# 第一章 总体梳理

## vue complete

1. init -> emit created

2. compile template to render function-> beforeMounted

3. exe render function, generate vdom, compile vdom to real dom(render view, or 挂载阶段) -> mounted

Refer to patch + render vdom

https://github.com/snabbdom/snabbdom

1. state change, vdom change -> beforeUpdate (这部分再确认下，是不是渲染函数变化， vdom理论上在下一步生成比较合理)

2. compile vdom to real dom(render view) -> updated

## vue runtime

## Some sumarries:

1. runtime beforemount 已经有el了 (beforeMounte === mounted)

2. 响应式在mounted之前到底有没有用？

在书P194页中，在beforeMount到mounted叫做挂载阶段，挂载过程中会开启watcher来持续最终依赖变化，

在mounted（挂载完成后）vue仍然会持续跟踪状态变化，当状态变化时，watcher会通知vdom重新渲染视图，

这时，渲染视图前触发beforeUpdate

渲染视图完毕后触发updated

3. P51，组件watcher实例，一个组件内引用了10次某个状态，其实也只有一个watcher在观察这个状态变化，一旦这个状态变化，通知组件，组件通过虚拟dom对比渲染view。

转化成代码，一个组件里面引用了10次状态A，状态A的dep list里面也只有一个组件watcher

4.何时收集依赖

在get中收集，在set中触发，状态的getter，setter是在什么时候执行的呢？在init还是在执行render函数时候(挂载)？

**# when to do dependency collection**

存疑?

猜想一：当render函数(createElement) 读取状态的时候 触发state的get

在beforeMount和mounted之间?

猜想二：感觉答案是在init的时候收集（执行状态的getter）

在挂载，状态改变的时候触发状态的setter

# 第二章 init以及响应式

## 2.1 init

src/core/instance/init.js:

## new vue, init state flow

Init event, life cycle

## beforecreated

init props

init methods

init data

init computed

init watch

## 2.2 reactive

组件级别watcher

Computed watcher

这里请看vue-theory.md里面的dependency collect

# 第三章 模板编译

We have 3 steps:

template -> AST -> createElement(渲染函数)

We have 3 function modules:

解析器，优化器 对应 template -> AST -> traverse AST and mark static node

代码生成器 对应 AST->createElement(渲染函数)

# 第四章 挂载

Execute render function, generate vdom, compile vdom to real dom

挂载的时候，或者状态变化更新vdom的时候，会对比(patch)

# vue diff

will use diff to improve the compare performance, reduce dom change

will have some principles:

1. compare in same hiracky

2. compare same key

3. compare same type same tag or same component

这部分再和书里面对照一下

# 第五章 Vue-server-renderer和template-compile

# 第六章 jsx在vue和react中, SFC在vue中

# 第七章 rn和weex

# 第八章 对比v3