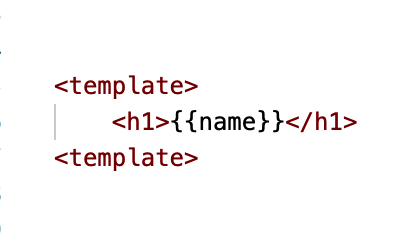
Vue源码分析

需要ES5和ES6的基础知识

总纲

1. 变化侦测
2. 虚拟dom
3. 模板编译原理
4. 整体流程
5. Vue中的设计模式

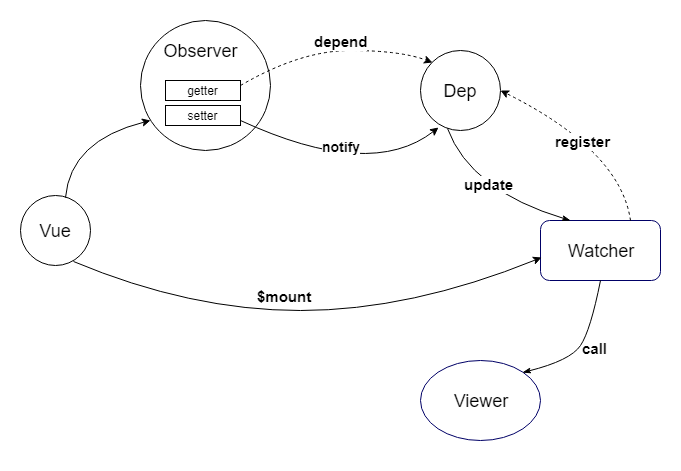
* 变化侦测
* 双向数据绑定的实现原理，改变数据后主动渲染DOM节点。
* 举例：该模板中使用了数据name，所以当它发生变化时，要向使用它的地方（现在取名为“依赖”）发送通知。



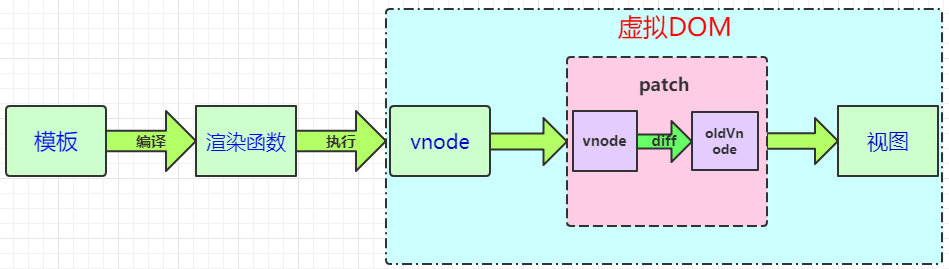
* Object.defineReactive方法侦测对象变化。



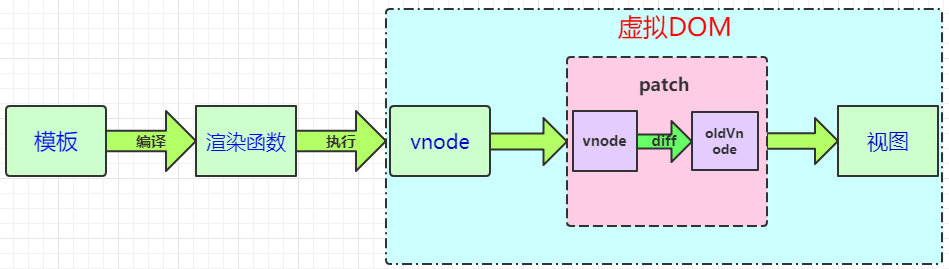
* UML类图



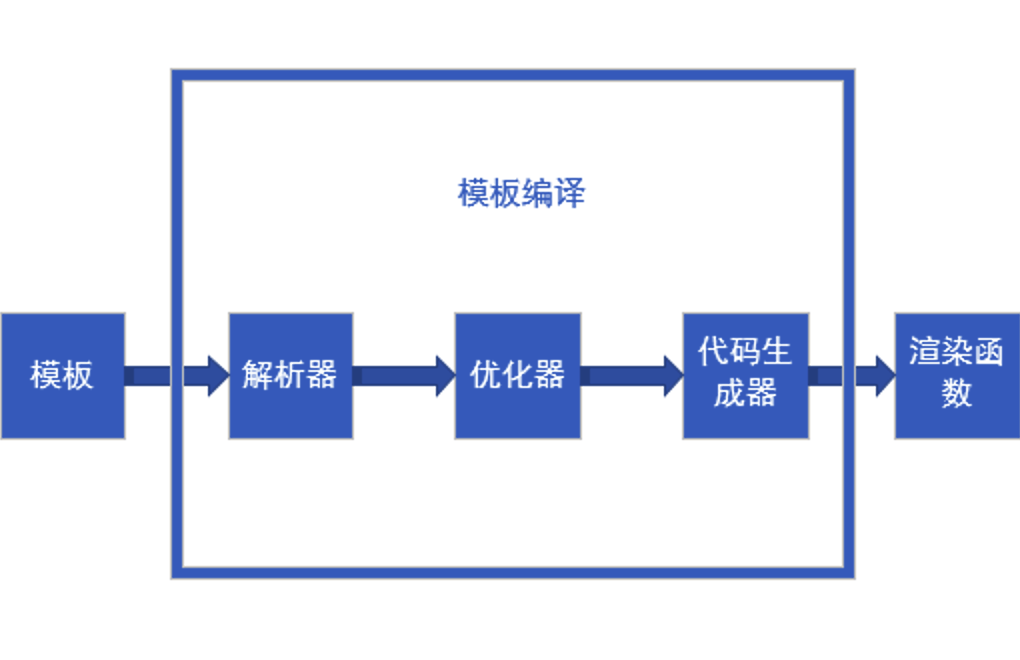
* 虚拟DOM
* 虚拟DOM其实就是一棵以VNode作为基础的树，用对象属性来描述节点，实际上它只是一层对真实DOM的抽象。虚拟DOM的最终目标是将虚拟节点渲染到视图上。vnode可以理解成节点描述对象，它描述了应该怎样去创建真实的DOM节点。



