# Vue源码分析——合并配置

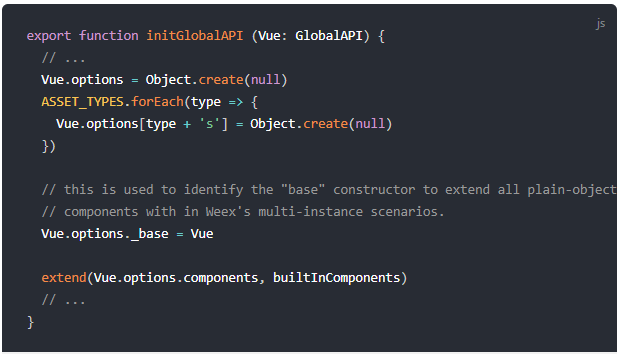
### 问题1：我们都知道在执行Vue.prototype.\_init的时候，首先进行了合并配置，那他是怎么合并的？

答： 要解决这个问题，还要回忆我们之前谈过的两种场景，一种是外部主动调用new Vue的时候，另一种情况是在我们实例化自组建的时候内部调用的new Vue。前者是直接通过mergeOptions方法，第二种是通过initInternalComponent。

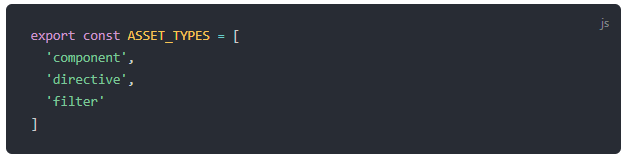


先分开来讨论，先看mergeOptions的情况。

首先resolveConstructorOptions(vm.constructor)我们先不纠结他的内部实现，现在只需要知道他返回的是Vue.options。Vue.options在我们import Vue的时候有通过initGlobalAPI进行过初始化。



在这里ASSET\_TYPES是:



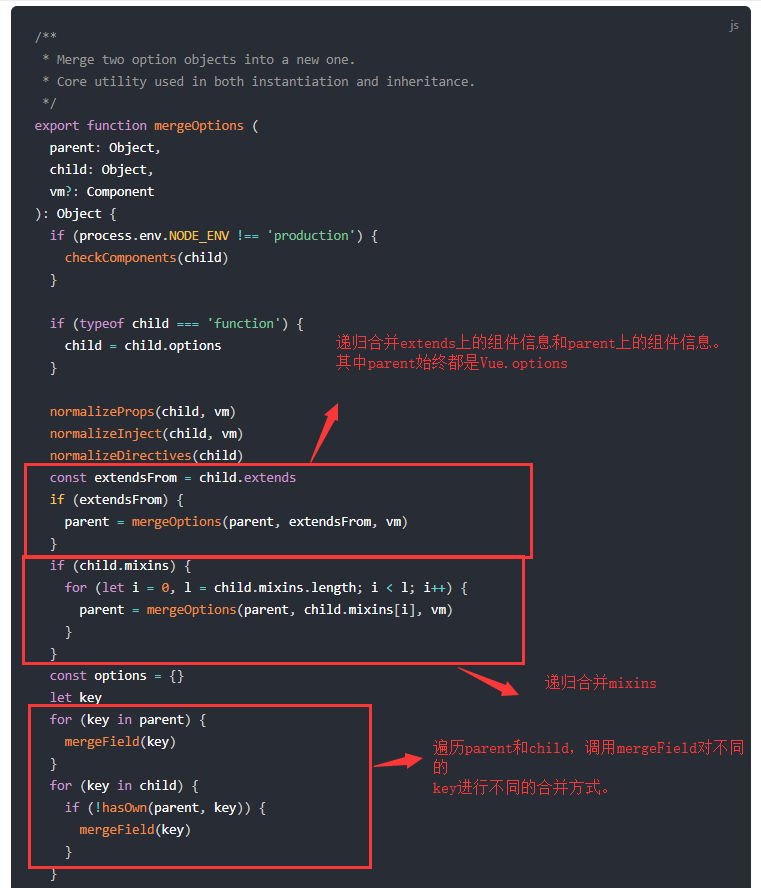
所以很好理解，第一步就是创建一个空对象，然后遍历ASSET\_TYPES,为Vue.options添加components、directive、filter三个全局属性。

