

02. 定义选项对象

学习要点：

1. 框架理解
2. options
3. 生命周期

本节课我们来开始学习 Vue 声明对象，并且理解传递选项对象的方法。

一. 框架理解

1. 所有的前端框架基本都是为了简化：模版渲染、事件绑定和用户交互问题；
2. Vue 参考了 MVVM 模型，即：视图(View)-视图模型(ViewModel)-模型(Model)；
3. 当我们 new Vue() 创建一个实例时，括号内传递的是一个选项对象(options)；
4. 这个选项对象一般包含：数据、模版、挂载元素、方法、生命周期钩子等等；
5. 当 Vue 实例被创建时，options 对象属性都会被添加到 Vue 的响应式系统中；
6. 所谓响应式系统，就是可以通过修改这些属性的值，让视图执行更新渲染；
7. 我们可以通过火狐 F12 控制器来修改 Vue 对象数据值，也可以使用 Vue 调试；
8. 当然，我们也可以将数据对象设置为保护不得修改，具体代码如下：

```
//数据对象
const dataObj = {
  message : 'Hello, Vue!'
};

//数据冻结
Object.freeze(dataObj);

//创建一个 Vue 实例
const app = new Vue({
  el : '#app',
  data : dataObj
});
```

二. options

1. 在 options 对象中，最为常用的就是模版、数据、方法以及挂载元素；
2. 首先，模版是视图中 {{var}} 这样的存在，它通过 el 属性挂载绑定指定元素；

```
//el 固定属性，挂载指定元素
el : '#app',

<!--渲染 id="app" 中的模版-->
<div id="app">
  {{message}}
</div>
```

3. 数据定义，使用 **data** 属性实现，它的值也是一个以对象形式的存在；

```
//data 固定属性，定义数据
//数据中的属性名和模板变量名一致
data : {
  message : 'Hello, Vue!'
}
```

4. 方法的执行，需要通过 **methods** 属性来定义方法，并使用指令执行；

```
//methods 固定属性，执行方法
methods : {
  alert : function () {
    alert('点击');
  }
}

<!--渲染 id="app" 中的模版-->
<div id="app">
  <button v-on:click="alert">点击</button>
</div>
```

三. 生命周期

1. 当我们创建 **Vue** 实例时，框架会执行一系列操作，比如监听、编译、挂载等；
2. 而这个过程也会执行一些生命周期钩子的函数，用于某个阶段进行触发；

```
//created 固定方法，生命周期钩子
//这个方法在初始化后自动执行
created : function () {
  console.log('初始化后：' + this.message);
},
//当修改并更新渲染后执行的钩子
updated : function () {
  console.log('修改后：' + this.message);
}
```

3. 那！整个生命周期的执行顺序是如何的？有没有流程图啊？有的，太大，官网见！
<https://cn.vuejs.org/v2/guide/instance.html#生命周期图示>