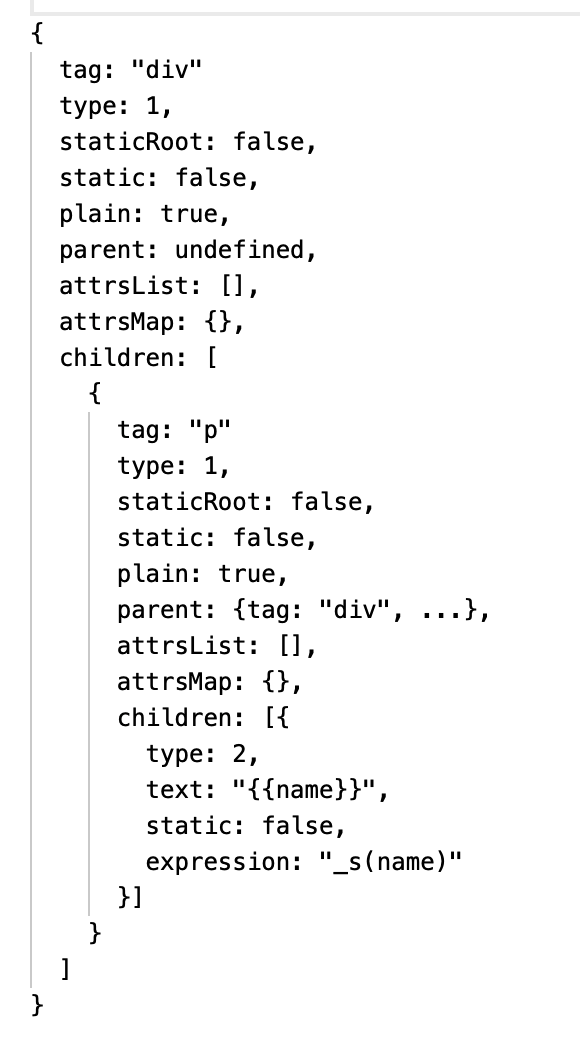
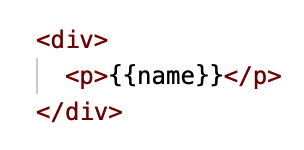
* 虚拟DOM最核心的部分patch，它可以将vnode渲染成真实的DOM。
* 创建新增节点
* 删除已经废弃的节点
* 修改需要更新的节点
* 模板编译原理
* Vue会将模板编译成虚拟DOM渲染函数。渲染函数根据虚拟DOM更新视图。



