

响应式数据

luckyadam

MVVM

WPF

Silverlight

Angular

Vue

Ÿ

```
<template>
  <div class="home">
    <input v-model="text" >
    <p>{{text}}</p>
  </div>
</template>
```

```
<script>
  export default {
    data () {
      return {
        text: ''
      }
    }
  };
</script>
```

双向绑定

- 脏检测
- `Object.observe()`
- `Object.defineProperty()`

脏检测

Angular

- 重新封装各种浏览器事件、如click、input、xhr事件等
- 为scope作用域初始化数据创建\$watch对象，加入\$\$watch队列
- 事件触发调用\$digest cycle
- 遍历每一个数据的\$watcher，比较新旧值
- 新旧值不同时触发监听回调

~~定时遍历检测~~

Object.defineProperty

ES5·原生支持

定义或修改对象属性，封装属性访问器

[illegible]

Object.defineProperty

Object.defineProperty(obj, prop, descriptor)

- obj , 待修改的对象
- prop , 属性名称
- descriptor 要求传入一个对象, 其默认值如下

```
{  
  configurable: false,  
  enumerable: false,  
  writable: false,  
  value: null,  
  set: undefined,  
  get: undefined  
}
```

Object.defineProperty

getter/setter

```
let value = 1;
const obj = {};
Object.defineProperty(obj, 'a', {
  configurable: true,
  enumerable: true,
  set (newVal) {
    console.log('设置了a的值');
    value = newVal
  },
  get () {
    console.log(`访问了a的值， 它的值为${value}`);
    return value;
  }
});
```

Object.defineProperty

监听一个对象的变化

使用递归的方式，监听对象的每一个属性

Object.defineProperty

监听数组的变化

重新包装影响数组的方法

['push', 'pop', 'shift', 'unshift', 'splice', 'sort', 'reverse']