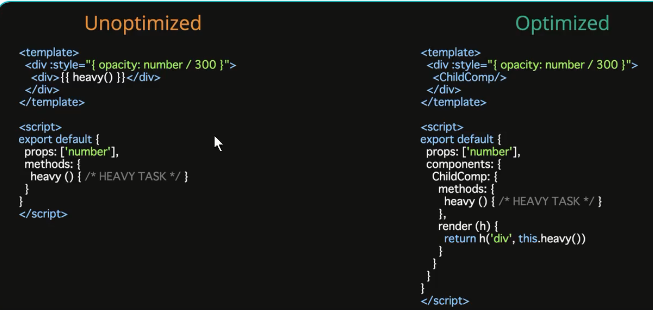
# 函数型组件

<template functional></template> 原理没有加functional的会因为生命周期有所耗时，加了的就是以函数形式运行处理



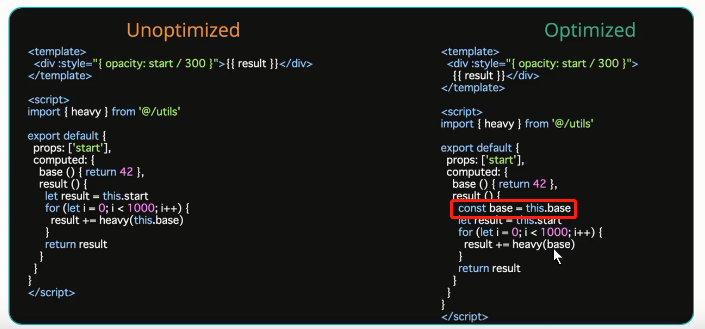
# 子组件拆分

重处理不要写在本组件中，最好单独一个组件出来，这涉及到父子组件的生命周期及更新周期的问题



# 局部变量

把定义在vue中的属性方法在其他地方调用时先局部定义，再调用



# 4活用v-show，少用v-if

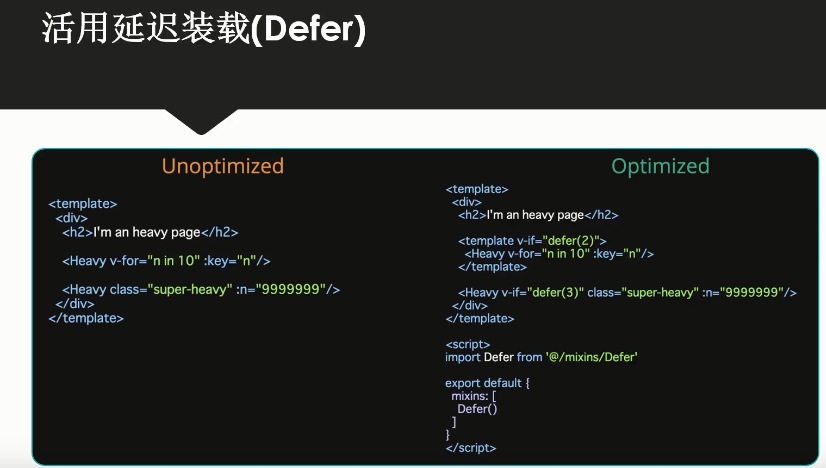
V-show不操作dom树，v-if会动态改变dom树

# 5使用keep-alive

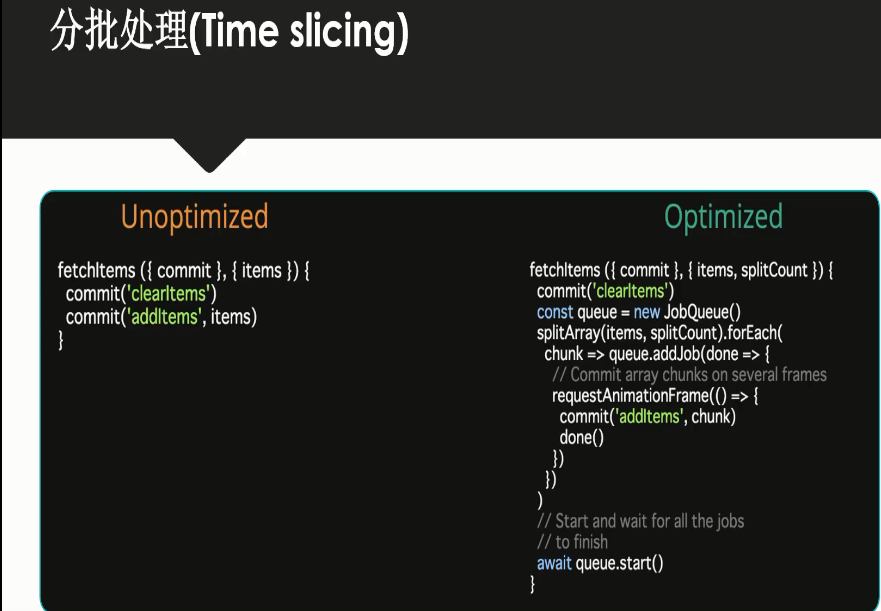


