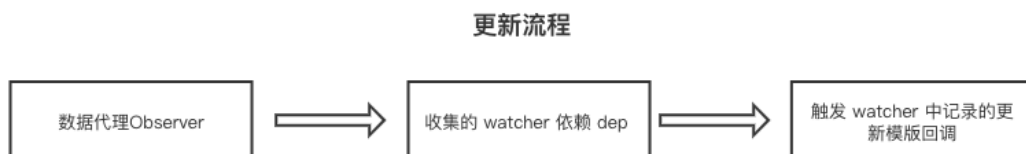
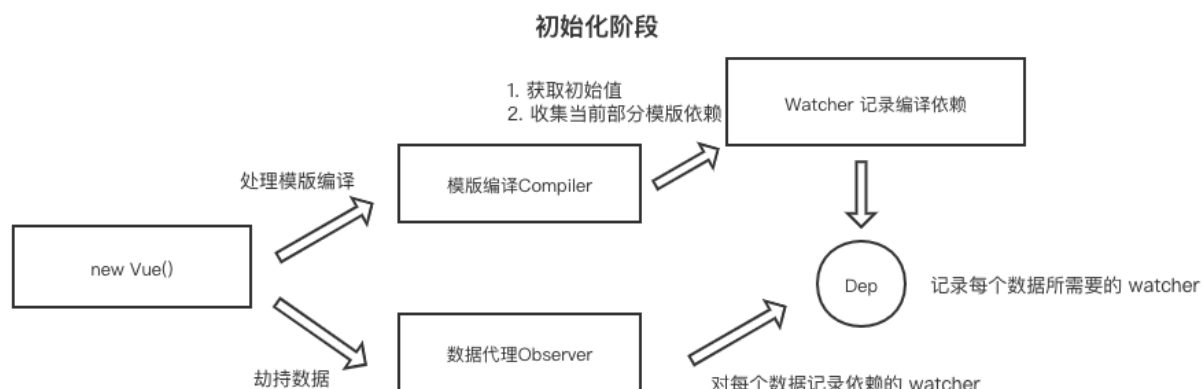


# Vue 源码解析

Vue 使用 `Object.defineProperty` + 观察者模式对数据和模版进行绑定，对于数据来说需要进行更新时，即会触发对应的 `getter` 和 `setter` 函数，在 `setter` 函数中，即可根据对应收集到的依赖，触发对应视图层更新。

对于一次收集和一次更新来说，大致流程如下：



- 实例化 Vue 之后，对内部所有的 data 进行劫持
- 处理对模版的编译 / 或者静态编译好的 render 函数，在处理 **绑定的变量** 时，创建 watcher

- 同时获取当前模版对应的初始值，在读取时触发数据的 getter 函数，进行绑定
- 绑定成功后，后续触发 setter，根据记录的 watcher，即可更新所有的模版内容

当然对于模版的编译来说，有运行时编译、静态编译多种形式，同时对于 Vue 来说，Vue@2 使用了 VNode 来对模版内容进行了描述，所以在模版编译的流程中，使用 VNode 能更好的提升编译和更新的性能。

- 编译模版 Compiler
- 数据处理 Observer
- 依赖收集 Watcher / Dep
- 节点 VNode / patch 更新

对于通用性的理解来说，vue@2 和 vue@1 并没有太多区别，更多的核心在于 VNode 部分，也就是把编译模版的部分使用了 VNode 进行描述，编译为了函数形式，以及增加了 patch 的过程来保证更新。

### 节点 VNode / patch 更新流程：

patch -> patchVNode -> updateChildren

- 单独更新一个 VNode 节点
- 判断 children 节点更新