* 模板编译先由解析器解析成AST抽象语法树，优化，再经代码生成器生成渲染函数。前面说过渲染函数根据vnode更新视图。

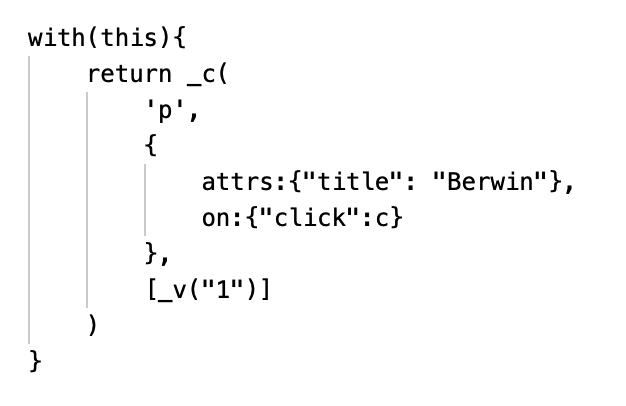


* 解析器生成AST抽象语法树示例

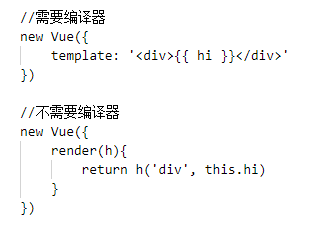


* 代码生成器生成代码字符串示例，代码生成器是模板编译的最后一步，将AST渲染成渲染函数中的内容，称为“代码字符串”。



\_c \_v 都是创建vnode的方法

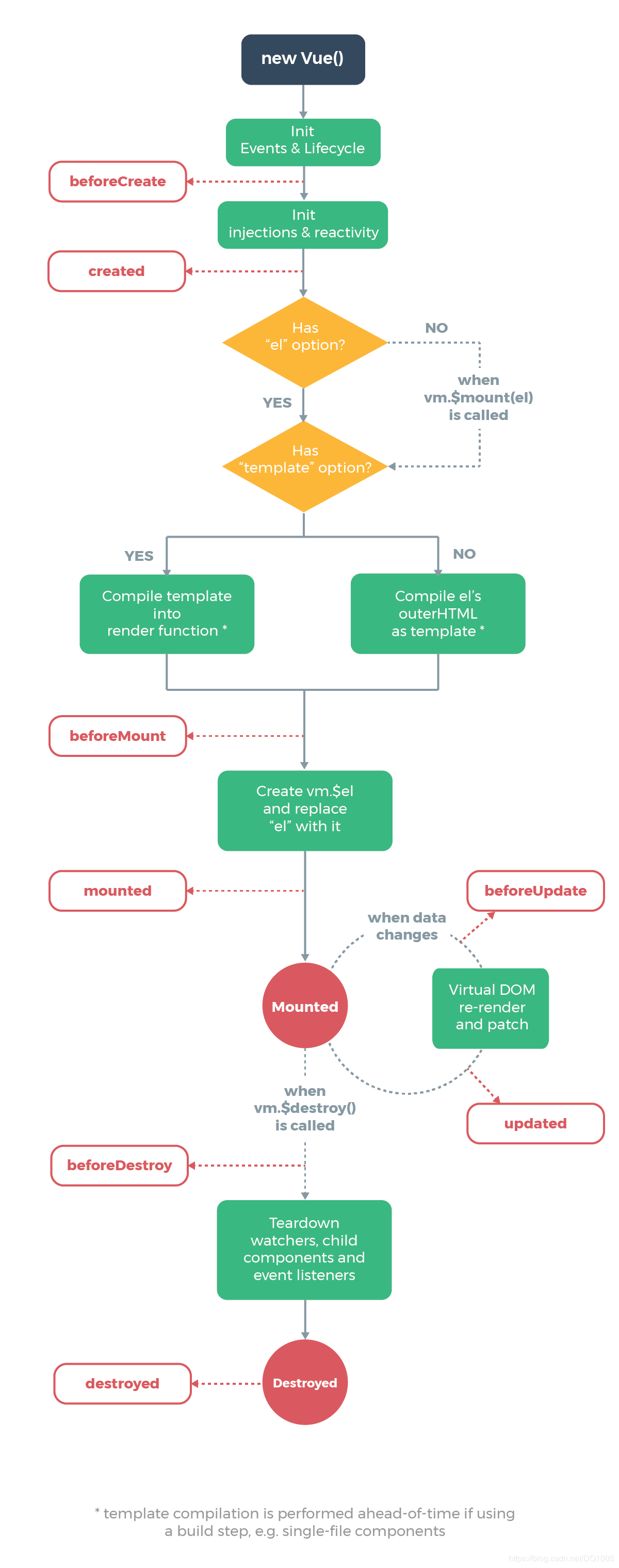
* 整体流程
* 不同的vue.js构建版本的区别，这里只讨论编译器和运行时，完整版包含编译器+运行时。



如果需要在客户端编译模板(比如传入一个字符串给template选项，或挂到一个元素上并以其DOM内部的HTML作为模板)，那么需要用到编译器，因此需要完整版。

当使用vue-loader或vueify的时候，\*.vue文件内部的模板会在构建时与编译成javascript，最终打包完成的文件实际上是不需要预编译的，只需要引入运行时版本即可。

* 如果有el属性或template属性就进行模版编译



* Vue中的设计模式
* 为什么要读源码，好的框架源码可以帮助我们理解什么是好代码, 读源码是为了能写出高质量的代码，像vue spring这些框架中有优质的代码可以借鉴，比如设计模式等。
* Vue混合使用了一些javascript的设计模式
* 观察者模式

通过Object.defineProperty 拦截数据的 get/set ，在set中收集依赖Watcher，在get中触发更新Watcher.notify()，这里就是观察者模式的应用。

* Mixin（混入）模式

Mixin是可以被一个子类或一组子类继承功能的类，目的是函数复用。



* prototype模式

作为其他对象的原型，prototype对象本身实际上是用作构造函数创建没个对象的蓝图。如Vue的实例方法。