然后让Vue.options.\_base执行自己也就是Vue。

最后通过extend去扩展Vue.options.components，这一步就是初始化一些全局的组件，他们分别是： keep-alive、transition、transition-group。这也就是为什么我们不能显示的声明keep-alive等组件就可以使用他们的原因。

在知道了Vue.options都是什么之后，我们把它和我们手写的组件配置以及当前vm实例通过mergeOptions进行合并，他的最终目的就是把Vue.options和我们手写的配置合并成一个新的对象然后返回，作为我们后续初始化Vue实例的数据基础。

来详细看一下mergeOptions方法，其中形参parent和child分别是Vue.options和我们手写的配置。



这个mergeOption的主要逻辑就是通过递归合并extends上的组件信息和parent上的组件信息，**其中parent并非始终都是Vue.options，只是第一次递归才是（图中标注写错了）。**

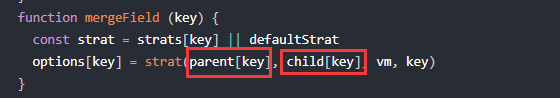
然后得到parent之后，我们就可以新建一个空的option:{}，然后通过遍历parent和child上的键名，一次吧键名对应的值按照某种逻辑合并到option上，最后整个mergeOption函数返回这个option即可完成融合所有任务。让我们来详细的解释一下：

首先我们手写的options可能包含extends的组件，因此，我们我们进一步初始化Vue的数据基础中不能少了extends中的数据，同样的mixins的数据也是如此。

extends和mixins中的组件并不是货真价实的‘组件’,可以理解为一种扩展的组件信息。因此我们只需要递归extend中的内容同时合并到Vue.options上，然后在让Vue.options和我们手写的其他配置合并，最终就可以得到结果。但是如何合并呢？

这就需要我们注意到mergeField方法，他才是真真正正处理合并的方法。

首先，我们得先知道合并的原则，extends上有的配置如果我们在手写的配置中，也就是带入new Vue({})中的{}中已经定义过了，那么他就需要被覆盖或者被合并，所以对不同的键有的需要覆盖而有的则需要合并，因此我们不得不定义一个可以根据当前键名来选取逻辑的方法，这个方法就是mergeField方法。



mergeField的逻辑非常简单，他就是通过接受到当前的键，然后通过调用strat方法得到相应的处理这种键的函数，根据利用得到的这个函数处理key即可。

值得注意的是，mergeField是一个定义在mergeOptions中的方法，他是一个局部函数，因此可以访问parent和child。**从strat方法带入的参数我们可以看到，对某个key进行合并的时候就同时的带入了parent[key]、child[key]。也就是说在获取到key的时候，strat就同时对parent和child上key对应的这一项进行了合并。**

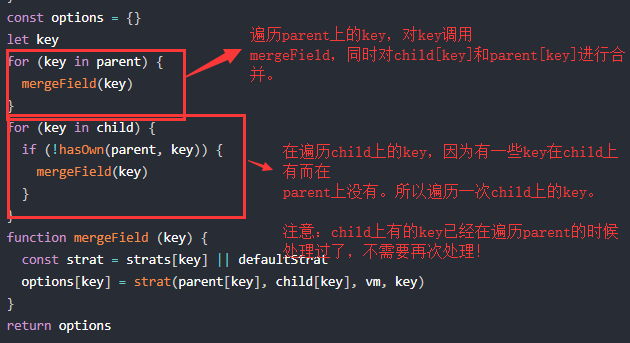
TIPS：最开始没注意到这一点坑了好久.......

因为strat对不同的key有不同的合并策略，所以我们只关注声明周期的合并，其他的需要看源码的时候再回头浏览：



这里就是对声明周期类key的合并处理方式，看上去用了很多三元运算符，逻辑也很简单，我们只需要知道这个结果就是，把child和parent的相同的生命周期按parent在前，extends在后的顺序合并为一个数组，然后在了options[key]上。

然后回到mergeField在mergeOption中的用武之地：



最终，我们获取到了融合过后的options，mergeOption只要返回options就完成了使命。

TIP：这个地因为光说合并的方式不直观，所以用一个实验来直观的表示：



