

## • Vue的双向绑定

### ■ 核心

#### Object.defineProperty()方法

### ■ 原理

通过**Object.defineProperty()**来劫持各个属性的setter, getter, 在数据发生变动时通知Vue实例, 触发相应的getter和setter回调函数。当把一个普通 Javascript 对象传给 Vue 实例来作为它的 data 选项时, Vue 将遍历它的属性, 用 Object.defineProperty 将它们转为 getter/setter。用户看不到 getter/setter, 但是在内部它们让 Vue 追踪依赖, 在属性被访问和修改时通知变化。

### ■ 特点

Vue的数据双向绑定将MVVM作为数据绑定的入口, 整合Observer, Compile和Watcher三者, 通过Observer来监听自己的model的数据变化, 通过Compile来解析编译模板指令 (vue中是用来解析 {{}}), 最终利用watcher搭起observer和Compile之间的通信桥梁, 达到数据变化 —> 视图更新; 视图交互变化 (input) —> 数据model变更双向绑定效果。

这就是通过数据劫持和发布-订阅者功能来实现的

### ■ js的简单实现

```
<body>
  <div id="app">
    <input type="text" id="txt">
    <p id="show"></p>
  </div>
</body>
<script type="text/javascript">
  var obj = {}
  Object.defineProperty(obj, 'txt', {
    get: function () {
      return obj
    },
    set: function (newValue) {
      document.getElementById('txt').value = newValue
      document.getElementById('show').innerHTML = newValue
    }
  })
  document.addEventListener('keyup', function (e) {
    obj.txt = e.target.value
  })
</script>
```