# 活用延时加载

使用defer混合，仅是先加载某已部分，造成在加载的错觉，

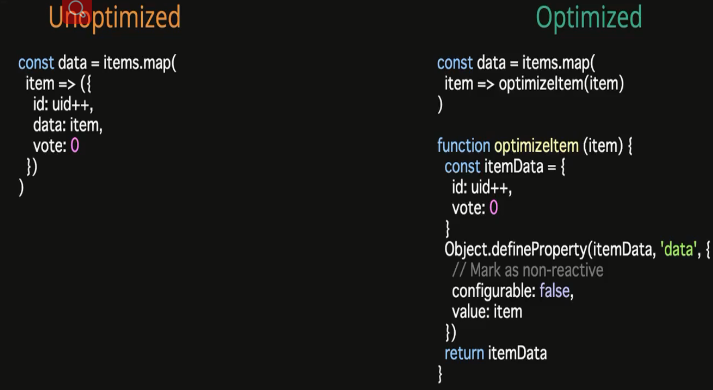


# 分批处理

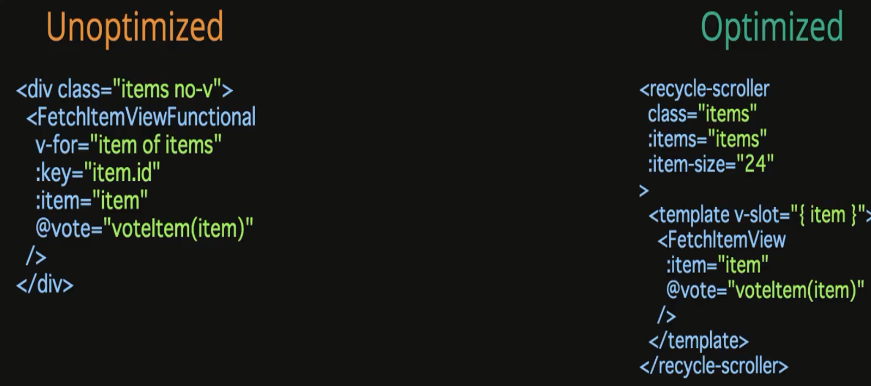


# 非响应模式（非观察模式）

因为vue是双向绑定，绑定的组件会实时更新，右边的我不想vue实时去监听他就用Object.defineProperty



# 仅渲染可视化的部分



# webpack 升级到4 的处理

// 大部分根据这个链接操作就可以升级

<https://blog.csdn.net/qq_25243451/article/details/80331269>

A.npm i webpack@4 -D

B.npm i webpack-dev-server@3 -D

C.npm i webpack-cli -D

D.npm i html-webpack-plugin@3 -D

E.